

T.C.  
SAKARYA ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

ENDÜKTİF ÖĞRENMEDE BİLGİ KAZANIMI İÇİN  
YENİ ALGORİTMALAR

736442

DOKTORA TEZİ

End.Yük.Müh. Ömer AKGÖBEK

Enstitü Anabilim Dalı : ENDÜSTRİ MÜHENDİSLİĞİ

Tez Danışmanı : Prof.Dr. Ercan ÖZTEMEL

HAZİRAN 2003

EC. YÜKSEKÖĞRETİM KURULU  
EĞİTİM ARAŞTIRMA MERKEZİ

T.C.  
SAKARYA ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

**ENDÜKTİF ÖĞRENMEDE BİLGİ KAZANIMI İÇİN  
YENİ ALGORİTMALAR**

**DOKTORA TEZİ**

End.Yük.Müh. Ömer AKGÖBEK

136442

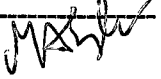
**Enstitü Anabilim Dalı : ENDÜSTRİ MÜHENDİSLİĞİ**

**Enstitü Bilim Dalı : ENDÜSTRİ MÜHENDİSLİĞİ**

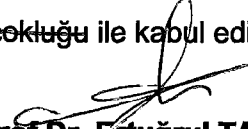
Bu tez . . . / . . . / 2003 tarihinde aşağıdaki jüri tarafından Oybirliği/Oyçokluğu ile kabul edilmiştir.

  
**Prof.Dr. Ercan ÖZTEMEL**

Jüri Başkanı

  
**Prof.Dr. M. Akif EYLER**

Jüri Üyesi

  
**Prof.Dr. Ertuğrul TAÇGIN**

Jüri Üyesi

  
**Doç.Dr. Orhan TORKUL**

Jüri Üyesi

**Y.Doç.Dr. Nejat YUMUŞAK**

Jüri Üyesi



## TEŞEKKÜR

Yaptığım bu çalışmada, tez konusunun seçiminden başlayarak, çalışmanın her safhasında bilgilerine başvurduğum Sayın **Doç.Dr. M. Sabih AKSOY**'a katkılarından dolayı teşekkür ederim. Tezin hazırlanması, yazımı, uygulama kısımlarının tamamlanması, programların yazılması ve kullanılan örnek veri setlerinin seçilmesi aşamalarında sürekli yol gösteren ve bu konularda destek ve bilgilerini hiç esirgemeyen, ayrıca kazandırdığı düzenli ve planlı çalışma temposundan dolayı tez danışmanım Sayın **Prof. Dr. Ercan ÖZTEMEL**'e teşekkürü bir borç bilirim. Tezin hazırlanması sırasında sürekli bilgi alışverişi yaptığım mesai arkadaşlarım Arş.Gör. Mümtaz İPEK, Arş.Gör. H.İbrahim DEMİR ve Arş.Gör. Özer UYGUN'a teşekkür ederim. Ayrıca, tezin tüm aşamalarında beni her zaman sabırla destekleyen aileme müteşekkirim.

## İÇİNDEKİLER

|                                                                  |      |
|------------------------------------------------------------------|------|
| TEŞEKKÜR .....                                                   | ii   |
| İÇİNDEKİLER .....                                                | iii  |
| SİMGELER VE KISALTMALAR .....                                    | vii  |
| ŞEKİLLER LİSTESİ .....                                           | viii |
| TABLolar LİSTESİ .....                                           | ix   |
| ÖZET .....                                                       | xii  |
| SUMMARY .....                                                    | xiii |
| BÖLÜM 1.                                                         |      |
| GİRİŞ .....                                                      | 1    |
| 1.1. Tezin Amacı .....                                           | 2    |
| 1.2. Tezin İçeriği .....                                         | 3    |
| BÖLÜM 2.                                                         |      |
| BİLGİSAYARLA ÖĞRENME ve BİLGİ KAZANIMI .....                     | 4    |
| 2.1. Giriş .....                                                 | 4    |
| 2.2. Tümevarım ve Endüktif Öğrenme .....                         | 7    |
| 2.2.1 Tümevarım süreçleri .....                                  | 7    |
| 2.2.2. Endüktif öğrenme .....                                    | 9    |
| 2.3. Endüktif Öğrenme Algoritmaları .....                        | 11   |
| 2.3.1. Endüktif öğrenme algoritmalarının sınıflandırılması ..... | 15   |
| 2.3.1.1. Karar ağacı tabanlı algoritmalar .....                  | 15   |
| 2.3.1.2. Kural tabanlı algoritmalar .....                        | 20   |
| 2.4. Endüktif Öğrenme Algoritmalarının Uygulaması .....          | 21   |
| 2.5. Bilgi Ölçümü, Entropi ve Bilgi Kazancı .....                | 23   |
| 2.5.1. Entropi ve bilgi kazancının hesaplanması .....            | 25   |

### BÖLÜM 3.

|                                                            |    |
|------------------------------------------------------------|----|
| ENDÜKTİF ÖĞRENME ALGORİTMALARI .....                       | 28 |
| 3.1. Giriş .....                                           | 28 |
| 3.2 Endüktif Öğrenme Tekniklerinin Sınıflandırılması ..... | 30 |
| 3.3. CLS Ailesi Algoritmaları .....                        | 30 |
| 3.3.1. Örnek uygulama .....                                | 31 |
| 3.3.2. CLS ailesinin diğer algoritmaları .....             | 36 |
| 3.4. ID3 Ailesi Algoritmaları .....                        | 37 |
| 3.4.1. Örnek uygulama .....                                | 38 |
| 3.4.2. ID3 algoritmasından türetilen algoritmalar .....    | 41 |
| 3.5. Rules Ailesi Algoritmaları .....                      | 43 |
| 3.5.1. RULES - 1 algoritması .....                         | 43 |
| 3.5.1.1. Örnek uygulama .....                              | 45 |
| 3.5.2. RULES-2 algoritması .....                           | 48 |
| 3.5.3. RULES-3 algoritması.....                            | 49 |
| 3.5.4. RULES-3 PLUS ve RULES-4 algoritmaları .....         | 50 |
| 3.6. AQ Ailesi Algoritmaları .....                         | 54 |
| 3.7. C4.5 algoritması .....                                | 57 |
| 3.8. J-Pruned Algoritması .....                            | 58 |
| 3.9. CART ve OC1 algoritmaları .....                       | 58 |
| 3.10. PRISM Algoritması .....                              | 59 |
| 3.11. Sonuç .....                                          | 60 |

### BÖLÜM 4.

|                                                                    |    |
|--------------------------------------------------------------------|----|
| REX-1 : ENDÜKTİF ÖĞRENME ALGORİTMASI .....                         | 62 |
| 4.1. Giriş .....                                                   | 62 |
| 4.2. Algoritmanın Tanımı .....                                     | 62 |
| 4.2.1. REX-1 algoritmasının özellikleri .....                      | 63 |
| 4.2.1.1. Entropi ve düzeltilmiş entropi değerlerini kullanma ..... | 64 |
| 4.2.1.2. Kısa işlem süresi .....                                   | 64 |
| 4.2.1.3. Eksik değerli örnekleri işleyebilme .....                 | 65 |
| 4.2.1.4. Sayısal değer içeren karakteristikleri işleyebilme .....  | 65 |

|                                                                |    |
|----------------------------------------------------------------|----|
| 4.2.2. Genel ihtiyaçlar .....                                  | 65 |
| 4.2.3. Algoritma .....                                         | 65 |
| 4.2.4. Algoritma ile uygulama .....                            | 67 |
| 4.3. Diğer Algoritmalarla Karşılaştırma .....                  | 74 |
| 4.3.1. Yatırım projesi örneğini kullanarak karşılaştırma ..... | 74 |
| 4.3.2. ID3 algoritması ile karşılaştırma .....                 | 76 |
| 4.3.3. IRIS veri setini kullanarak karşılaştırma .....         | 76 |
| 4.3.4. Lens24 veri setini kullanarak karşılaştırma .....       | 80 |
| 4.4. Sonuçlar .....                                            | 82 |

## BÖLÜM 5.

|                                                                           |     |
|---------------------------------------------------------------------------|-----|
| REX-2 : ENDÜKTİF ÖĞRENME ALGORİTMASI .....                                | 84  |
| 5.1. Giriş .....                                                          | 84  |
| 5.2. Algoritmanın Tanımı .....                                            | 85  |
| 5.2.1. Algoritma .....                                                    | 87  |
| 5.2.2. Algoritmanın uygulanması .....                                     | 88  |
| 5.3. Diğer Algoritmalarla Karşılaştırma .....                             | 98  |
| 5.3.1. REX-1 algoritması ile karşılaştırma .....                          | 98  |
| 5.3.2. C4.5 algoritması ile karşılaştırma .....                           | 99  |
| 5.3.3. IRIS veri setini kullanarak diğer algoritmalarla karşılaştırma ... | 99  |
| 5.3.4. ID3 Algoritması ile karşılaştırma .....                            | 100 |
| 5.4. Sonuçlar .....                                                       | 103 |

## BÖLÜM 6.

|                                                        |     |
|--------------------------------------------------------|-----|
| REX-3 : ENDÜKTİF ÖĞRENME ALGORİTMASI .....             | 104 |
| 6.1. Giriş .....                                       | 104 |
| 6.2. Algoritmanın Tanımı.....                          | 105 |
| 6.2.1 Algoritmanın özellikleri .....                   | 105 |
| 6.2.2 Algoritma .....                                  | 106 |
| 6.2.3. Algoritmanın uygulanması .....                  | 106 |
| 6.3. Diğer Algoritmalarla Karşılaştırma .....          | 112 |
| 6.3.1. ILA ve C4.5 ile karşılaştırma .....             | 113 |
| 6.3.2. IRIS veri setini kullanarak karşılaştırma ..... | 113 |

|                                                                                      |     |
|--------------------------------------------------------------------------------------|-----|
| 6.3.3. RULES-3 algoritması ile karşılaştırma .....                                   | 114 |
| 6.3.4. IRIS_TEST veri setini kullanarak karşılaştırma .....                          | 114 |
| 6.4. Sonuçlar .....                                                                  | 116 |
| <br><b>BÖLÜM 7.</b>                                                                  |     |
| <b>SONUÇLAR</b> .....                                                                | 117 |
| 7.1. Giriş .....                                                                     | 117 |
| 7.2. Diğer Algoritmalarla Performans Karşılaştırma .....                             | 117 |
| 7.3. BALLOONS ve BALANCE Veri Setlerini Kullanarak Performans<br>Karşılaştırma ..... | 120 |
| 7.4. MONK Problemlerini Kullanarak Performans Karşılaştırma .....                    | 121 |
| 7.5. J-Pruned ile Performans Karşılaştırma .....                                     | 123 |
| 7.6. TDIDT ve PRISM Algoritmaları ile Performans Karşılaştırma .....                 | 124 |
| 7.7. Geliştirilen Endüktif Öğrenme Programı (REX123) .....                           | 126 |
| 7.7.1. Programın özellikleri .....                                                   | 126 |
| 7.7.2. REX123 programı .....                                                         | 127 |
| <br><b>BÖLÜM 8.</b>                                                                  |     |
| <b>TARTIŞMA ve ÖNERİLER</b> .....                                                    | 130 |
| <b>KAYNAKLAR</b> .....                                                               | 132 |
| <b>EK-A. DÜZELTİLMİŞ ENTROPİ KULLANIMININ ÖNEMİ</b> .....                            | 148 |
| <b>ÖZGEÇMİŞ</b> .....                                                                | 153 |

## SİMGELER ve KISALTMALAR

|                   |                                                                                |
|-------------------|--------------------------------------------------------------------------------|
| $S, A, X$         | : Örnek seti                                                                   |
| $C$               | : Karakteristik seti                                                           |
| $X'$              | : Alt örnek seti                                                               |
| $K$               | : Geçmiş bilgi                                                                 |
| $d$               | : Sınıf                                                                        |
| $s$               | : Sınıf sayısı                                                                 |
| $e$               | : Örnek sayısı                                                                 |
| $H$               | : Hipotez uzayı                                                                |
| $h$               | : Hipotez                                                                      |
| $U$               | : Örnek uzayı                                                                  |
| $V$               | : Karakteristiklere ait değerler dizisi                                        |
| $P, \text{freq}$  | : Karakteristiklere ait değerlerin olasılığı                                   |
| $I$               | : Bilgi                                                                        |
| $E$               | : Entropi                                                                      |
| $D\text{Entropi}$ | : Düzeltilmiş entropi                                                          |
| Kazanç            | : Bilgi kazancı                                                                |
| Bilgi             | : Örnek setinin bilgi değeri                                                   |
| KısmiBilgi        | : Karakteristiğın bilgi değeri                                                 |
| KazançOran        | : Karakteristiğın bilgi kazanç oranı                                           |
| $E^c$             | : İfade tarafından kapsanan örneklerin sayısı                                  |
| $E_i^c$           | : Sınıf $i$ 'ye ait olan ve ifade tarafından kapsanan örneklerin toplam sayısı |
| $E_i$             | : Hedef sınıfına ait olan eğitim setindeki örneklerin toplam sayısı            |
| $n_{co}$          | : Şartların sayısı                                                             |
| $n_a$             | : Karakteristiklerin sayısı                                                    |
| $m_{PRSET}$       | : PRSET dizisindeki elemanların sayısı                                         |
| $T\_PRSET$        | : PRSET gibi aynı boyuta sahip, geçici dizi değişken                           |
| $T\_EXP$          | : İfadelerin oluşturduğu geçici dizi değişken                                  |

## ŞEKİLLER LİSTESİ

|                                                                                   |     |
|-----------------------------------------------------------------------------------|-----|
| Şekil 2.1. Örnek seti .....                                                       | 13  |
| Şekil 2.2. Şekil 2.1'deki örnek setinden çıkarılan kural seti ve karar ağacı ..   | 14  |
| Şekil 2.3. Böl-ve-Fethet endüktif öğrenme metodunda ayrıştırma prosedürü          | 17  |
| Şekil 3.1. Karar ağacı örneği .....                                               | 29  |
| Şekil 3.2. CLS Algoritması .....                                                  | 31  |
| Şekil 3.3. Tablo 3.1'de verilen örnekler setine ait birinci kısmi karar ağacı ..  | 35  |
| Şekil 3.4. Tablo 3.3'te verilen örnekler seti için ikinci kısmi karar ağacı ..... | 35  |
| Şekil 3.5. Tablo 3.2'de verilen örnekler seti için son karar ağacı .....          | 36  |
| Şekil 3.6. ID3 Algoritması .....                                                  | 37  |
| Şekil 3.7. Sıcaklık karakteristiğine göre hazırlanan karar ağacı .....            | 39  |
| Şekil 3.8. Sıcaklık karakteristiğine göre hazırlanan kısmi karar ağacı .....      | 40  |
| Şekil 3.9. Mevsimler örnek seti için son karar ağacı .....                        | 41  |
| Şekil 3.10. RULES-1 algoritması .....                                             | 44  |
| Şekil 3.11. Rules-3 Plus algoritması .....                                        | 53  |
| Şekil 3.12. AQ algoritması .....                                                  | 56  |
| Şekil 3.13. PRISM algoritması .....                                               | 60  |
| Şekil 4.1. REX-1 algoritması .....                                                | 66  |
| Şekil 5.1. REX-2 algoritması .....                                                | 87  |
| Şekil 5.2. ID3 tarafından üretilen karar ağacı .....                              | 101 |
| Şekil 6.1. REX-3 algoritması .....                                                | 107 |
| Şekil 7.1. REX123 programının ana menüsü .....                                    | 127 |
| Şekil 7.2. Bilgiler ekranı .....                                                  | 128 |
| Şekil 7.3. Kural çıkarma ekranı .....                                             | 128 |

## TABLolar LİSTESİ

|                                                                                                     |    |
|-----------------------------------------------------------------------------------------------------|----|
| Tablo 2.1. Şekil 2.1'deki örnekler setinin karakteristik-değer biçimi .....                         | 14 |
| Tablo 2.2. Karar ağacı üreten algoritmalar .....                                                    | 18 |
| Tablo 2.3. Doğrudan kural üreten algoritmalar .....                                                 | 21 |
| Tablo 3.1. Mevsimler örnek seti .....                                                               | 28 |
| Tablo 3.2. CLS algoritması için örnek seti .....                                                    | 32 |
| Tablo 3.3. Tablo 3.2'deki örnek setinin Pozitif ve Negatif olarak ayrı ayrı sınıflandırılması ..... | 33 |
| Tablo 3.4. Şekil 3.5'teki karar ağacından elde edilen kural seti .....                              | 36 |
| Tablo 3.5. Mevsimler eğitim seti .....                                                              | 38 |
| Tablo 3.6. Karakteristikler ve değerlere ait entropi değerleri .....                                | 39 |
| Tablo 3.7. <i>SI</i> alt seti için entropi ve bilgi kazancı değerleri .....                         | 40 |
| Tablo 3.8. Mevsimler problemi için elde edilen kurallar (Şekil 3.9'dan) .....                       | 41 |
| Tablo 3.9. Mevsimler için örnek seti .....                                                          | 45 |
| Tablo 3.10. Birinci aşama için karakteristik değerlerinin yer aldığı sınıflar ....                  | 46 |
| Tablo 3.11. Birinci aşama sonunda elde edilen kurallar .....                                        | 46 |
| Tablo 3.12. Birinci aşama sonunda sınıflandırılmayan örnekler listesi .....                         | 46 |
| Tablo 3.13. Birinci aşamadan sonra kural çıkarılamayan değerler listesi .....                       | 47 |
| Tablo 3.14. İkinci aşama ile elde edilen kurallar .....                                             | 47 |
| Tablo 3.15. Mevsimler eğitim seti için Rules-2 tarafından üretilen kurallar ...                     | 47 |
| Tablo 3.16. Mevsimler eğitim seti için Rules-3 tarafından üretilen kurallar ...                     | 50 |
| Tablo 3.17. AQ tarafından elde edilen kurallar (Mevsimler eğitim seti) .....                        | 56 |
| Tablo 4.1. Mevsimler eğitim seti .....                                                              | 67 |
| Tablo 4.2. Karakteristikler ve değerlere ait entropi değerleri .....                                | 71 |
| Tablo 4.3. Yeniden düzenlenmiş örnek seti (Tablo 4.1) .....                                         | 72 |
| Tablo 4.4. Birinci aşama sonunda sınıflandırılmayan örnekler .....                                  | 73 |
| Tablo 4.5. Tablo 4.1'de verilen örnek seti için üretilen kurallar .....                             | 74 |
| Tablo 4.6. Yatırım projesi sınıflandırma eğitim seti .....                                          | 75 |

|                                                                                                           |     |
|-----------------------------------------------------------------------------------------------------------|-----|
| Tablo 4.7. ID3, C4.5, ILA, RULES-3 ve REX-1 ile üretilen kurallar .....                                   | 75  |
| Tablo 4.8. Geliştirilen algoritmanın ID3 algoritması ile karşılaştırılması .....                          | 76  |
| Tablo 4.9. IRIS Veri Seti .....                                                                           | 77  |
| Tablo 4.9. (Devam) IRIS veri seti.....                                                                    | 77  |
| Tablo 4.10a. RULES-3 tarafından elde edilen kural seti (Iris veri seti) .....                             | 79  |
| Tablo 4.10b. ID3 tarafından elde edilen kural seti (Iris veri seti) .....                                 | 79  |
| Tablo 4.10c. Rules-3 Plus tarafından elde edilen kural seti (Iris veri seti) ....                         | 79  |
| Tablo 4.10d. REX-1 tarafından elde edilen kural seti (Iris veri seti) .....                               | 80  |
| Tablo 4.11. Kural sayısı ve kural başına şartların ortalaması (Iris veri seti) ...                        | 80  |
| Tablo 4.12. Lens24 veri seti .....                                                                        | 81  |
| Tablo 4.13. GDT-NR, GDT-RS ve REX-1 tarafından üretilen kurallar .....                                    | 81  |
| Tablo 4.14. C4.5prune tarafından elde edilen kurallar .....                                               | 82  |
| Tablo 5.1. Golf oynama problemi eğitim seti .....                                                         | 88  |
| Tablo 5.2. Karakteristikler ve değerlere ait entropi değerleri .....                                      | 92  |
| Tablo 5.3. Yeniden düzenlenmiş örnek seti .....                                                           | 96  |
| Tablo 5.4. REX-2 algoritması ile üretilen kurallar (Golf problemi) .....                                  | 98  |
| Tablo 5.5. REX-1 algoritması ile üretilen kurallar (Golf problemi) .....                                  | 99  |
| Tablo 5.6. C4.5 algoritması ile üretilen kurallar (Golf Problemi) .....                                   | 99  |
| Tablo 5.7. REX-2 tarafından elde edilen kurallar (IRIS veri seti) .....                                   | 99  |
| Tablo 5.8. Kural sayısı ve kural başına şartların ortalaması (IRIS veri seti)...                          | 100 |
| Tablo 5.9. Tolerans dizaynı için örnek seti .....                                                         | 101 |
| Tablo 5.10. ID3 tarafından elde edilen kurallar (Şekil 5.2'deki karar ağacından) .....                    | 102 |
| Tablo 5.11. REX-2 tarafından elde edilen kurallar.....                                                    | 102 |
| Tablo 5.12. Kurallar tarafından sınıflandırılan örnekler .....                                            | 103 |
| Tablo 6.1. Yatırım Projesi-2 veri seti .....                                                              | 107 |
| Tablo 6.2a. İlk üç kural ile sınıflandırılan örnekler .....                                               | 110 |
| Tablo 6.2b. Elde edilen kuralar ile sınıflandırılan örnekler .....                                        | 112 |
| Tablo 6.3. REX-3 ile Yatırım Projesi-2 kullanılarak elde edilen kurallar .....                            | 112 |
| Tablo 6.4. REX-3 tarafından elde edilen kurallar (IRIS veri seti) .....                                   | 113 |
| Tablo 6.5. REX-3 tarafından elde edilen kural ve kural başına şartların ortalaması (IRIS veri seti) ..... | 113 |
| Tablo 6.6. REX-3 ve Rules-3 tarafından elde edilen kurallar .....                                         | 114 |

|                                                                                  |     |
|----------------------------------------------------------------------------------|-----|
| Tablo 6.7. IRIS_Test veri seti .....                                             | 115 |
| Tablo 6.8. IRIS test veri seti kullanılarak elde edilen doğruluk oranları .....  | 116 |
| Tablo 7.1. Eğitim veri setlerine ait özellikler .....                            | 118 |
| Tablo 7.2. Test veri setlerine ait özellikler .....                              | 119 |
| Tablo 7.3. Algoritmalar tarafından elde edilen doğruluk oranları .....           | 120 |
| Tablo 7.4. Balloons ve Balance veri setleri kullanılarak elde edilen sonuçlar.   | 121 |
| Tablo 7.5 Monk problemleri için elde edilen kural ve şart sayıları .....         | 122 |
| Tablo 7.6. Test veri setleri kullanılarak elde edilen doğruluk oranları .....    | 123 |
| Tablo 7.7. <i>J-Pruned</i> ile performans karşılaştırma .....                    | 124 |
| Tablo 7.8. Algoritmalar tarafından elde edilen sonuçlar .....                    | 125 |
| Tablo 7.9. Test setlerini kullanarak doğruluk oranlarının karşılaştırılması .... | 125 |
| Tablo A.1. Golf Problemi örnek seti .....                                        | 143 |
| Tablo A.2. Sıcaklık ve Nem karakteristikleri için aralık tanımı .....            | 144 |
| Tablo A.3. Aralık tanımlamalarına göre düzenlenmiş örnek seti .....              | 144 |
| Tablo A.4. Golf Problemi için hesaplanan entropi değerleri .....                 | 145 |
| Tablo A.5. Entropi değerlerine göre üretilen kurallar .....                      | 146 |
| Tablo A.6. Entropi değerlerine göre üretilen kurallar .....                      | 147 |

## ÖZET

**Anahtar Kelimeler :** Bilgisayarla öğrenme, Endüktif öğrenme, Bilgi kazanımı, Entropi, Endüktif öğrenme algoritmaları.

Bilgi, genel olarak bir uzmandan, konu ile ilgili kaynaklardan, arşiv bilgilerinden, gözlem veya deneylerden elde edilebilir. Bir uzmandan bilgi elde etme, karşılıklı görüşmeler gerektiren, uzun zaman alan, dikkat isteyen ve sistematik çalışmalar gerektiren bir işlemdir. Uzmanlar, uzmanlık bilgilerini günlük çalışmalarında rahatlıkla kullanabilmelerine rağmen, bunları özetleme ve bir uzman sistemde kullanılabilir hale getirmede aynı başarıyı gösteremeyebilirler. Bu bilgilerin değerlendirilmesi ve bir uzman sistem için bilgi tabanı haline getirilmesi ayrı bir uzmanlık ister. Bilgi tabanının bu yolla oluşturulması hem uzun zaman almakta hem de yüksek maliyet gerektirmektedir. Çünkü bu işlemleri yerine getirecek uzman kişilere ihtiyaç vardır. Bu elemanları bulmak çok zor ve istihdamı da yüksek maliyet gerektirmektedir. Araştırmacılar bilgi kazanımını, bir uzman sistemin geliştirilmesindeki en büyük darboğaz olarak kabul etmektedirler.

Bütün bu ve benzeri zorluklar, araştırmacıları bu darboğazı aşmak için alternatif teknikler geliştirmeye sevk etmiştir. Bu konuda birçok başarılı teknikler geliştirilmiş ve geliştirilmeye devam edilmektedir. Yapılan çalışmalarda olumlu sonuçlar elde edilmiştir. Geliştirilen bu tekniklerin amacı bilgi kazanımını otomatik hale getirmektir.

Bu çalışmada, otomatik bilgi kazanımını sağlamak amacıyla üç algoritma geliştirilmiştir: REX-1, REX-2 ve REX-3. Geliştirilen algoritmaların verimliliğini test etmek amacıyla dünyada bu alanda kullanılan ve gerçek hayattan alınan veri setleri kullanılmıştır. Bu veri setleri kullanılarak, günümüzün en çok kullanılan algoritmalarından olan ID3, C4.5, RULES-3, RULES 3 PLUS, RULES-4, ILA, ILA-2, PRISM, OC1, CN2, J-PRUNED, GDT-NR ve GDT-RS ile performans karşılaştırmaları yapılmıştır. Yapılan performans karşılaştırmalarında, algoritmalar tarafından elde edilen sonuçların iyi olduğu tespit edilmiştir. Algoritmaların büyük veri setleri için işlem yapması için DELPHI-6 programlama dili kullanılarak bir program hazırlanmıştır.

# NEW ALGORITHMS FOR KNOWLEDGE ACQUISITION IN INDUCTIVE LEARNING

## SUMMARY

**Keywords :** Machine learning, Inductive learning, Knowledge acquisition, Entropy, Inductive learning algorithms.

In general, knowledge can be gathered from an expert, from the resources related to the subject, from archives, or from observations and experiments. The process of knowledge gathering from an expert usually involves interviews, and thus is a time consuming method requiring careful and systematic examination. Although the experts can utilize their skills comfortably in their daily life, they may not demonstrate the same achievement in summarizing the knowledge, as well as making use of it within an expert system. To evaluate the available data and to translate it into an expert system database demands a different proficiency. The creation of the database through this route is time consuming and costly because of the need for the skilled human resources to accomplish the job which is a difficult task requiring higher costs. Therefore, researchers identify the knowledge acquisition as the most important element in the process of developing an expert system.

Such difficulties have directed researchers to developing alternative techniques for going beyond this obstruction. Several successful procedures have been/are being developed on this subject and encouraging results have already been obtained. The aim of these newly developed techniques is to automatize the process of knowledge acquisition.

In this study, three algorithms have been developed to make available the knowledge acquisition: REX-1, REX-2 and REX-3. In order to evaluate the performance of these algorithms, the database taken from real life and are employed in this field in the world have been used. Performance comparisons with today's mostly used algorithms, such as ID3, C4.5, RULES-3, RULES 3 PLUS, RULES-4, ILA, ILA-2, PRISM, OC1, CN2, J-PRUNED, GDT-NR and GDT-RS, have been carried out and the results from our algorithms were found to be acceptable. A program written in DELPHI-6 was also prepared in order to utilize these algorithms for larger databases.

## BÖLÜM 1. GİRİŞ

“Bilgi Çağı” ve “Bilgi Toplumu” gibi terimlerin sıklıkla kullanıldığı günümüzde bilginin önemi daha açık bir şekilde ortaya çıkmaktadır. Bilginin önemi arttığı oranda o bilgiye ulaşabilmeyi sağlayan sistemlerin de önemi artmaktadır. Bilgisayar teknolojisindeki büyük gelişmeler sayesinde, dünyanın herhangi bir yerinde üretilen bilginin sayısal hale getirilerek saklanması ve o bilgiye dünyanın herhangi bir yerinden çok kısa sürede erişim mümkün olmaktadır. Bununla birlikte programlama dillerindeki büyük gelişmeler sayesinde bilgiyi işlemek ve istenen bilgiye erişmek de kolaylaşmaktadır.

Bilgisayar teknolojisinin büyük bir hızla gelişmesi ile beraber yapay zeka alanında da çok büyük ve hızlı gelişmeler yaşanmaktadır. Bu gelişmelerle beraber yapay zekanın bir alt dalı olan uzman sistemlerde de büyük gelişmeler yaşanmıştır.

Bilgi, genel olarak bir uzmandan, konu ile ilgili kaynaklardan, arşiv bilgilerinden, gözlem veya deneylerden elde edilebilir. Bir uzmandan bilgi elde etme, karşılıklı görüşmeler gerektiren, uzun zaman alan, dikkat isteyen ve sistematik çalışmalar gerektiren bir işlemdir. Uzmanlar, uzmanlık bilgilerini günlük çalışmalarında rahatlıkla kullanabilmelerine rağmen, bunları özetleme ve bir uzman sistemde kullanılabilir hale getirmede aynı başarıyı gösteremeyebilirler. Bu bilgilerin değerlendirilmesi ve bir uzman sistem için bilgi tabanı haline getirilmesi ayrı bir uzmanlık ister. Bilgi tabanının bu yolla oluşturulması hem uzun zaman almakta hem de yüksek maliyet gerektirmektedir. Çünkü bu işlemleri yerine getirecek uzman kişilere ihtiyaç vardır. Bu elemanları bulmak çok zor ve istihdamı da yüksek maliyet gerektirmektedir.

Bilginin elde edilmesine bilgi kazanımı denir. Araştırmacılar bilgi kazanımını, bir uzman sistemin geliştirilmesinde en büyük darboğaz olarak kabul etmektedirler. Feigenbaum bu hususu şöyle dile getirmektedir: “Bilginin gösterimi, kullanılması ve

kazanılması ile ilgili çözülmesi gereken çok önemli problemler vardır. Bunlardan bilgi kazanımı problemi en önemli kritik darboğazı oluşturmaktadır” [1].

Bütün bu ve benzeri zorluklar araştırmacıları bu darboğazı aşmak için alternatif teknikler geliştirmeye sevk etmiştir. Bu konuda birçok başarılı teknik geliştirilmiş ve geliştirilmeye devam edilmektedir. Bu çalışmalarla olumlu sonuçlar elde edilmiştir. Geliştirilen bütün bu tekniklerin amacı bilgi kazanımını otomatik hale getirmektir. Bunlardan bir tanesi de Endüktif Öğrenme (Inductive Learning) tekniğidir.

### 1.1. Tezin Amacı

Bilgisayarla öğrenmenin (Machine Learning) bir alt alanı olan Endüktif Öğrenme (Inductive Learning)’de temel amaç bilgi tabanını oluştururken direkt olarak uzmandan bilgiyi almak yerine, uzmanlık sahası ile ilgili örneklerden yararlanarak bu işlemi gerçekleştirmektir. Bu şekilde bilgi tabanı daha kolay, hızlı ve pratik olarak oluşturulmaktadır. Ayrıca bilgi mühendislerine de ihtiyaç kalmamaktadır. Uzman açısından da daha kolay olan bu yolla, uzmanın uzmanlık alanıyla ilgili bilgileri örnekler haline getirmesi çok daha kolay olmaktadır. Örneğin bir tıp doktorunun hastalarına sormuş olduğu soruları, aldığı cevapları, yaptırdığı testleri, bunların sonuçlarını ve nihayet teşhislerini bir yere kaydetmek çok daha kolaydır. Böylece sürekli olarak doktora sorular sorarak bilgi almak yerine, eldeki bilgilere bakarak bir takım bilgiler çıkarmak mümkündür. Çünkü aranan bilgi eldeki örneklerde gizli olacaktır. Bu örneklerin belirli bir teknik kullanılarak gerekli bilgiyi elde etmek çok daha kolaydır[2]. Böylece yukarıda bahsedilen bilgi kazanımı darboğazı da bu şekilde aşılmış olacaktır.

Bu tezin amacı otomatik bilgi kazancı için yeni teknikler ve algoritmalar geliştirmektir. Geliştirilecek olan bu yeni teknik ve algoritmalar ile mevcut teknik ve algoritmaların eksiklik ve dezavantajlarının giderilmesi amaçlanmaktadır. Günümüzde en çok kullanılan tekniklerin başında karar-ağacı oluşturma teknikleri gelmektedir. Ancak karar-ağacı oluşturma ve bunu kural setine dönüştürmenin zorlukları ve sakıncaları vardır. Bunun için doğrudan kural çıkarma yaklaşımına göre algoritmalar geliştirilecektir.

## 1.2. Tezin İçeriği

Tezin birinci bölümünde, bilgi kazancının ve bilgiyi elde etmenin önemi üzerinde duruldu.

İkinci bölümde, Endüktif Öğrenme hakkında detaylı bilgiler verilmiştir. Endüktif öğrenmede kullanılan teknikler, bu tekniklerin sınıflandırılması ile endüktif öğrenmenin kullanım alanları hakkında detaylı bilgiler ve bu alanda yapılan çalışmalar özetlenmiştir.

Üçüncü bölümde, Endüktif Öğrenme algoritmaları hakkında detaylı bilgiler verilecektir. En iyi bilinen Endüktif öğrenme algoritmalarının çalışma prosedürleri adım adım gösterilmiştir. Bazı algoritmalar hakkında birer örnek uygulama yapılarak temel algoritmaların daha iyi anlaşılması hedeflenmiştir.

Dördüncü bölümde, yeni geliştirilen ilk algoritma olan REX-1 algoritması detaylı olarak anlatılmış ve diğer algoritmalarla performans testleri yapılmıştır.

Beşinci bölümde, yeni geliştirilen ikinci algoritma olan REX-2 algoritması detaylı olarak anlatılmış, bir örnek problem çözülmüş ve diğer algoritmalarla performans testleri yapılmıştır.

Altıncı bölümde, yeni geliştirilen üçüncü algoritma olan REX-3 algoritması anlatılmıştır. Diğer algoritmalarda olduğu gibi bu algoritma da detaylı olarak anlatılmış ve diğer algoritmalarla gerekli testler yapılmıştır.

Yedinci bölümde, geliştirilen her üç algoritma, dünyada bu alanda kullanılan gerçek veri setleri kullanılarak bu alandaki diğer algoritmalarla performans karşılaştırmaları yapılmış ve sonuçları sunulmuştur.

Sekizinci bölümde ise tartışma ve geleceğe yönelik yapılabilecek çalışmalar hakkında bilgiler verilmiştir.

## **BÖLÜM 2. BİLGİSAYARLA ÖĞRENME VE BİLGİ KAZANIMI**

### **2.1. Giriş**

Bilgisayarla Öğrenme (Machine Learning), yapay zekanın bir alt alanıdır. Sistemlerinin performanslarını artırmak veya bilgi esaslarını genişletmek amacıyla sistem davranışlarında meydana gelen kalıcı değişiklikler “öğrenme” olarak adlandırılır. Öğrenme yeteneği, insanlar ve diğer canlı organizmalarda mevcut olup bilgisayar sistemlerinde de büyük bir değere sahiptir. Bilgisayarla öğrenmenin temel amacı bilgi kazanımını otomatik hale getirmektir. Çünkü, bilgi tabanlı sistemler hazırlamak çok pahalıdır. Uzmanlardan bilgiyi elde etmenin ve sonucu yararlı bir şekilde göstermenin süreci yavaştır. Ayrıca, hem konuyla ilgili uzmanların hem de yapay zeka uzmanlarının hizmetlerini gerektirir. Bilgi transfer sürecini bir makine tarafından doğrudan elde etme ve kendi iç gösterimini yapma, yeni uzman sistemlerin maliyetini düşürmeyi ve kendi kendini sürekli olarak geliştiren sistemler için hızlı ve verimli araçlar sağlamayı hedefler [3].

İnsanlar ve bilgisayarlar arasındaki büyük farklardan biri, insanlar bir faaliyete girişirken genellikle eş zamanlı geliştirme yoluyla kendi deneyimlerini kullanarak gerçekleştirirler. Buna karşın, bilgisayarlar kendileri için sağlanan prosedürleri (programları) tam olarak yerine getirirler. Bilgisayarlar, programları çok verimli olarak icra edebilirler ama bilgisayarların performansı kendi deneyimleri ile geliştirilmemiştir[4].

Öğrenen bir sistem, bir veya daha fazla görevi gerçekleştirirken kendi bilgi veya yeteneğini de geliştirir. Gelişme, bilgi işleme aktivitelerinin bir sonucu olarak meydana gelir. Gelişme bir çok şekilde ele alınmalıdır[3]:

- Bir bilgi tabanının büyüülmesi,

- Varolan bilginin arıtılması süresince bilgi tabanının boyutundaki azaltma,
- Buluşsal yöntemle veya arama alternatiflerini kullanarak kazanım süresi boyunca problemin hızlı çözümü,
- Gereksiz verilerin elenmesi süresince problemi çözmek için ihtiyaç duyulan süreyi azaltma,
- Çözüm bulma kalitesinin düzeltilmesi
- Sistemin çözebildiği problemin sınıf veya setindeki artışlar (daha büyük örnekleri çözebilme yeteneği).

Bilgisayarla öğrenmede sonuç çıkarma için farklı yollar benimsenmiştir. Tümdengelim, tümevarım ve benzerlik oluşturma gibi ana çıkarım tipleri kullanılır.

Gözlem ve deneylerden öğrenme, insan öğrenme sürecinin en önemli özelliğidir. Aynı zamanda, öğrenme bilgi kazanımının en iyi yoludur. Bilgisayarla öğrenme araştırmalarının amacı; deney, gözlem ve olgu içeren örneklerden yeni bilgi elde etmek için bilgisayar programları geliştirmektir[5, 6].

Bilgisayarla öğrenme araştırmaları, bilgisayar alanındaki gelişmelere paralel olarak başladı ve bu alandaki ilk başarılar 1950 ile 1960 yılları arasında elde edildi. İzleyen yıllarda yürütülen kararlı çalışmaların sonucunda birçok metod ve algoritma geliştirildi[7].

1970'li yılların ortasında Michalski, endüktif öğrenme ve AQ[8] algoritmasını kanıtlamak için VL mantık sistemini sundu. Bu tarihten sonra AQ11 [9], CN2 [10,11] ve AQ15[12,99] algoritmalarını da içeren AQ ailesi algoritmaları geliştirilmiştir. AQ ailesi algoritmaları endüktif öğrenme alanındaki ana yaklaşımlarından birini temsil eder ve bu algoritmalar kapsama yaklaşımını kullanırlar.

Bir diğer yaklaşım ise Böl-ve-Fethet (Divide-and-Conquer) yaklaşımıdır. Bu yaklaşımı kullanan algoritmaların başında Quinlan tarafından geliştirilen ve entropi ölçüsünü kullanan ID3[13] öğrenme algoritması gelir. ID3 algoritması, ID3-IV, GID3 [14], ID5 [15], PRISM [16] ve C4.5 [17] endüktif öğrenme algoritmalarının

temelini teşkil eder. Bir diğer en iyi bilinen ve böl-ve-fethet yaklaşımını kullanan algoritma ise CART [18] algoritmasıdır. CART, özellikle sayısal değer içeren örnek setleri üzerinde işlem yapmak amacıyla geliştirilmiştir.

1982 yılında, Pawlak ‘Kaba-Set’ (Rough-Set) teorisini geliştirmiştir[19]. Kaba-Set, yapay zeka ve bilgisayarla öğrenme alanında kullanılabilen yeni bir yaklaşımdır. Özellikle eksik bilgiler ile işlem yapılması durumunda büyük avantaj sağlamaktadır. Kaba-Set, bilgi kazanımına farklı bir yaklaşım getirmiştir [3].

Bilgisayarla öğrenme alanındaki diğer başarılı metotlar olarak Minimum Hellinger Uzaklığı [20] ve bir sınıflandırma sistemi olan AutoClass [21] sayılabilir. Bilgisayarla öğrenmenin alanı, genetik algoritmaları ve yapay sinir ağlarını içine alacak şekilde genişletilebilir [22, 23].

Bilgisayarla öğrenme metotları genellikle sonuç çıkarmanın bazı çeşitlerini kullanırlar. Temel sonuç çıkarma teknikleri olarak tümdengelim, tümevarım ve benzerlik sayılabilir.

- **Tümdengelim:** Tümdengelim klasik mantık yürütme metodudur. A gerçekler kümesi ve  $A \models B$  çıkarım kuralı olsun.  $\models$  işareti doğruluk-korunumu veya yeniden formülasyonu temsil eder.  $A \models B$  ifadesi, B sonucunun A’dan çıkarılabileceğini gösterir. Tümdengelim bir tür gerçeklerin korunumu çıkarımıdır. Tümdengelim ile “Eğer A doğru ise B’de doğrudur” sonucuna varılır[25]. Birtakım yasa, ilke yada kurallardan hareket ederek özel bir olayı, durumu yada örnekleri inceleme ve açıklama yöntemidir. Herhangi bir ilkenin, kuralın yada formülün ne ölçüde gerçeğe uygun yada geçerli olduğunu araştırmada yararlanır.
- **Tümevarım:** Bu yanlışlık-korunumu çıkarım türüdür.  $A \Rightarrow B$  ile uygun olan  $h$  hipotezini bulmaya çalışır. Burada  $\Rightarrow$  işareti mantıksal gerektirmeyi ifade eder. Bu çoğunlukla gerçeklerden çıkarılan teorilerden oluşan süreci tanımlar [24, 26]. Özel durumlardan genel bir sonuca erişmek için yapılan araştırma yöntemidir. Bu yöntem örneklerden, sorunlardan, olaylardan ve özel

durumlardan hareket ederek genel sonuçlara, kurallara yada kanılara varmak için kullanılır.

- **Benzerlik:** Bu yanlışlık-korunumu çıkarım türü olup ancak tümevarımdan farklıdır. Benzerlik, varolan bilgi ile yeni bilginin karşılaştırılmasına dayanır ve örnek-tabanlı fikir yürütme ile ilgilidir[4].

Tümevarım bir parça bilgiden, mantıksal olarak bundan bağımsız olan bir sonuca varmak için akıl yürütmektir. Tümdengelim sonucunu ise tümevarımın aksine öneri ile sınırlıdır. Tümevarım, varolan bilginin ötesinde bilgi çıkarma yoludur. Tümevarım bu yüzden insan bilgisini genişletir.

## 2.2. Tümevarım ve Endüktif Öğrenme

Bu bölümde, tümevarım süreçleri ve endüktif öğrenme hakkında genel bilgiler verilecektir.

### 2.2.1. Tümevarım süreçleri

Temel tümevarım süreçlerinin aşağıda kısaca açıklanan yedi tipi vardır. Bunlardan ilk ikisi olan genelleme ve kaçırma, bilgisayarla öğrenmede çok yaygın olarak kullanılan metotlardır [24, 26, 27].

- **Kaçırma (Dışaçekim):** Bu tümevarımın temel tiplerinden biri olarak kabul edilir.  $A$  gerçekler kümesini dikkate alarak,  $A$  üzerinde bazı  $\{h\}$  hipotezleri oluşturur. Bundan dolayı  $A \Rightarrow h$  yazılabilir. Bu tip tarafından birçok mümkün sonuç bulunabilir. Buna karşın, tümdengelim tarafından sadece bir sonuç bulunabilir. Genellikle, gerçeklerin özelliklerine göre anlamlı sonuçlardan sadece biri seçilir[3, 27].
- **Genelleme:** Bu, çeşitli gerçekler arasında çok genel özellikleri bulan bir süreçtir.  $A1$  ve  $A2$  gibi iki gerçek olduğu kabul edilsin. Genelleme süreci,  $\sigma A1=A2$  olsun diye  $\sigma$  yerine geçici bir ifade bulur[26, 28, 29, 30]. Genelleme

karmaşık bir bilgi gösterim mekanizmasıdır. Bilgiyi ağaç şeklinde gösterir ve ağacın üst tarafındaki düğümleri en aza indirerek daha genel bir yapı elde edilmesini sağlar. Genelleme bilgisayarla öğrenmede yaygın olarak kullanılan bir bilgi gösterim türüdür.

- **Nedensellik belirleme:** Aynı anda birkaç olgunun ortaya çıkması halinde, diğerlerinin nedenini belirten bir hipotez bulmak için benimsenmiştir [26]. Örneğin, kullanılan takımında bir aşınma varsa işlenen parçanın yüzey hassasiyeti düşecektir.
- **Özellikleri niteleme:** Bu bir çeşit kavram analizidir. Gerçeklerin bir kümesine dayanır ve sette bulunan gerçeklerin biri kavram ve diğerleri ise kavramın özellikleri olarak kabul edilir. Bu yüzden bilgi, kavramlar ve bunların özellikleri tarafından temsil edilir [3].
- **Şekilsel ve geçici kapsama :** Geçici olarak kapsanmış  $X$  bölgesinde meydana gelen her olay için, eğer  $Y, X'$ 'i içeriyorsa bunlar  $Y$  bölgesinde de oluyordur.  $X$  ve  $Y$  iki set,  $X \subset Y$  olarak kabul edilsin. Eğer bu olay  $X$ 'de oluyorsa aynı şekilde  $Y$ 'de de oluyordur. Bu;  $\forall x[olay(x, x \in X) \Rightarrow olay(x, x \in Y)]$  olarak ifade edilir [27].
- **Kümeleme:** Bu bir veya birkaç belirli özellikler ile beraber düşük-seviyeli bilginin birkaç parçasını bağlar. Kümelenmiş bilgi, orijinal bilginin ortak özelliklerini gösteren yüksek-seviyeli bilgidir [24, 28].
- **Sayısal tümevarım:** Bu istatistiksel metotlarla ilgilidir [26].  $X$  bir set ve  $X'$  ise  $X$ 'in bir alt seti olsun ( $X' \subset X$ ). Sayısal tümevarım,  $X', X$  için istatistiksel anlamlı olması durumunda temeldir. Bu durumda  $X'$  'den çıkarılan  $h$  sonucu  $X$  için geçerlidir.

Bilgi kazanımı görüşü noktasında ilk iki süreç bilgiyi keşfetmek için daha uygun iken, diğerleri bilgiyi organize veya yapılandırmak için daha uygundur.

### 2.2.2. Endüktif öğrenme

Endüktif öğrenme, bir örnek setinden genel bilgi çıkarma yeteneğidir[31]. Örnek setini oluşturan örnekler deney, gözlem veya olaylardan elde edilebilir. Endüktif öğrenme otomatik bilgi kazanımı için pratik yaklaşımlardan birisi olarak kabul edilir. Son yıllarda araştırmacılar endüktif öğrenilebilirliğin teorisi üzerinde çalışmışlardır. Onlar temel teorik analizler üzerinde endüktif öğrenmenin mümkün olabilirliğini kanıtlamayı başarmışlardır. 1971 yılında Vapnik ve Chervonenkis öğrenilebilmenin bir ölçüsü olarak VC-boyutu'nu sunmuşlardır [32]. Daha sonra öğrenilebilme alanında birçok teorik çalışma yapılmıştır [33-40].

Eğitme örneklerinden bir kavramı açıklayan bilgiler elde etme endüktif öğrenme işlemidir. Tümevarımın tabiatından dolayı öğrenme örneklerinden karar kuralları üretmenin mümkün birçok yolları vardır. Tekli varsayımları üreten algoritmalar, öğrenilecek kavramın en iyi tanımının ne olduğuna karar vermek için mekanizmalara ihtiyaç duyarlar. Çünkü, örnek setindeki bir örnek tek başına bir kavramı açıklamak için yetersiz kalmaktadır[41].

Öğrenme algoritmalarının tasarımındaki en büyük problem düzensiz örneklerden aşırı derecede karmaşık tanımlamaları üretmekten sakınmanın nasıl olacağıdır. Bozuk, düzensiz verilerden öğrenme, çok sayıda verimsiz örnekler tanımlama, karmaşık karar kurallarının çok sayıda olmasına sebep olabilir. Bu durumda üretilen kavram tanımı genellikle eksik olmaktadır [41].

Genel olarak endüktif öğrenme aşağıdaki gibi tanımlanabilir [9, 42]:

Verilenler:

- Bazı deney veya durumlar hakkında bilgiyi gösteren bir örnek seti ( $A$ ).
- Geçici/kesin olmayan endüktif iddia (boş olabilir).
- Geçmiş bilgi  $K$ .

Bulunanlar:

- Örnekleri zayıf olarak ifade ve geçmiş  $K$  bilgisini temsil eden bir  $h$  hipotezi. Bunun gösterimi  $h \wedge K \Rightarrow A$  şeklindedir. Bu ifade,  $h$  hipotezi ve geçmiş  $K$  bilgisinin,  $A$  örnek setini ifade ettiğini gösterir. Burada  $\wedge$  mantıksal çıktıyı gösterir.

$A$ 'nın yapısına bağlı olarak endüktif öğrenmenin iki mümkün durumu vardır. Bunlar: danışmanlı ve danışmansız.  $A$ 'daki örnekler önceden sınıflandırılmışsa bu öğrenmeye danışmanlı, diğer durumda danışmansız olarak adlandırılır.

Bir  $A$  örnek setinde  $e$  adet örnek ve  $s$  adet  $\{d_1, d_2, \dots, d_s\}$  farklı sınıf olsun.  $e(d_i)$ ,  $d_i$  sınıfına ait bir örneği gösterebilir.  $A$ 'dan bilgi veya kavram öğrenme;

$$h: \{Des(d_i)\} \quad 1 \leq i \leq s$$

olarak gösterilir. Burada  $Des(d_i)$ ,  $d_i$  sınıfı kavramın tanımıdır.

Herhangi bir endüktif öğrenme prosedürü tarafından tahmin edilebilen iki şart vardır. Bunlar, eksiksizlik şartı ve tutarlılık şartıdır.

Eksiksizlik:

$$\forall e(d_i) \in A (Des(d_i) \Rightarrow e(d_i))$$

Bu şart ile bütün örneklerde,  $d_i$  sınıfı  $Des(d_i)$  kavramı tarafından kapsanmalıdır.

Tutarlılık:

$$\forall e(d_i), e(d_j) \in A (e(d_j) \Rightarrow \approx Des(d_i)) \text{ eğer } j \neq i$$

Bu şart endüktif öğrenmede çok önemlidir.  $Des(d_i)$  kavramı,  $d_i$  sınıfına ait olmayan herhangi bir örneği kapsamamalıdır.

Bir öğrenme işleminde, endüktif öğrenme ile birkaç mümkün çıktı elde edilmiş olacaktır.  $H$  bütün mümkün çıktıları içeren bir hipotez uzayı olsun[43]. Endüktif öğrenme, bir  $h$  hipotezi bulmak için  $H$  hipotez uzayında bir arama prosedürü olarak açıklanabilir. Arama işlemini yaparken eksiksizlik ve tutarlılık şartlarını göz önünde tutarak aynı zamanda  $K$  geçmiş bilgisini de tahmin eder.

$U$  örnekler uzayı olsun.  $U$ daki örnekler bir şart karakteristik seti  $C=\{C_1, C_2, \dots, C_n\}$  ve bir  $D$  karar karakteristiği tarafından tanımlanır. Şart karakteristiklerinin bilgi alanı  $V$  tarafından verilir ( $V=\{V_1, V_2, \dots, V_n\}$ ). Herbir  $C_i \in C$  olan  $C_i$  şart karakteristiğinin bilgi alanı  $V_i$  değer setidir.  $V_i$ ,  $V_i \in V$  olan gözlemlenebilir bir değer setidir.  $H$ 'nin boyutu veya mümkün hipotezlerin sayısı;

$$|H| = \sum_{i=1 \dots n} |V_i| + \sum_{\substack{i,j=1 \dots n \\ i \neq j}} |V_i \parallel V_j| + \dots + \sum_{\substack{i,j,\dots,k=1 \dots n \\ i \neq j \neq \dots \neq k}} |V_i \parallel V_j \parallel V_k| + \prod_{i=1 \dots n} |V_i|$$

olarak hesaplanır. Şart karakteristikleri ikili değerlere sahip ise  $|H|$ ;

$$|H| = \sum_{i=1 \dots n} C_n^i 2^i \quad \text{olur.}$$

Burada açıkça görülüyor ki,  $n$  çok büyük bir değer olduğunda bütün hipotez uzayını araştırmak pratik hatta mümkün değildir.

Arama metodu, endüktif öğrenme prosedürleri için çok önemlidir. Genellikle ‘açgözlü’ (greedy) arama tekniği kullanılır[13, 43, 44]. Bu teknik 1987 yılında geliştirilmiştir[46]. Buna alternatif olarak 1988 ve 1990 yıllarında hipotez uzayını mümkün olan en az seviyeye indiren birer metod geliştirilmiştir[41, 35].

### 2.3. Endüktif Öğrenme Algoritmaları

Endüktif öğrenmenin sonuçları bir çeşit bilgidir. Bu bilgi ya karar ağacı yada kural seti şeklinde gösterilebilir.

Böl-ve-fethet yaklaşımını kullanan endüktif öğrenme algoritmaları bilgiyi bir karar ağacı şeklinde gösterirler. Bu yaklaşım, karar ağacı kullanımından dolayı gereğinden fazla tanımlama içerebilir. Genellikle sonuçlar basitleştirilmiş şekilde değildir. Karar ağacı ile her ne kadar daima doğru olmasa da bir tek sınıflama veya tahmin elde edilebilir.

Kapsama yaklaşımını kullanan endüktif öğrenme algoritmaları bilgiyi doğrudan bir kural seti şeklinde gösterirler. Bazen, bir kural seti bir tek sınıflama veya tahmini vermeyebilir. Bu durumda, verilmiş olan her bir cevap güvenilir derecededir.

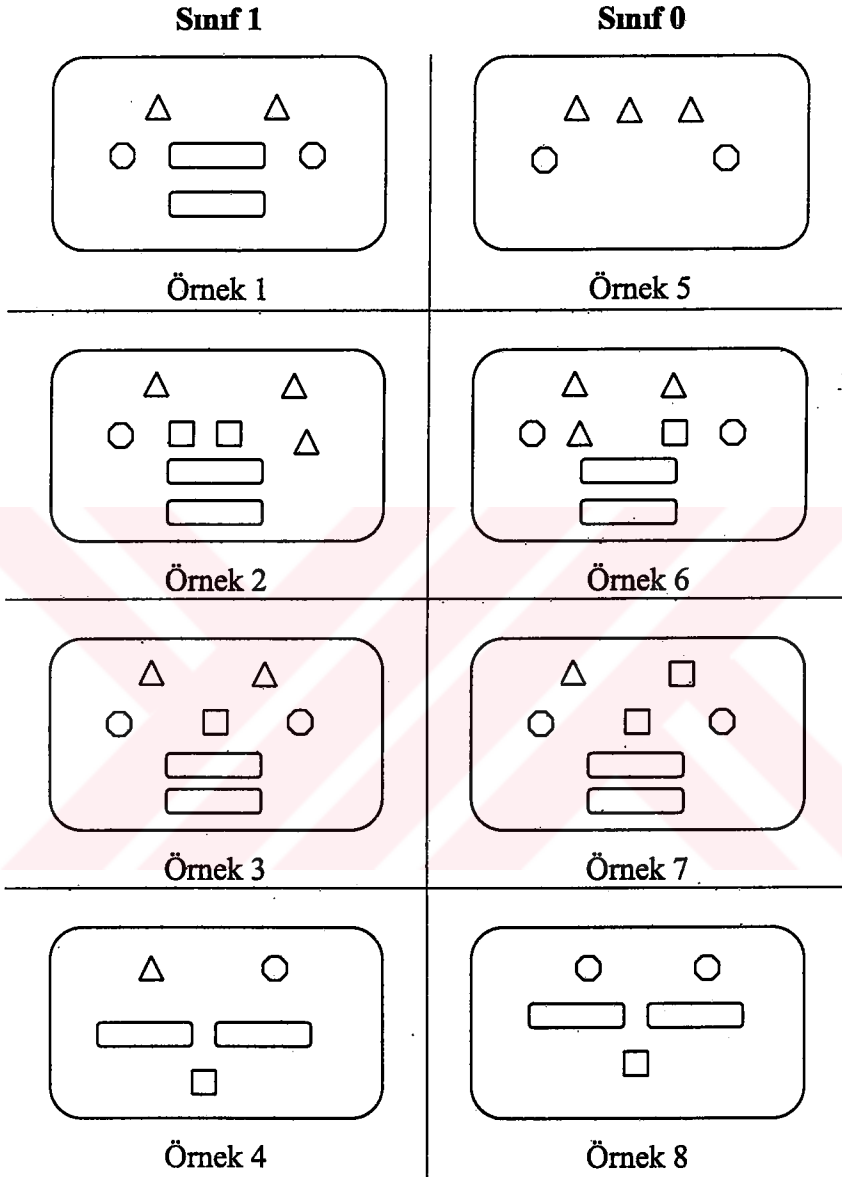
Genellikle bir karar ağacı, bir kural setine dönüştürülebilir ama dönüşüm daima doğru değildir.

Öğrenme sonuçları ya kesin yada tahmini sonuçlar olarak bir karar ağacı veya bir kural seti şeklinde sınıflandırılarak gösterilebilir. Kesin bir sonuç, bir öğrenme seti kullanılarak test edildiğinde hatasız olduğu görülür. Karar ağacı veya kural seti test örnekleri üzerinde kullanıldığında tahmini bir hatanın olamayacağı garanti değildir.

Bazen, kesin çözümler bulmak için bu çok zor veya gerekli değildir. Bu durumda, kesin sonuçlar yerine tahmini sonuçlar bulunur. Bir tahmini sonuç örnekler seti üzerinde test edildiği zaman hatalar oluşacaktır. Fakat bunlar bir eşik değerini aşmazsa, örneğin %5 gibi, çözüm kabul edilebilir olarak alınacaktır.

Karar ağacı ve kural seti kavramları Şekil 2.1'de gösterilen iki sınıflı sınıflandırma problemi için açıklanmıştır. Şekil 2.1'deki örneklerin şekilsel gösterimlerini, endüktif öğrenme prosedürlerindeki gösterimini kolaylaştırmak için Tablo 2.1'de gösterildiği gibi karakteristik-değer şekline dönüştürülmüştür. Tablo 2.1'deki örnekler dört şart karakteristiği ( $C_1, C_2, C_3, C_4$ ) ve bir karar karakteristiği (Sınıf) tarafından tanımlanmıştır. Her bir şart karakteristiği bir geometrik şekle karşılık gelmektedir. Örneğin,  $C_1$  daireyi (  $\bigcirc$  ),  $C_2$  uzun çubuğu (  $\text{—}$  ),  $C_3$  üçgeni (  $\triangle$  ) ve  $C_4$  ise kareyi (  $\square$  ) göstermektedir. Bir şart karakteristiğinin değeri ( $C_i$ ),  $C_i$  için ilgili şeklin birkaç elemanını gösterir.

Tablo 2.1'deki her bir satır bir örneği gösterir. Örneğin, ikinci satır  $\{C_1=1, C_2=2, C_3=3, C_4=2, Sınıf=1\}$  Şekil 2.1'deki ilk sınıfın ikinci örneğini gösterir. Burada; 1 adet daire, 2 adet uzun çubuk, 3 adet üçgen ve 2 adet kare bulunmaktadır.



Şekil 2.1. Örnek seti [3]

Şekil 2.2 örnek setinden kural seti ve karar ağacı çıkarımını göstermektedir. Kural setindeki ilk kural şöyle yorumlanabilir:

**Eğer 2 üçgen var İse 1 nolu sınıfa aittir**

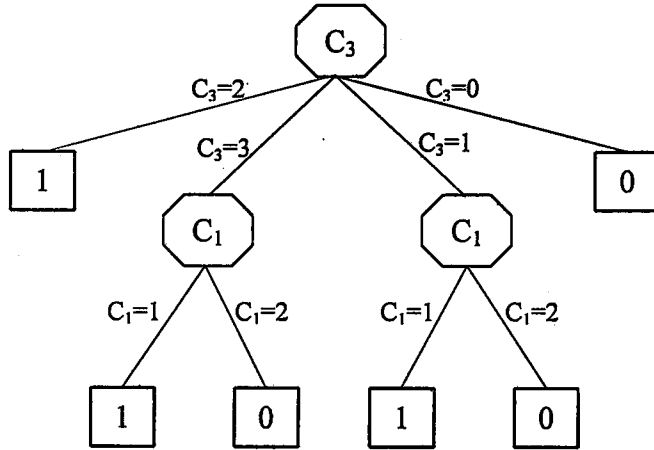
Not: Karar ağacı 1 nolu sınıf için gereksiz tanımlamalar içerir ve buna karşın çıkarılan kural setinden biraz daha basittir.

Tablo 2.1. Şekil 2.1'deki örnekler setinin karakteristik-değer biçimi [3]

| Örnek No | C <sub>1</sub> | C <sub>2</sub> | C <sub>3</sub> | C <sub>4</sub> | Sınıf |
|----------|----------------|----------------|----------------|----------------|-------|
| 1        | 2              | 2              | 2              | 0              | 1     |
| 2        | 1              | 2              | 3              | 2              | 1     |
| 3        | 2              | 2              | 2              | 1              | 1     |
| 4        | 1              | 2              | 1              | 1              | 1     |
| 5        | 2              | 0              | 3              | 0              | 0     |
| 6        | 2              | 2              | 3              | 1              | 0     |
| 7        | 2              | 2              | 1              | 2              | 0     |
| 8        | 2              | 2              | 0              | 1              | 0     |

Kural 1 : Eğer C<sub>3</sub> = 2 ise Sınıf = 1  
 Kural 2 : Eğer C<sub>1</sub> = 1 ise Sınıf = 1  
 Kural 3 : Eğer C<sub>1</sub>=2 ve C<sub>3</sub>=3 ise Sınıf = 0  
 Kural 4 : Eğer C<sub>1</sub>=2 ve C<sub>3</sub>=1 ise Sınıf = 0  
 Kural 5 : Eğer C<sub>3</sub>=0 ise Sınıf = 0

Kural seti (REX-1 algoritması ile elde edilmiştir)



Karar ağacı (C4.5 algoritması ile elde edilmiştir)

Şekil 2.2. Şekil 2.1'deki örnek setinden üretilen kural seti ve karar ağacı[3]

### 2.3.1. Endüktif öğrenme algoritmalarının sınıflandırılması

Bir kavramın temel karakteristiklerini tanımlamak için yeterince genel kurallar üretmeye izin verecek ve eğitime örneklerinin tutarsız sınıflandırılmasını önleyecek yaklaşımlar vardır. Bu yaklaşımlardan biri karar ağacı oluşturarak kural şartlarının ve kuralların çıkarılmasını sağlar. Bu amaçla geliştirilen ID ailesi algoritmaları ve bunların devamı olan C4.5[17] algoritması bulunmaktadır. Bunlardan ikincisi, çok büyük sayıdaki örnekleri kapsayan, bir kavramın en genel karakteristiklerini dikkate alan ve oluşan gereksiz kuralları atan yaklaşımdır. Bu teknikleri kullanan algoritmalara AQ ailesi ve RULES ailesi algoritmaları örnek olarak verilebilir[7, 9]. Bu algoritmalar, özel prosedürler yardımıyla bir sınıfı diğer sınıflardan ayıran örneklerdeki değerlerin bir setini kullanarak kavramı en iyi tanıtan, en genel kuralları üreten deneysel öğrenme algoritmalarıdır. Bu yaklaşımların uygulanması sırasında çok karmaşık prosedürlere ihtiyaç duyulması, yüksek miktarda hafıza gerektirmesi ve uzun işlem zamanı gerektirmesinden dolayı bazı problemlerle karşılaşılır.

Endüktif öğrenme için diğer bir yaklaşım ise tümevarım ve mantık programlamanın bir birleşimi olan Endüktif Mantık Programlama'dır. Bu yaklaşım, geleneksel endüktif öğrenme yaklaşımlarından farklı olarak yeni kavramları göstermek ve örnekleri tanımlamak için mantık önermeyi kullanır. Endüktif mantık programlama yeni bir kavramı, geçmiş bilgiyi ve eğitime örneklerini göstermek için sonucun daha güçlü mantık kurallarına dayandırılmasını sağlar. Bu metot eğitime örneklerinin ve geçmiş bilginin farklı şekillerinin kullanımına izin verir. Bu metodun iki tipi vardır. Birincisi, yukarıdan aşağıya doğru genelleme/özelleştirme metodudur. İkincisi, ters kararlılık ilkesi metodudur[85].

#### 2.3.1.1. Karar ağacı tabanlı algoritmalar

Karar ağacı-tabanlı algoritmalar genellikle, karar ağacını oluşturmak için en büyük bilgi kazancını veren özelliklerden arama yaparak bilgi entropi ölçüsünü kullanırlar [3, 4]. Bu amaçla karar ağacını oluşturmak için örnekler setini küçük alt setlere

bölerler. Karar ağacı geliştirme prosedürü, eğitime örneklerini kullanıcının belirlediği sonlandırma ölçütüne göre doğru sınıflandırılıncaya kadar devam eder.

Karar ağacı-tabanlı algoritmalar Böl-ve-Fethet metoduna göre işlem yaparak örnekleri alt setlere ayırırlar. Böl-ve-Fethet endüktif öğrenme metodunun metodunun ana fikri, bir örnek setini, her bir alt sette sadece tek bir sınıf kalıncaya kadar birkaç alt sete ayırır. Bu metodun sonucu bir karar ağacı şeklindedir. Genel Böl-ve-Fethet endüktif öğrenme prosedürü aşağıdaki gibidir:

1. Verilen  $S$  örnek seti,  $S_{11}, S_{12}, \dots, S_{1n}$  şeklinde alt setlere ayrıştırılır. Bu;

$$S = \bigcup_i S_{1i} \text{ and } \bigcap_i S_{1i} = \emptyset \text{ şeklinde ifade edilir.}$$

2. Her bir  $S_{1i}$  alt seti için,  $S_{1i}$ 'de tek bir sınıf varsa ayrıştırma işlemi tamamlanır ve buraya bir sınıf etiketi olarak düğüm konur.

Şekil 2.3'te ayrıştırma prosedürü gösterilmiştir. Burada, bir sınıfı göstermek üzere düğüm olarak dikdörtgen şekli kullanılmıştır. Prosedürün başlangıç noktası en düşük ayrıştırma seviyesini gösterir. Bir düşük ayrıştırma seviyesinde bir düğüm görülebilir. Bunlar yüksek seviyelerden daha geneldir. Ayrıca aynı ayrıştırma seviyesinde, 'çocuk' düğümü ile onların 'ebeveyn' düğümü arasındaki ilişkiler için;

$$S_{(1,i) \dots (l,j)} = \bigcup_k S_{(1,i) \dots (l,j)(l+1,k)}$$

ve

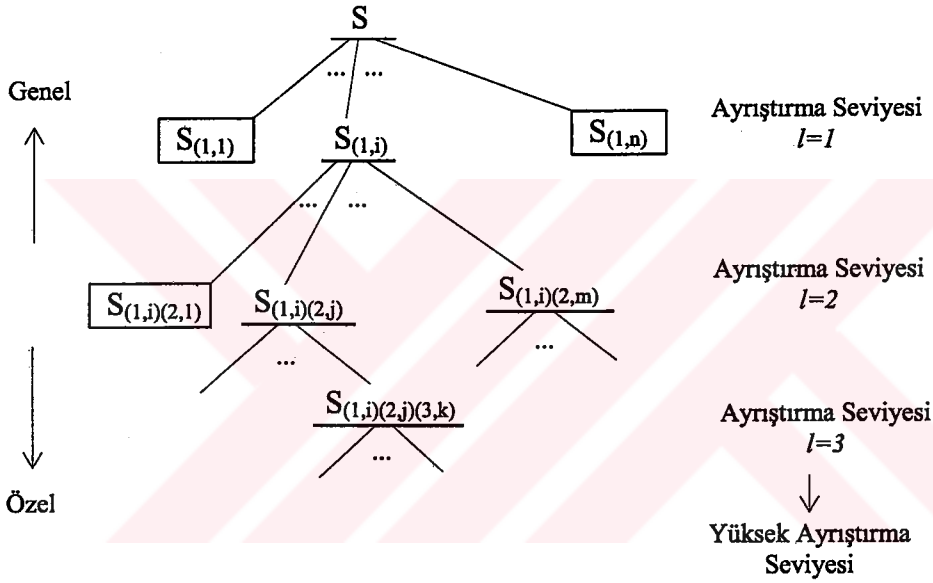
$$\bigcap_k S_{(1,i) \dots (l,j)(l+1,k)} = \emptyset$$

ifadeleri yazılabilir. Buradaki  $l$  ayrıştırma seviyesini gösterir.

$S=(C,D,V)$  bir örnek setini gösterir. Bu örnek setindeki elemanlar sırasıyla;  $C=\{C1,C2,\dots,Cn\}$  karakteristik setini,  $V=\{V1,V2,\dots,Vn\}$  değerler setini ve  $D=\{d1,d2,\dots,dn\}$  ise karar karakteristiklerini (sınıf) setini oluşturur. Tablo 2.1'de

verilen örnek seti, Böl-ve-Fethet endüktif öğrenme metoduna göre çok farklı alt setlere ayrıştırılabilir. Bu örnek seti;  $C=\{C1,C2,C3,C4\}$ ,  $V=\{V1,V2,V3,V4\}=\{\{1,2\}, \{0,1,2\}, \{0,1,2,3\}, \{0,1,2\}\}$  ve  $D=\{0,1\}$  bilgilerinden oluşmuştur. Bu bilgilere göre 239 mümkün alt set oluşturulabilir. Bu metot ile her bir düğümdeki ayrıştırmalar optimize edilerek verimli ve kabul edilebilir bir yapı elde edilebilir.

Karar ağaçları oluşturulduktan sonra, ağacın her dalı bir kural oluşturacak şekilde kurallara dönüştürülür. Ancak karar ağacını karar kurallarına çevirmenin bazı dezavantajları vardır[120]. Karar ağaçları ile işlem yapmak oldukça zordur.



Şekil 2.3. Böl-ve-Fethet endüktif öğrenme metodunda ayrıştırma prosedürü

Herhangi bir sınıflandırma hakkında bilgi çıkarmak bütün ağacın gözden geçirilmesini gerektirir. Bu problem çözmek, karar ağacını kurallara dönüştürmekle mümkündür. Ancak ağaçlar tarafından gösterilemeyecek kurallar da vardır. Örnek olarak, aşağıdaki kural seti verilsin.

Kural 1: EĞER  $a = 1$  VE  $b = 1$  İSE Sınıf = 1

Kural 2: EĞER  $c = 1$  VE  $d = 1$  İSE Sınıf = 1

Kural-1 ve Kural-2'nin Sınıf 1'i kapsadığını ve diğer tüm kuralların ise Sınıf 2'ye ait olduğu kabul edildiğinde, her iki kural tek bir ağaç tarafından gösterilemez. Çünkü ağacın kökü bir karakteristiğe göre ayrılır ve her iki kurala da ait olan bir karakteristik yoktur. Bu durumu göstermek için verilen kurallara bir şartın fazladan eklenmesi gerekecektir. Bu da sonuçta fazladan en az bir kural daha çıkmasına sebep olacaktır [119].

Karar ağacı üreten algoritmaların listesi Tablo 2.2'de verilmiştir.

Tablo 2.2. Karar ağacı üreten algoritmalar

| Algoritmanın Adı     | Algoritmayı Geliştiren(ler)                              | Yılı |
|----------------------|----------------------------------------------------------|------|
| CLS[1-9]             | HUNT E. B., MARIN J., STONE P. J.                        | 1966 |
| ID3                  | QUINLAN J. R.                                            | 1983 |
| ID3-IV, GID3         | CHENG J., FAYYAD U.M., IRANI K.B., QIAN Z.,              | 1988 |
| ID5, ID5R, ID5R-hat  | UTGOFF, P.E.                                             | 1989 |
| ID4                  | SCHLIMMER J.C., FISHER D.H.                              | 1986 |
| C4.5                 | QUINLAN J. R.                                            | 1993 |
| C4.5Rules, C4.5Prune | QUINLAN J. R.                                            | 1995 |
| CART                 | BREIMAN L., FRIEDMAN J.H., OLSHEN R. A.,<br>STONE C. J., | 1984 |
| OC1                  | CRAWFORD, S.L.,                                          | 1990 |
| PRISM                | CENDROWSKA, J.                                           | 1988 |
| J-Pruned             | BRAMER, M.A.                                             | 2002 |

Karar ağacını kullanan ilk algoritma, Hunt [47] tarafından geliştirilen CLS (Concept Learning Sytem) algoritmasıdır. CLS serisi algoritmaların toplam sayısı dokuzdur. Bunlardan ilk sekizi sadece iki sınıflı örnek setleri ile çalışabilirken, CLS9 algoritmasında çok sınıflı örnekler çözülebilmektedir.

Daha sonra bu metodu kullanan ID3 ailesi algoritmalarının büyük bir kısmı Quinlan tarafından geliştirilmiştir (ID3, ASSISTANT ve C4.5).

1983 yılında, Quinlan CLS algoritmasının soyundan gelen ID3 endüktif öğrenme algoritmasını geliştirdi[13]. ID3 algoritmasında, ayırtırmaya rehberlik edecek bir bilgi entropi ölçüsü kullanılmıştır. Bilginin entropisi (2.1) ve (2.2) eşitlikleri yardımıyla hesaplanır.

$$I(S) = -\sum_i p(D = d_i) \log_2 p(D = d_i) \quad (2.1)$$

$$E(C_i, S) = \sum_{\substack{C_i = v_{jk} \\ v_{jk} \in V_i}} \frac{|S_j|}{|S|} I(S_j) \quad (2.2)$$

$I(S)$ ,  $S$  örnek setinin bilgi değerini gösterir.  $E(C_i, S)$ ,  $C_i$  sınıfındaki örneklerle ait bilginin entropisini gösterir. Bilgi kazancı ise (2.3) eşitliği ile hesaplanır.

$$Kazanç(C_i, S) = I(S) - E(C_i, S) \quad (2.3)$$

ID3 algoritması bilgi kazancını maksimize eden bir düğümü en uygun düğüm olarak seçer. Her düğüm için ilgili alt setin bilgi kazancı hesaplanarak bu düğüme göre alt setlere ayrıştırma işlemine her düğümde tek sınıf kalıncaya kadar devam edilir.

Daha sonra, ID3 endüktif öğrenme algoritmalarının bir serisi geliştirildi. Bu seri ID3 ailesi algoritmaları olarak adlandırılır[49]. ID3 algoritmasının versiyonları, ID3-IV[14], GID3[14], ID4 [50], ID5[15], ID5R ve ID5R-hat[51] olarak sayılabilir. C4.5 algoritması ID3 algoritmasının endüstriyel versiyonudur [17]. C4.5 algoritmasının en büyük özelliği karar ağacının gereksiz dallarını tespit ederek budamasıdır. C4.5 algoritmasının bir diğer özelliği ise eksik veri ve sayısal değerlerle çalışabilmesidir.

1984 yılında Breiman tarafından CART (Classification and Regression Trees) isimli algoritma geliştirilmiştir [18]. CART sayısal verileri işlemek üzere tasarlanmıştır. Bu algoritma Böl-ve-Fethet endüktif öğrenme metoduna göre işlem yapmakta ve sadece ikili karar ağacı oluşturmaktaydı.

CART, her düğümündeki tüm karakteristikleri birer birer araştırır. Her bir karakteristik için, en çok katkıyı sağlayan en iyi ayrıştırmayı bulur.  $n$  adet aday içerisinde en iyi ayrıştırmayı seçer.

Daha sonra karar ağacının kısa olmasını sağlamak için Crawford tarafından CART algoritmasının uzantısı olan OC1 algoritması geliştirilmiştir[52]. Karar ağacını yatık hale getirmek için yeni bir algoritma daha geliştirilmiştir [53].

### 2.3.1.2. Kural tabanlı algoritmalar

Kural-tabanlı algoritmalar kapsama metodunu kullanırlar. Bu metot verilen örnek setinden daha genel kurallar elde etmek için örnek setini sınıflara ayırır.

Kapsama metodu verilen örnek setine dayalı bir kural uzayı hipotezi kurar. Bu kural uzayı hipotezi, arama işlemleri boyunca elimine edilerek en genel kurallar bulunmaya çalışılır. Kural uzayı hipotezi *versiyon uzayı* olarak adlandırılır.

İlk olarak Mitchell, kapsama metodunu kullanarak aday-eleme algoritmasını geliştirmiştir[47]. Bu metodu kullanan algoritmalar aşağıdaki mantığa göre işlem yaparlar.

$S$  bir örnek seti olsun. Örnekler sınıflarına göre ya pozitif yada negatif olarak sınıflandırılırlar. Seçilen örnek ile aynı sınıfta olan örnekler pozitif olarak kabul edilirken, diğer tüm sınıflara ait örnekler negatif olarak dikkate alınır.  $H$  versiyon uzayını gösterebilir. Başlangıçta  $H$ , verilen pozitif örneklerdeki tüm mümkün kavramları kapsar. Bu durumda gösterilen örneklerdeki gereksiz aday kavramlar  $H$ 'den elenerek,  $H$ 'de sadece bir sınıfa ait kavramların kalması sağlanır. Eleme işlemi şöyle yapılır: Pozitif örneklerde gösterilen ve  $H$ 'yı genelleyen tüm özel kavram tanımlamaları  $H$ 'den silinir. Bu işlem negatif sınıflar üzerinde de tekrarlanarak, çok genel kavram tanımlamaları elde edilir. Bu yolla  $H$ , sadece istenen kavram tanımlamaları kalıncaya kadar kademeli olarak küçültülür. Öğrenilen kavramlar '*Eğer tanım ise karar*' şeklinde gösterilir.

Bu algoritma verilen örnekler setinden çok sayıda kabul edilebilir kavramı bulabilir. Ancak, çok büyük örnek seti için başlangıçta  $H$ 'ı oluşturmak çok zor, hatta bazı durumlarda imkansızdır.

$S$  bir örnek seti olduğuna göre, pozitif set ( $S^+$ ) ve negatif set ( $S^-$ ) olmak üzere bölünebilir.  $S^+$ 'daki örneklerin hepsi aynı sınıfa aittir ve  $S^+ \cup S^- = S$  ve  $S^+ \cap S^- = \emptyset$  olur.

AQ ailesi algoritmaları kapsama metodunu kullanarak Michalski tarafından geliştirilmişlerdir[8]. Kullanılan yöntem aday-eleme algoritmalarındaki ile benzerdir. Farklı olarak başlangıçta oluşturulan  $H$ 'nın, sadece boş tanımlamaları içermesidir.

Doğrudan kural üreten algoritmaların listesi Tablo 2.3'te verilmiştir.

Tablo 2.3. Doğrudan kural üreten algoritmalar

| Algoritmanın Adı | Algoritmayı Geliştiren(ler)                     | Yılı      |
|------------------|-------------------------------------------------|-----------|
| CN2              | CLARK P., NIBLETT T.,                           | 1989      |
| RULES-1          | PHAM, D.T., AKSOY, M.S.                         | 1993      |
| RULES-2          | PHAM, D.T., AKSOY, M.S.                         | 1993      |
| RULES-3          | PHAM, D.T., AKSOY, M.S.                         | 1995      |
| RULES-3 PLUS     | PHAM D. T., DIMOV S. S.                         | 1997      |
| RULES-4          | PHAM D. T., DIMOV S. S.                         | 1997      |
| AQ Ailesi        | MICHALSKI R.S.                                  | 1983-2001 |
| ILA-1            | TOLUN M. R., ABU-SOUD S.M.                      | 1998      |
| ILA-2            | TOLUN M. R., SEVER H., ULUDAĞ M., ABU-SOUD S.M. | 1998      |
| GDT-NR, GDT-RS   | ZHONG N., DONG J., OHSUGA S.                    | 2001      |

## 2.4. Endüktif Öğrenme Algoritmalarının Uygulaması

Endüktif öğrenme uygulamalarda geniş bir kullanım alanı bulmaktadır. Çok başarılı uygulama alanlardan ikisi [67]'de tanıtılmıştır. Bu uygulamalardan birincisi, 1986 yılında BP tarafından GASOIL uzman sisteminin geliştirilmesidir. Bu sistem, gaz ve petrolü ayırtmak için tasarlanmıştır. Bu sistemde endüktif öğrenme kullanılarak birtakım örneklerden yaklaşık olarak 2,500 kural çıkarılmıştır. GASOIL zaman ve maliyetlerde büyük tasarruflar sağlamıştır. Diğer uygulama ise uçuş için tasarlanmıştır. Bu sistemde, insan pilotun vasıflarını öğrenen bir uçuş similatörü geliştirilmiştir. Pilotun eylemlerinden 90,000 örnek oluşturulmuştur. Quinlan'ın C4.5 algoritması kullanılarak uçuş sırasında similatörün hareketlerine rehberlik edecek

olan karar ağacı hazırlanmıştır. Similatör, eylem ve hareketleri insan pilotlardan daha iyi öğrenebiliyordu. Çünkü, similatörün pilotlar tarafından zaman zaman yapılan aynı hataları tekrarlaması önlenmişti.

Endüktif öğrenme, oyun problemlerinde de yaygın olarak kullanılmaktadır. Özellikle satranç gibi bilgi-tabanlı oyunlar için uygulamalar geliştirilmiştir [13, 68].

Bu çalışmaları, endüstride endüktif öğrenme tekniklerinin uygulaması izledi. Örneğin, bir endüktif öğrenme tekniği ile yarı iletken yongaların imalatında sınıflandırma hatalarına uygulanmıştır[69, 55]. Diğer endüstriyel uygulamalar, kolay kırılabilen malzemeleri teşhis ederek sınıflandırma[70], dönen bir tuğla fırının kontrolü[64], bir makine işleme similatöründen kurallar çıkarma[71], sonlu-elemanlar ağ tasarımı[73], karakter tanıma [48, 74], dinamik sistem tanımlama [75, 76] ve kalite kontrolde hataların bilgisayar tarafından bulunması ve sınıflandırılmasında[85], zeki proses kontrolünde[54], yarı iletken ve renkli-CRT imalat proseslerinde bir zeki proses kontrol sistemi olarak ve üretim prosesinin ayarlanmasında[54], istatistiksel kalite kontrolde[56], görüntü tanımda[48], makine hatalarının belirlenmesinde uzman sistem ve bütünlük veritabanı hazırlanmasında [72, 78] kullanılmaktadır.

Bir diğer endüktif öğrenme uygulaması ise NASA'nın uzay iletişim ağında hata bulmak için kullanılmıştır[79]. Öğrenilmiş bilgi, karmaşık askeri sistemlerin bakımı için en uygun kararı vermek için bir uzman sistemle birleştirilmiştir. Sonuç olarak, bakım maliyetleri en alt seviyeye indirgenmiştir.

Tıp alanında endüktif öğrenme uygulamalarına çok rastlanmaktadır. Bağımsız bir bilgi kazanım sistemi olan ENIGMA, örneklerden kural çıkarmak için tasarlanmış ve elde edilen kuralları kullanarak otomatik olarak teşhis koyabilen bir uzman sistem meydana getirilmiştir[77]. Ayrıca, hastaların belirtilerinden bilgi kazanımı için tasarlanmış ve LEARN olarak adlandırılan bir program geliştirilmiştir[80]. Geliştirilen programlar tıp alanında birçok alanda uygulanmaktadır: Göğüs kanseri ve yinelenmesinin tahmini[57, 58], başlıca tümör yerlerinin tahmininde[12] ve

yüksek derecede seçici özelliğine sahip (HSV) cihazlar kullanılarak ince bağırsak ülseri ile ilgili elde edilen sonuçların analizinde[65].

Ticari alanda çok başarılı bir endüktif öğrenme uygulaması ise kredi kartı için başvuran müşterilerin kredilendirme değerlendirmelerinde kullanılmasıdır[17].

Ayrıca, lazerle kesme yapan robotun kontrolünde[60], insanlarda omurga zedelenmelerinin elektrik ile uyarılmalarının kontrolünde[61], karmaşık ve gürültülü örneklerin sınıflandırılmasında[62], betonarme çerçevelerinin inşa edilebilirlik analizlerinde[63] kullanılmaktadır[66].

Endüktif öğrenme ile ilgili birçok program geliştirilmiştir[66]. En iyi bilinen programlar olarak böl ve fethet yaklaşımını kullanan ID3 programı[81], kapsama yaklaşımını kullanan AQ programı[82], endüktif mantık programlama yaklaşımını kullanan FOIL[83] ve GOLEM programları[83]. Bu programlar kesin karar kuralları üretmektedirler. Bunun yanında bulanık(fuzzy) kurallar üretmek için geliştirilmiş programlar da vardır[84].

## 2.5. Bilgi Ölçümü, Entropi ve Bilgi Kazancı

Entropi, en kaba tanımıyla bir sistemin düzensizliğinin ölçüsüdür. Entropi öylesine yararlı bir fiziksel kavramdır ki, pek çok disiplin içinde ayrı ayrı entropi fonksiyonları kullanılır. Örneğin termodinamik entropi, topolojik entropi vs... gibi. Sistemin düzensizliği arttıkça artan herhangi bir fonksiyon rahatlıkla entropi fonksiyonu olabilir.

Bilgi ölçümü, olasılık ile ters orantılıdır. Örneğin, bir  $x$  çıktısı için olasılık  $p(x)$  ise, aynı çıktı için bilgi ölçümü ( $I(x)$ ) denklem (2.4) gibi gösterilir[86].

$$I(x) = f\left(\frac{1}{p(x)}\right) \quad (2.4)$$

$x_1$  ve  $x_2$  gibi iki çıktının olasılığı, ikisinin ayrı ayrı olasılıklarının çarpımına eşittir.

$$p(x_1 \text{ and } x_2) = p(x_1) p(x_2) \quad (2.5)$$

(2.5) eşitliğinin her iki tarafının 1'e bölünmesi durumunda eşitlik değişmeyecektir.

$$\frac{1}{p(x_1 \text{ and } x_2)} = \frac{1}{p(x_1)} \frac{1}{p(x_2)} \quad (2.6)$$

Denklem (2.6)'ün her iki tarafının logaritması alındığında (2.7) nolu denklem elde edilir.

$$\log \frac{1}{p(x_1 \text{ and } x_2)} = \log \frac{1}{p(x_1)} + \log \frac{1}{p(x_2)} \quad (2.7)$$

Ayrıca  $x_1$  ve  $x_2$  gibi iki çıktının bilgi ölçümü, bu çıktıların ayrı ayrı bilgi ölçümleri toplamına eşittir. Bu işlem denklem (2.8) ile gösterilebilir.

$$I(x_1 \text{ AND } x_2) = I(x_1) + I(x_2) \quad (2.8)$$

$$I(x_1) = \log \left( \frac{1}{p(x_1)} \right) \quad (2.9)$$

(2.9) nolu denklem geliştirildiğinde, bir  $x$  çıktısı için bilgi ölçümünü veren denklem, (2.10)'deki denklem elde edilir.

$$I(x) = f \left( \frac{1}{p(x)} \right) = \log \left( \frac{1}{p(x)} \right) \quad (2.10)$$

Bir olayın çıktılarındaki ortalama bilgi ölçümü, bir olayın entropisi olup olayın mümkün olan tüm tüm çıktılarının bilgi ölçümlerinin ağırlıklı ortalaması olarak hesaplanır. Burada kullanılan ağırlıklar çıktıların olasılıklarıdır.

Sadece iki çıktısı olabilen bir  $X$  olayı için, bu olayın olasılıkları  $p$  ve  $(1-p)$ 'dir. Yukarıda verilen tanıma göre bu olayın entropisi  $E(X)$ , denklem (2.11) ile hesaplanabilir.

$$E(X) = p \log_2 \frac{1}{p} + (1-p) \log_2 \frac{1}{1-p} \quad (2.11)$$

Burada  $I(p) = \log(1/p)$  yazıldığı takdirde (2.12) nolu denklem elde edilir.

$$E(X) = p \cdot I(p) + (1-p) \cdot I(1-p) \quad (2.12)$$

Buna göre  $n$  adet çıktısı olan bir  $X$  olayının entropisi (2.13), (2.14) ve (2.15) yardımıyla hesaplanabilir.

$$E(X) = \sum_{i=1}^n p_i I_i \quad (2.13)$$

$$E(X) = \sum_{i=1}^n p_i \log \left( \frac{1}{p_i} \right) \quad (2.14)$$

$$E(X) = - \sum_{i=1}^n p_i \log(p_i) \quad (2.15)$$

Burada logaritma için istenilen herhangi bir taban kullanılabilir. İki tabanının kullanılması daha yaygındır. Ancak bu durumda bilgi ölçümü ve entropi için birim olarak *bit* (binary digit) kullanılabilir.

### 2.5.1. Entropi ve bilgi kazancının hesaplanması

Bilgi kazancı, eğitim verilerinden öncelikli olarak hangi karakteristiğin seçileceğine karar vermek için kullanılır. Bilgi kazancı, entropi olarak bilinen ve genellikle bilgi teorisinde kullanılan bir ölçüdür. Sadece iki sınıftan oluşan bir  $S$  eğitim verisi için entropi denklem (2.13) ile hesaplanır[87].

$$E(S) = -p_{(+)} \log_2 p_{(+)} - p_{(-)} \log_2 p_{(-)} \quad (2.13)$$

Buradaki  $p_{(+)}$   $S$  örnek setindeki pozitif örneklerin ve  $p_{(-)}$  ise negatif örneklerin oranını göstermektedir. Örnek setindeki sınıf sayısının ikiden fazla olması durumunda entropi, (2.14)'deki denklem yardımıyla hesaplanabilir.

$$E(S) = -\sum_{i=1}^n p_{(i)} \log_2 (p_{(i)}) \quad (2.14)$$

Entropi ölçüsü bir örnek ile kolaylıkla açıklanabilir.  $S$  örnek setinin on adet örnekten oluştuğu kabul edilsin. Yedi adet örneğin pozitif sınıfa ve üç adet örneğin ise negatif sınıfa ait olması durumunda entropi aşağıdaki gibi hesaplanır[7+,3-]:

$$E(S) = -\frac{7}{10} \log_2 \frac{7}{10} - \frac{3}{10} \log_2 \frac{3}{10} = 0.360 + 0.521 = 0.881 \text{ bit.}$$

Pozitif ve negatif örnek sayılarının birbirine eşit olması durumunda ( $p_{(+)} = p_{(-)} = 0.5$ ) entropi ölçüsü 1 ve bütün örneklerin aynı sınıfa ait olması durumunda ise entropi ölçüsü 0 (sıfır) olacaktır. Pozitif ve negatif örneklerin sayısı birbirine eşit değilse bu durumda entropi ölçüsü 0 ile 1 arasında olacaktır.

Bilgi kazancı, bir  $A$  karakteristiğine göre  $S$  örnek setinin örnekleri kısımlara ayrıştırılırken entropisinde beklenen azalmaz. Bilgi kazancı (2.15) yardımıyla hesaplanır.

$$Kazanç(S, A) = E(S) - \sum_{v \in Değerler(A)} \frac{|S_v|}{|S|} E(S_v) \quad (2.15)$$

Burada  $Değerler(A)$ ,  $A$  karakteristiği için mümkün tüm değerlerin bir kümesini ve  $S_v$  ise  $A$  karakteristiği için  $S$  örnek setindeki  $v$  değerine sahip örneklerin oluşturduğu alt seti gösterir. Denklemdaki ilk terim orijinal veri setinin entropisi ve ikinci terim ise  $A$  karakteristiğinin entropisidir. İkinci terim, alt sete ait olan örneklerin sayısı

tarafından ağırlıklandırılmış olan her bir  $S$ , alt seti entropilerinin toplamından çok değildir. Aşağıda, hayali bir örnek seti üzerinde  $Kazanç(S,A)$ 'nın nasıl hesaplandığı gösterilmiştir.  $S$  örnek seti on örnekten (7 pozitif ve 3 negatif) ve bir *Sıcaklık* karakteristiğinden oluşsun. *Sıcaklık* karakteristiği ise  $\{Sıcak, Soğuk\}$  olmak üzere iki farklı değer içersin. *Sıcak* değeri 4 örnekte geçmekte ve bunların 3 tanesi pozitif, 1 tanesi ise negatif sınıfa aittir. *Soğuk* değeri ise 6 örnekte geçmekte ve bunların 4 tanesi pozitif, 2 tanesi ise negatif sınıfa aittir.  $Kazanç(S, Sıcaklık)$  aşağıdaki gibi hesaplanır:

$$S = [7+, 3-]$$

$$Sıcak = [3+, 1-]$$

$$Soğuk = [4+, 2-]$$

$$\begin{aligned} Kazanç(S, Sıcaklık) &= 0.881 - \frac{4}{10} Entropi(Sıcak) - \frac{6}{10} Entropi(Soğuk) \\ &= 0.881 - \frac{4}{10} \times 0.811 - \frac{6}{10} \times 0.918 \\ &= 0.032 \text{ bit} \end{aligned}$$

### BÖLÜM 3. ENDÜKTİF ÖĞRENME ALGORİTMALARI

#### 3.1. Giriş

En genel tanımıyla Endüktif Öğrenme (Inductive Learning), bir konu ile ilgili özel örneklerden genel kurallar çıkarma tekniği olarak ifade edilebilir[88, 89, 90]. Endüktif Öğrenme, bilginin genelleştirilmesi, özelleştirilmesi, dönüşümü, düzeltilmesi ve iyileştirilmesi işlemlerini içermektedir[82].

Endüktif Öğrenme’de girdi bir örnekler setidir. Bu örnekler daha önce de açıklandığı gibi arşivlerden elde edilebileceği gibi, konu ile ilgili bir takım gözlem ve deneylerden de elde edilebilir. Örnekler setini oluşturan her bir örnek, konu ile ilgili tanımlanmış bir takım karakteristik değerlerinden (attribute-value) ve örneğin ait olduğu sınıftan oluşur. Örneği oluşturan sınıf dışındaki bilgiler nesne (object) olarak adlandırılır.

Tablo 3.1. Mevsimler örnek seti

| Hava     | Ağaçlar/Yapraklar | Sıcaklık | Mevsim   |
|----------|-------------------|----------|----------|
| Yağmurlu | Sarı              | Normal   | Sonbahar |
| Güneşli  | Sarı              | Normal   | Sonbahar |
| Yağmurlu | Yapraksız         | Normal   | Sonbahar |
| Yağmurlu | Yeşil             | Normal   | İlkbahar |
| Güneşli  | Yeşil             | Normal   | İlkbahar |
| Güneşli  | Yapraksız         | Düşük    | Kış      |
| Karlı    | Yapraksız         | Düşük    | Kış      |
| Yağmurlu | Yapraksız         | Düşük    | Kış      |
| Karlı    | Yeşil             | Düşük    | Kış      |
| Güneşli  | Yeşil             | Yüksek   | Yaz      |
| Yağmurlu | Yeşil             | Yüksek   | Yaz      |

Tablo 3.1. de verilen mevsimler ile ilgili örnekler setinde, *Hava*, *Ağaçlar/Yapraklar* ve *Sıcaklık* bilgilerinin her biri karakteristik olarak adlandırılır. Hava karakteristiği için *Yağmurlu*, *Güneşli* ve *Karlı* bilgileri ise değerler olarak adlandırılır. *İlkbahar*,

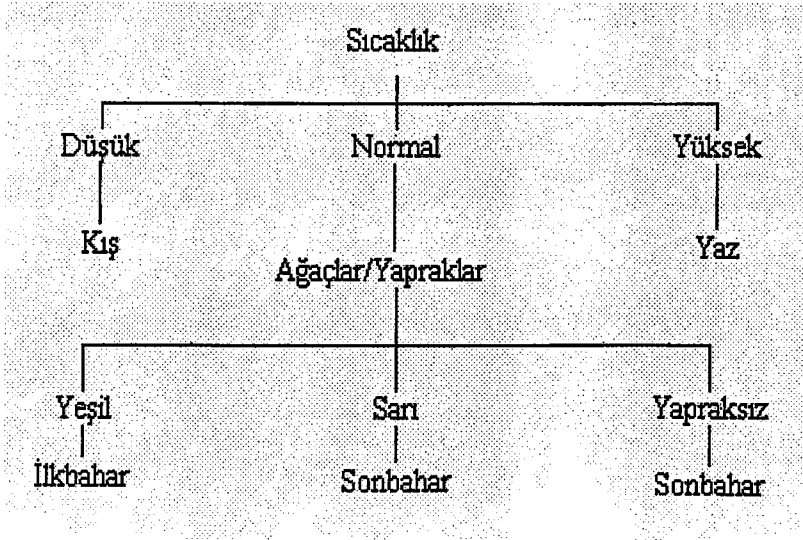
*Yaz, Sonbahar ve Kış* bilgileri ise sınıfları oluştururlar. Tablo 3.1.'de verilen karakteristikler ve bunlara ait değerler listesi aşağıdaki gibi gösterilebilir:

| Karakteristik     | Değerler listesi         |
|-------------------|--------------------------|
| Hava              | Yağmurlu, Güneşli, Karlı |
| Ağaçlar/Yapraklar | Sarı, Yeşil, Yapraksız   |
| Sıcaklık          | Normal, Düşük, Yüksek    |

Tablo 3.1'deki birinci örnek; **EĞER** *Hava=Yağmurlu* **VE** *Ağaçlar/Yapraklar=Sarı* **VE** *Sıcaklık=Normal* **İSE** *Mevsim Sonbahar*'dır anlamına gelmektedir. Bu örnekte görüldüğü gibi üç temel karakteristiğin değerleri ve bir de sonuç (*Mevsim*) yer almaktadır.

Endüktif Öğrenme tekniğinden beklenen eldeki örnek setindeki tüm örnekleri sınıflandırabilecek ve ayrıca benzer örnekler sunulduğunda bunların da hangi sınıfa dahil olabileceğini mümkün olduğunca doğru tahmin etmesini sağlayacak bir bilgi tabanı oluşturmastır.

Bir Endüktif Öğrenme algoritmasının çıktısı ya bir **karar ağacı** veya **kurallar seti** olabilir. Karar ağacından kurallar kolaylıkla çıkarılabilir[91, 92]. Çünkü karar ağacının her bir dalı bir kural olarak değerlendirilir[92].



Şekil 3.1. Karar ağacı örneği

Şekil 3.1. de verilen karar ağacından aşağıdaki kural seti çıkarılabilir:

Kural 1. EĞER Sıcaklık Düşük İSE Mevsim Kış

Kural 2. EĞER Sıcaklık Normal VE Ağaçlar/Yapraklar Yeşil İSE Mevsim İlkbahar

Kural 3. EĞER Sıcaklık Normal VE Ağaçlar/Yapraklar Sarı İSE Mevsim Sonbahar

Kural 4. EĞER Sıcaklık Normal VE Ağaçlar/Yapraklar Yapraksız İSE Mevsim Sonbahar

Kural 5. EĞER Sıcaklık Yüksek İSE Mevsim Yaz

### 3.2 Endüktif Öğrenme Tekniklerinin Sınıflandırılması

Endüktif Öğrenme teknikleri genel olarak iki gruba ayrılır [93]:

1. Karar ağacı üreten teknikler.
2. Kural seti üreten teknikler.

Karar ağacı üreten teknikler verilmiş olan örnekler setini alt setlere ayırır ve daha sonra bu alt setlere göre karar ağacını oluştururlar. Bu örneklerin çıktısı bir karar ağacıdır. Karar ağacı üreten algoritmalara, CLS [94] ve ID3 [13] örnek olarak verilebilir.

Doğrudan kural çıkaran teknikler ise karar ağacını oluşturmadan doğrudan kuralları oluştururlar. Doğrudan kural çıkaran algoritmalara AQ [9] ve RULES [59] ailesi algoritmalar örnek olarak verilebilir.

Her iki teknikteki algoritmalar uygulamalı yapay zekada önemli bir yer tutmaktadırlar.

### 3.3. CLS Ailesi Algoritmaları

CLS (Concept Learning System-Kavram Geliştirme Sistemi) algoritması, Endüktif öğrenmede geliştirilmiş ilk algoritma olarak kabul edilmektedir. Hunt ve arkadaşları tarafından geliştirilen bu algoritmanın dokuz farklı versiyonu vardır. Bu algoritmalar, CLS-1,..., CLS-9 [94] olarak isimlendirilmektedirler. İlk sekiz algoritma sadece iki

sınıflı problemlere uygulanabilmelerine rağmen, son versiyon çok sınıflı problemlere uygulanabilmektedir. CLS algoritmalarının temel mantığı Şekil 3.2’de adım adım açıklanmıştır[59].

- Adım 1.** Verilmiş örnekler setindeki sınıflardan biri Pozitif Sınıf diğeri ise Negatif Sınıf olarak kabul edilir ve örnekler seti bu sınıf değerlerine göre iki farklı gruba ayrılır.
- Adım 2.** Örnekler setinden rasgele bir örnek seçilir. Seçilen bu örnek Pozitif veya Negatif sınıfa ait olabilir.
- Adım 3.** Seçilen bu örnek içindeki bir veya birkaç değerden sadece Pozitif sınıfa ait olan ve aynı zamanda Negatif sınıfa ait olmayanlar bulunur. Ayrıca, eğer seçilen örnek ile bir sonraki örnek arasında aynı karakteristiklere ait ortak hiçbir değer yoksa Adım 5’e geçilir.
- Adım 4.** Eğer Adım 3 ile bulunan değer veya değerler var ise bu değere göre ağaç dallandırılır. (Bulunan bu değer ilk değer ise ağacın kökü olarak kabul edilir.)
- Adım 5.** Bu aşamaya kadar bir değer bulunmaz ise seçilmiş olan örnekteki bir veya birkaç değerden sadece Negatif sınıfa ait olan ve aynı zamanda Pozitif sınıfa ait olmayanlar bulunur. Ayrıca, eğer seçilen örnek ile bir sonraki örnek arasında aynı karakteristiklere ait ortak hiçbir değer yoksa Adım 7’ye geçilir.
- Adım 6.** Eğer adım 5 ile bulunan değer veya değerler var ise bu değere göre ağaç dallandırılır. (Bulunan bu değer ilk değer ise ağacın kökü olarak kabul edilir.)
- Adım 7.** Eğer bu aşamaya kadar hiçbir değer bulunmaz ise Pozitif sınıfa ait karakteristik değerlerinin bağıl frekansı hesaplanır. En yüksek frekansa sahip değer seçilir. Örnekler seti, seçilen bu değeri bulunduran veya bulundurmeyen olmak üzere iki alt sete ayrılarak dallandırılır. Her alt set de kendi içerisinde Pozitif veya Negatif sınıf değerlerine göre ayrılır.
- Adım 8.** Aynı işlemler oluşturulan alt setlere de uygulayarak karar ağacı oluşturulur. Bunun için, eğer işlem tamamlanmamışsa yeni örnek seti için 2 nolu adıma gidilir.

Şekil 3.2. CLS Algoritması

### 3.3.1. Örnek Uygulama

Tablo 3.2’de verilen örnekler setini kullanarak karar ağacının oluşturulması istensin. Burada dört karakteristik (k1, k2, k3, k4) ve iki sınıf (Pozitif, Negatif) olduğu görülmektedir.

Tablo 3.2. CLS algoritması için örnek seti [94]

| No | K1 | K2 | K3 | K4 | Sınıf   |
|----|----|----|----|----|---------|
| 1  | J  | R  | T  | K  | Pozitif |
| 2  | Q  | P  | Q  | Z  | Pozitif |
| 3  | B  | R  | Y  | L  | Pozitif |
| 4  | X  | V  | Q  | M  | Pozitif |
| 5  | P  | T  | Q  | W  | Pozitif |
| 6  | F  | R  | K  | V  | Pozitif |
| 7  | N  | S  | Q  | K  | Pozitif |
| 8  | J  | V  | B  | K  | Negatif |
| 9  | Q  | P  | W  | Y  | Negatif |
| 10 | N  | T  | Y  | D  | Negatif |
| 11 | M  | R  | Q  | M  | Negatif |
| 12 | J  | C  | T  | K  | Negatif |
| 13 | B  | R  | Q  | V  | Negatif |
| 14 | L  | R  | Q  | W  | Negatif |

Tablo 3.2’de verilen örnek seti için CLS algoritmasının uygulaması aşağıda adım adım anlatılmıştır.

#### Birinci aşama:

**Adım 1.** Örnek seti Pozitif ve Negatif sınıf değerlerine göre iki farklı gruba ayrılır. Bu şekilde hazırlanan örnekler seti Tablo 3.3’te verilmiştir.

**Adım 2.** Pozitif sınıftaki örnekler setinden ilk örnek seçilir. Bu örnek {J, R, T, K} değerlerinden oluşmaktadır.

**Adım 3.** Bu örnekteki tüm değerler hem Pozitif hem de Negatif sınıflarında geçmektedir. Bu değerlerden hiçbirini seçilemez. Ayrıca seçilen örnekteki değerlerden {J,R,T,K} hiçbirini bir sonraki örnekte {Q, P, Q Z} geçmediğinden dolayı bu değerlerden biri seçilemez. Bu durumda 5. adıma geçilir.

Tablo 3.3. Tablo 3.2'deki örnek setinin Pozitif ve Negatif olarak ayrı ayrı sınıflandırılması

| Pozitif |    |    |    |    | Negatif |    |    |    |    |
|---------|----|----|----|----|---------|----|----|----|----|
| No      | K1 | K2 | K3 | K4 | No      | K1 | K2 | K3 | K4 |
| 1       | J  | R  | T  | K  | 8       | J  | V  | B  | K  |
| 2       | Q  | P  | Q  | Z  | 9       | Q  | P  | W  | Y  |
| 3       | B  | R  | Y  | L  | 10      | N  | T  | Y  | D  |
| 4       | X  | V  | Q  | M  | 11      | M  | R  | Q  | M  |
| 5       | P  | T  | Q  | W  | 12      | J  | C  | T  | K  |
| 6       | F  | R  | K  | V  | 13      | B  | R  | Q  | V  |
| 7       | N  | S  | Q  | K  | 14      | L  | R  | Q  | W  |

**Adım 5.** Negatif sınıfında yer alan ilk örnek seçilir. Bu örnek {J, V, B, K} değerlerinden oluşmaktadır. Bu değerlerin hepsi aynı zamanda Pozitif sınıfında da geçtiğinden dolayı seçilemezler. Ayrıca bu örnekteki değerlerden hiçbirisi bir sonraki örnekte de geçmediklerinden dolayı bu değerlerden hiçbirisi seçilemez. Bu durumda 7. adıma geçilir.

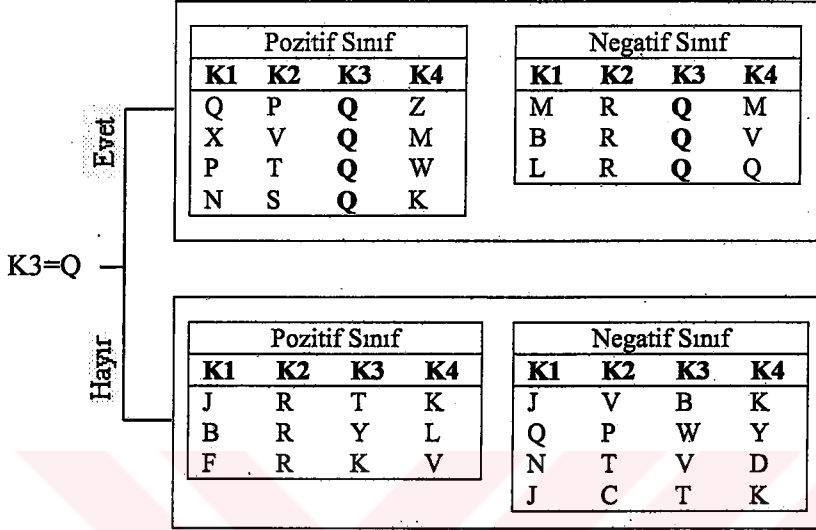
**Adım 7.** Bu adıma kadar hiçbir değer bulunmadığından dolayı, Pozitif sınıftaki her karakteristik için en çok tekrar eden değerlerin bağıl frekansları hesaplanır. K1 karakteristiğine ait hiçbir değer birden fazla sayıda tekrar etmemektedir. K2 karakteristiğine ait R değeri üç defa tekrarlanmaktadır. K2 karakteristiğinin bağıl frekansı 3 olarak alınır. K3 karakteristiğine ait Q değeri dört defa tekrar ettiğinden dolayı bağıl frekansı 4 olarak alınır. K4 karakteristiğine ait K değeri iki defa tekrar ettiğinden dolayı bağıl frekansı 2 olarak alınır. Bağıl frekansı en büyük olan değer K3 karakteristiğine ait Q değeridir. Bu değer karar ağacının kökü olarak kabul edilir.

K3 karakteristiğinin Q değerine göre karar ağacı yeniden dallandırılır. Karar ağacının bir dalında K3 karakteristik değeri Q olan örnekler Pozitif ve Negatif olmak üzere iki gruba ayrılır. Karar ağacının diğer dalında ise K3 karakteristiğinin değeri Q olmayan örnekler, Pozitif ve Negatif olmak üzere iki gruba ayrılır. Bu bilgilere göre oluşturulan kısmi karar ağacı Şekil 3.3'te gösterilmiştir.

### İkinci aşama:

Bu aşamada K3 karakteristiği daha önce seçildiği için dikkate alınmaz.

**Adım 2.** Üst tarafta yer alan Pozitif sınıfındaki ilk örnek {Q, P, Q, Z} seçilir.



Şekil 3.3 Tablo 3.1’de verilen örnekler setine ait birinci kısmi karar ağacı

**Adım 3.** İlk örnekte yer alan {Q,P,Q, Z} değerlerinden hiçbirini ikinci örnekte {X, V, Q, M} geçmediğinden dolayı hiçbir değer seçilemez (K3=Q dikkate alınmaz) ve 5 nolu adıma gidilir.

**Adım 5.** Şekil 3.3’teki üst tarafta yer alan Negatif sınıfında bulunan K2=R ortak değerlere sahiptir. Ayrıca sol taraftaki Pozitif sınıfında hiçbir örnek için K2=R değildir. Bundan dolayı üst taraftaki dal için **K2=R** bir çözümdür. Bu durumda diğer adımlar atlanır. Bu bilgilere göre oluşturulan kısmi karar ağacı Şekil 3.4’te gösterilmiştir.

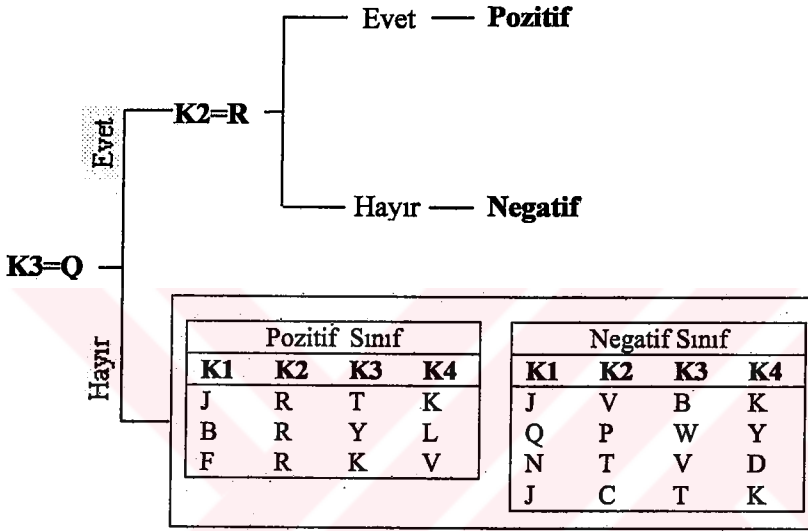
### Üçüncü aşama:

Bu aşamada da K3 karakteristiği daha önce seçildiği için dikkate alınmaz.

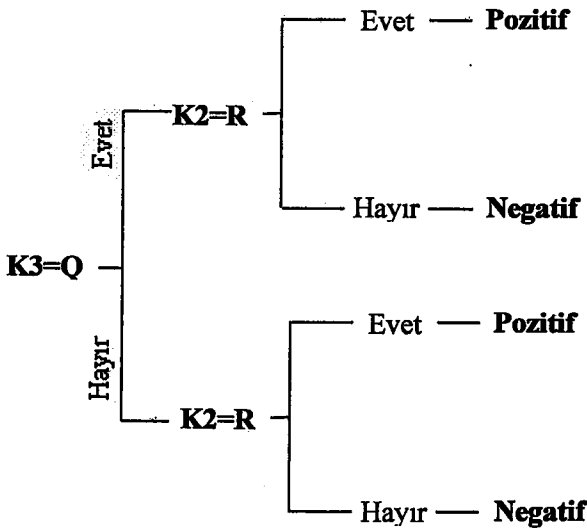
**Adım 2.** Alt tarafta yer alan Pozitif sınıfındaki ilk örnek {J, R, T, K} seçilir.

**Adım 3.** Sağ taraftaki Negatif sınıfındaki hiçbir değer için  $K2=R$  değildir. Ayrıca ilk örnekte yer alan  $\{J, R, T, K\}$  değerlerinden  $K2=R$  değeri diğer iki örnekte de geçmektedir. Bu durumda, kalan alt set için  $K2=R$  bir çözümdür. Algoritma bu şekilde tamamlanmış olur.

Bu bilgilere göre oluşturulan son karar ağacı Şekil 3.5'te gösterilmiştir.



Şekil 3.4. Tablo 3.3'te verilen örnekler seti için ikinci kısmi karar ağacı



Şekil 3.5. Tablo 3.2'de verilen örnekler seti için son karar ağacı

Şekil 3.5'te verilen karar ağacından, Tablo 3.4'teki kural seti elde edilir.

Tablo 3.4. Şekil 3.5'teki karar ağacından elde edilen kural seti

| Kural No | Kural Tanımı                                      |
|----------|---------------------------------------------------|
| 1        | Eğer $K3 = Q$ VE $K2 = R$ İSE Sınıf Pozitif       |
| 2        | Eğer $K3 = Q$ VE $K2 \neq R$ İSE Sınıf Negatif    |
| 3        | Eğer $K3 \neq Q$ VE $K2 = R$ İSE Sınıf Pozitif    |
| 4        | Eğer $K3 \neq Q$ VE $K2 \neq R$ İSE Sınıf Negatif |

### 3.3.2. CLS ailesinin diğer algoritmaları

CLS algoritmasının dokuz farklı versiyonu bulunmaktadır. İlk üç algoritmanın çalışma şekli birbirine benzemektedir. Aralarındaki farklar ise şunlardır: CLS-2 ile CLS-3'te bilgisayardaki maksimum RAM (Random Access Memory - Rasgele Erişimli Hafıza) bellek kapasitesini gösteren bir kontrol parametresi kullanılmıştır. Bu algoritmaların geliştirildiği yıllarda bilgisayardaki bellek kapasitesi çok düşüktü. Ayrıca karar ağacı oluşturulurken, örneklerin ve ara işlemlerin sonuçlarının RAM hafızada tutulma zorunluluğu vardı. Algoritmalar buna göre geliştirilmiştir. Örnek sayısının fazla olması durumunda bellek kapasitelerinin yetersiz olmasından dolayı problemler yaşanmaktaydı[59].

CLS-2 ile CLS-3 algoritmaları bilgileri bellekte saklama şekli ile birbirinden ayrılırlar. CLS-4 ve CLS-5 algoritmaları CLS-1 algoritmasından farkı ise işlem görececek örneklerin farklı bir yöntemle seçilmesidir. Her iki algoritmada da seçilecek örnekler sınıflandırma sırasında ortaya çıkan hataya göre seçilmektedir. CLS-6 algoritmasında alt setler bağıl frekanslarına göre seçilmektedir. CLS-7'de de Pozitif setin tanımı farklı bir şekilde yapılmaktadır. Her aşamada iki alt setten, örnek sayısı büyük olan Pozitif set olarak kabul edilir. CLS-8 algoritması da CLS-7 algoritmasından farklı olarak, iki alt setten, örnek sayısı küçük olan Pozitif set olarak kabul eder. CLS-9 algoritması ise kendisinden önceki tüm algoritmaların avantajlarını taşımakla beraber farklı olarak çok dallı karar ağaçlarının oluşturulmasını sağlayabilir. Sadece iki sınıflı problemlerde değil, çok sınıflı problemlerde de çözüm bulabilir [59].

### 3.4. ID3 Ailesi Algoritmaları

ID3 algoritması, CLS algoritması gibi karar ağacı üreten bir algoritmadır. Karar ağacı şeklinde kural üreten algoritmaların en iyilerinden biri olarak kabul edilir. İki yada daha fazla sınıfa ait örneklerin sınıflandırabilmesi için kurallar üreten bir algoritmadır[13]. Sadece ikili ağaç değil, çok dallı ağaç da oluşturabilir. CLS-9 algoritmasından farklı olarak, karar ağacını oluştururken sadece bağıl frekansı değil, örnek setinde geçen karakteristiklerin bilgi kazancını hesaplayarak verilen örnekler setini alt setlere ayırır ve karar ağacını bu şekilde oluşturur (Bilgi kazancının hesaplanması sırasında bağıl frekans dikkate alınır). Bu algoritma verilen bir örneği sınıflandırmak için gerekli olan kural veya kuralları minimum sayıda test yaparak oluşturmayı amaçlamaktadır. ID3 algoritmasının adım adım gösterilişi Şekil 3.6'da verilmiştir.

|                |                                                                                                                                                                          |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Adım1.</b>  | Verilen örnekler setinden bir alt örnek seti seçilir.                                                                                                                    |
| <b>Adım 2.</b> | Seçilen alt sette kullanılan tüm karakteristiklerin bilgi kazancı hesaplanır.                                                                                            |
| <b>Adım 3.</b> | Bilgi kazancı en fazla olan karakteristik ağacın kökü olarak kabul edilir. Bu ağaçta kullanılan değerlere göre ağaç dallandırılır.                                       |
| <b>Adım 4.</b> | Tüm örnek seti bu değerlere göre alt setlere ayrılır.                                                                                                                    |
| <b>Adım 5.</b> | Her alt sette farklı sınıflara ait örnekler kalmadıysa işlem tamamlanır aksi halde alt setlerde bulunan diğer karakteristiklerin her biri için bilgi kazancı hesaplanır. |
| <b>Adım 6.</b> | Bilgi kazancı en fazla olan karakteristik düğüm olarak alınır ve set alt-setlere ayrılır.                                                                                |
| <b>Adım 7.</b> | Eğer farklı sınıflara ait örnekler kalmadıysa işlem tamamlanır aksi halde alt setlerde farklı sınıflara ait örnekler kalmayınca kadar bu işlemler tekrarlanır.           |

Şekil 3.6. ID3 Algoritması

ID3 algoritması, bilgi kazancı en fazla olan karakteristiği seçerek bu karakteristiği ilk aşamada ağacın kökü, diğer aşamalarda ise bir düğüm (alt set için kök) olarak alır ve

buna göre ağacını dallandırır. Bilgi kazanımı birden fazla karakteristik için aynı ise herhangi biri seçilebilir.

### 3.4.1. Örnek Uygulama

ID3 algoritmasının çalışma prensibini daha iyi anlamak için Tablo 3.5'te bir örnek seti verilmiştir. ID3 algoritması adım adım uygulanarak, verilen örnek setinden karar ağacı oluşturulacaktır.

Tablo 3.5. Mevsimler eğitim seti

| Hava     | Ağaçlar   | Sıcaklık | Mevsim   |
|----------|-----------|----------|----------|
| Yağmurlu | Yapraksız | Normal   | Sonbahar |
| Yağmurlu | Yapraksız | Düşük    | Kış      |
| Karlı    | Yapraksız | Düşük    | Kış      |
| Güneşli  | Yapraksız | Düşük    | Kış      |
| Yağmurlu | Sarı      | Normal   | Sonbahar |
| Yağmurlu | Yeşil     | Yüksek   | Yaz      |
| Yağmurlu | Yeşil     | Normal   | İlkbahar |
| Güneşli  | Yeşil     | Normal   | İlkbahar |
| Güneşli  | Yeşil     | Yüksek   | Yaz      |
| Güneşli  | Sarı      | Normal   | Sonbahar |
| Karlı    | Yeşil     | Düşük    | Kış      |

Tablo 3.5'te verilen örnekler setinde üç temel karakteristik bulunmaktadır. Bu karakteristikler ve değerleri;

| Karakteristik | Değerler                 |
|---------------|--------------------------|
| Hava          | Yağmurlu, Karlı, Güneşli |
| Ağaçlar       | Yapraksız, Sarı, Yeşil   |
| Sıcaklık      | Normal, Düşük, Yüksek    |

şeklindedir.

Örnek setinde sınıf olarak *Mevsim* verilmiş ve değerleri ise *İlkbahar*, *Yaz*, *Sonbahar*, *Kış*'tır. Karar ağacını hazırlayabilmek için tüm karakteristiklerin bilgi kazancını hesaplamak gerekir. Bilgi kazancının nasıl hesaplandığı Bölüm-2'de detaylı olarak açıklanmıştır.

Karar ağacını oluşturmak için tüm örneğin bilgi ölçüsünü hesaplamak gerekir. Tablo 3.5'teki örnek setinde *İlkbahar* sınıfına (mevsimine) ait örneklerin sayısı 2, *Yaz* sınıfına ait örneklerin sayısı 2, *Sonbahar* sınıfına ait örneklerin sayısı 3 ve *Kış* mevsimine ait örneklerin sayısı ise 4'tür. Bu örnekler setindeki örnek sayısı 11 olduğundan, örnekler setinin bilgi ölçüsü hesaplandığında, sonuç olarak  $I(S)=1.94$  bit değeri bulunur. Her karakteristik için entropi değerlerinin hesaplanması durumunda Tablo 3.6'daki sonuçlar elde edilir.

Tablo 3.6. Karakteristikler ve değerlere ait entropi değerleri

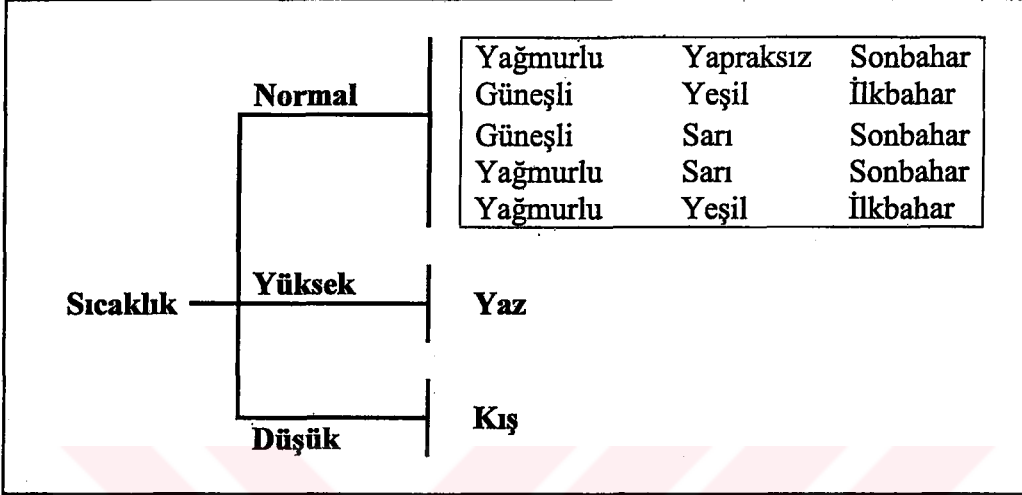
| Karakteristik | Entropi (bit) | Kazanç (bit) |
|---------------|---------------|--------------|
| Hava          | 1.601         | 0.339        |
| Ağaçlar       | 0.987         | 0.953        |
| Sıcaklık      | 0.441         | 1.499        |

Tablo 3.6'daki bilgilere göre bilgi kazancı en yüksek olan *Sıcaklık* karakteristiği karar ağacının kökü olarak alınır. Daha sonra *Ağaçlar* ve ardından da *Hava* gelecek şekilde karar ağacı oluşturulur. Bu bilgilere göre hazırlanan karar ağacı Şekil 3.7'deki gibi olacaktır.

|          |        |          |           |          |
|----------|--------|----------|-----------|----------|
| Sıcaklık | Normal | Yağmurlu | Yapraksız | Sonbahar |
|          |        | Güneşli  | Yeşil     | İlkbahar |
|          |        | Güneşli  | Sarı      | Sonbahar |
|          |        | Yağmurlu | Sarı      | Sonbahar |
|          |        | Yağmurlu | Yeşil     | İlkbahar |
|          | Yüksek | Yağmurlu | Yeşil     | Yaz      |
|          |        | Güneşli  | Yeşil     | Yaz      |
|          | Düşük  | Güneşli  | Yapraksız | Kış      |
|          |        | Yağmurlu | Yapraksız | Kış      |
|          |        | Karlı    | Yapraksız | Kış      |
|          |        | Karlı    | Yeşil     | Kış      |

Şekil 3.7. *Sıcaklık* karakteristiğine göre hazırlanan karar ağacı

Şekil 3.7'deki *Sıcaklık* karakteristiğine ait karar ağacında da görüldüğü gibi  $\{Sıcaklık, Düşük\}$  karakteristik-değeri 4 örnekte ve sadece *Kış* mevsimine karşılık gelmektedir. Aynı şekilde  $\{Sıcaklık, Yüksek\}$  karakteristik-değeri sadece *Yaz* mevsiminde geçmektedir. Buna göre kısmi karar ağacı Şekil 3.8'deki gibi olur.



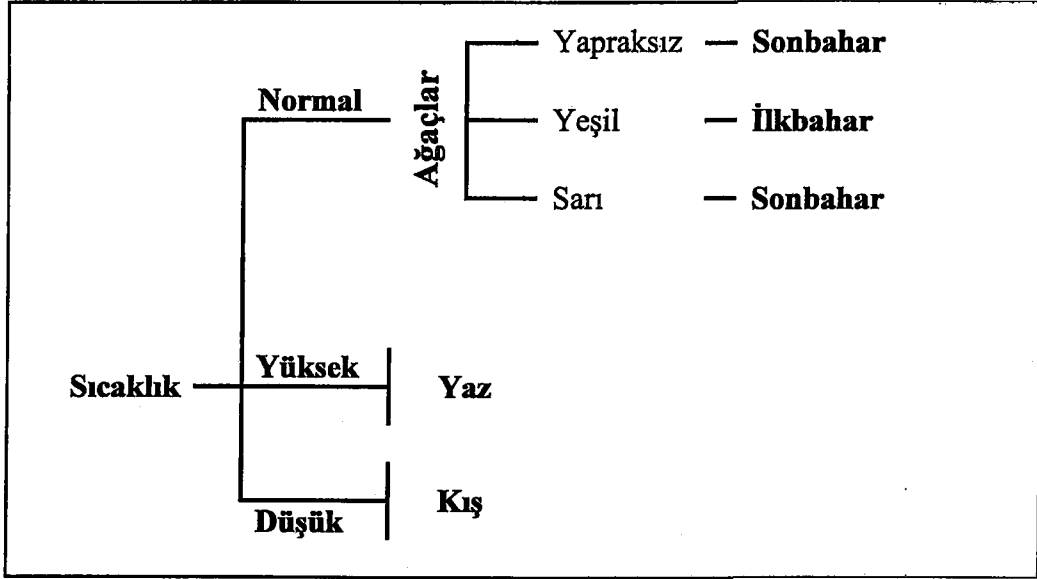
Şekil 3.8. *Sıcaklık* karakteristiğine göre hazırlanan kısmi karar ağacı

Şekil 3.8'deki kısmi karar ağacının  $\{Sıcaklık, Normal\}$  dalında yeni bir alt set (*SI*) meydana gelmiştir. Sadece bu alt set için bilgi kazançlarının hesaplanması durumunda Tablo 3.7'deki değerler elde edilir. *SI* alt setinin bilgi değeri  $I(SI)=0.97$  olarak bulunur.

Tablo 3.7. *SI* alt seti için entropi ve bilgi kazancı değerleri

| Karakteristik | Entropi (bit) | Bilgi Kazancı (bit) |
|---------------|---------------|---------------------|
| Hava          | 0.95          | 0.02                |
| Ağaçlar       | 0.97          | 0                   |

Kazancı en yüksek olan *Ağaçlar* karakteristiği, *SI* alt setinin kökü (örnek seti için düğüm noktası) olarak seçilir. Buna göre hazırlanan son karar ağacı Şekil 3.9'daki gibi olur. Şekil 3.9'daki karar ağacından elde edilen kurallar ise Tablo 3.8'de verilmiştir.



Şekil 3.9. Mevsimler örnek seti için son karar ağacı

Tablo 3.8. Mevsimler problemi için elde edilen kurallar (Şekil 3.9'dan)

| Kural No | Kural Tanımı                                                      |
|----------|-------------------------------------------------------------------|
| 1        | EĞER Sıcaklık Düşük İSE Mevsim Kış'tır                            |
| 2        | EĞER Sıcaklık Yüksek İSE Mevsim Yaz'dır                           |
| 3        | EĞER Sıcaklık Normal VE Ağaçlar Yapraksız İSE Mevsim Sonbahar'dır |
| 4        | EĞER Sıcaklık Normal VE Ağaçlar Yeşil İSE Mevsim İlkbahar'dır     |
| 5        | EĞER Sıcaklık Normal VE Ağaçlar Sarı İSE Mevsim Sonbahar'dır      |

### 3.4.2. ID3 algoritmasından türetilen algoritmalar

ID3 algoritması, değişmeyen bir örnek seti için iyi kurallar üretebilecek karar ağacını oluşturur. Sık sık değişen örnek seti için karar ağacının her defasında yeniden oluşturulması gerekir. Bu eksikliği gidermek için Quinlan ve diğer araştırmacılar, ID3 algoritmasının özüne bağlı kalmak şartıyla birkaç farklı versiyonunu geliştirmişlerdir.

Schlimmer ve Fisher ID4 algoritmasını geliştirmişlerdir[50]. ID4 algoritması, mevcut örnekler setine yeni eklemeler yapıldığında karar ağacını yeniden oluşturmak yerine eski karar ağacını güncelleştirebilme özelliğine sahiptir. Bu algoritma her örneği tek

tek ele alarak oluşturulan kısmi karar ağacını her defasında güncelleştirerek son karar ağacını oluşturur. Daha sonra Utgoff bu işlemi farklı bir şekilde yapabilen ID5 algoritmasını geliştirmiştir[15]. ID4 algoritması bir test karakteristiğinin yerini değiştirirken bundan sonraki tüm alt setleri yok edip yeniden oluşturur. ID5 ise bu durumda en iyi karakteristiği yukarıya doğru çekerek ağacı güncelleştirir.

Bu algoritmalar ile üretilen karar ağacı, gereksiz ve alakasız durumları içerdiği için üretilecek kurallar genel olmayabilir. Çünkü, algoritma seçtiği her karakteristik için bir dal oluşturur. Ancak bu karakteristikteki değerler sınıflandırma için gereksiz kurallar içerebilir. Bu durumda detaylanmış sınıflandırma kurallarının oluşmasına yol açar. Bu özelliği düzeltmek için GID3(General version of ID3) algoritması geliştirilmiştir[14]. Genel olarak GID3 algoritması, ID3 algoritmasına benzemektedir. GID3 algoritması karakteristiğin her değeri için bir dal oluşturmak yerine, gereksiz değerler için dallanmayı engellemek için sadece gerekli değerlerle dallanmayı yapar.

ID3 algoritması örnekler setinde kullanılan tüm karakteristik değerlerinin belirli olması halinde çalışır. Bazı karakteristik değerlerinin bilinmemesi durumunda çalışmaz. Quinlan daha sonra geliştirdiği ve ID3'nin bir devamı olan C4.5 algoritması yardımıyla bu problemi çözmüştür[17]. ID3 ve C4.5 algoritmaları, günümüzde uygulamalarda en çok kullanılan algoritmalar olarak kabul edilmektedir.

ID4 algoritması karar ağaçlarını artan bir şekilde oluşturur. Yeni bir değişiklik meydana geldiği anda karar ağacı otomatik olarak güncellenir. Bütün örnekleri yükleyip, metodu bütün veriler üzerinde çalıştırarak aramayan bir algoritma da kullanılabilir. Bu, bütün verilerin saklanması gerektirir. ID5R[51] ağaçtaki her noktada (node) bulunan olayların özellikleri hakkında istatistik bilgilerini saklar. Yeni bir örnek girildiği zaman, örneğin dağılım üzerindeki etkisi hesaplanır ve metod, mevcut noktanın farklı bir özellik ile yer değiştirmesi gerektirip gerektirmediğini kontrol eder. Bu işlem yukarıdan aşağıya doğru özyinelemeli (rekürsiv) olarak her alt-dal için yapılabileceği gibi, aşağıdan yukarıya doğru da yapılabilir.

### 3.5. RULES Ailesi Algoritmaları

RULES ailesi algoritmaları Rules-1, Rules-2, Rules-3, Rules-3 Plus ve Rules-4 olmak üzere beş adet algorithmadan oluşmaktadır.

#### 3.5.1. RULES-1 algoritması

RULES-1 (RULES Extraction System) algoritması, verilen örnekler setinden doğrudan kural çıkarabilen bir algorithmadır[95]. RULES-1 algoritması RULES ailesinin ilk algoritmasıdır. RULES-1 algoritması iki farklı örnek seti ile çalışır. Bunlardan biri verilen örneklerin bulunduğu *örnekler seti* diğeri ise kural çıkarma sırasında kullanılan *kural çıkarma setidir*. Her aşamada kendisi için kural çıkarılmış olan örnek veya örnekler sonraki aşamalarda dikkate alınmazlar. Çıkarılan kurallar ile sınıflandırılmayan örnekler kural çıkarma setini oluştururlar. Kendisi için en az bir kural çıkarılan örnekler *sınıflandırılmış örnek* ve kendisi için hiçbir kural çıkarılmamış örnekler ise *sınıflandırılmamış örnek* olarak kabul edilirler.

Örnekler setinde yer alan karakteristikler için kullanılan değerlerin tümü *değerler dizisini* oluşturur. Değerler dizisinin eleman sayısı, kullanılan tüm karakteristiklerdeki değerlerin toplamına eşittir.

RULES-1 algoritması kural çıkarma işlemini aşağıdaki gibi gerçekleştirir:

Örnekler setinden bir değerler dizisi oluşturulur. Bu dizinin her elemanı tek tek ele alınarak kural oluşturup oluşturmadığı test edilir. Bunun için, seçilen değer için sınıflandırılmamış tüm örneklere bakılır. Eğer seçilen değer sınıflandırılmamış örnekler setinde sadece bir sınıfta geçiyorsa bu değer kural haline getirilir. Çıkarılan kurallar varsa bu kurallarla sınıflandırılabilen örnekler ayıklanır. Tüm örneklerin sınıflandırılması halinde işlem tamamlanır.

Eğer sınıflandırılmayan örnekler varsa bu örneklerdeki değerlerden yeni bir değerler dizisi oluşturulur. Bu değerler dizisinin tüm ikili kombinasyonları bulunur (Her kombinasyonda aynı karakteristiğe ait sadece bir değer bulunabilir).

Seçilen her ikili değer kombinasyonu için sınıflandırılmamış tüm örnekler bakılır. Eğer seçilen ikili değer kombinasyonu sadece bir sınıfta geçiyorsa bu kombinasyon kural haline getirilir. Çıkarılan kurallar varsa bu kurallarla sınıflandırılabilen örnekler ayklanır. Tüm örneklerin sınıflandırılmaması halinde işlem tamamlanır.

Bu işlemler sınıflandırılmamış örnek veya örneklerin kalması durumunda en fazla karakteristik sayısı kadar kombinasyon oluşturarak, tüm örnekler sınıflandırılıncaya kadar tekrar edilir.

RULES-1 algoritması Şekil 3.10'da gösterilmiştir[95].

|                |                                                                                                                                                       |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Adım 1.</b> | $N_c=0$                                                                                                                                               |
| <b>Adım 2.</b> | $N_c=N_c+1$                                                                                                                                           |
| <b>Adım 3.</b> | Eğer $N_c > N_a$ ise BİTİR                                                                                                                            |
| <b>Adım 4.</b> | Sınıflandırılmamış örneklerde kullanılan karakteristik değerleriyle bir değerler dizisi oluştur.                                                      |
| <b>Adım 5.</b> | Değerler dizisindeki elemanların $N_c$ 'li kombinasyonlarını bul.                                                                                     |
| <b>Adım 6.</b> | Eğer bu kombinasyondan en az bir tanesi kontrol setindeki tüm örneklerde sadece bir defa geçiyorsa bunu kural haline getir. Aksi halde Adım 2'ye git. |
| <b>Adım 7.</b> | Gereksiz şart kontrolü yap.                                                                                                                           |
| <b>Adım 8.</b> | Çıkarılmış kurallar ile sınıflandırılabilen örnekleri ayıkla.                                                                                         |
| <b>Adım 9.</b> | Şayet tüm örnekler sınıflandırılmış ise BİTİR, Aksi halde Adım 2'ye git.                                                                              |

Şekil 3.10. RULES-1 algoritması

RULES-1 algoritması, verilen örnekler setinden kurallar setini oluşturmak için en fazla;

$$\sum_{i=1}^{N_a} \frac{N_v!}{(N_v - i)!i!}$$

adet karşılaştırma işlemi yapar.

Burada kullanılan;  $N_v$  değeri sayısını,  $N_a$  ise karakteristik sayısını göstermektedir. Ancak sınıflandırılan örneklerin bir daha işleme dahil edilmediği düşünüldüğünde, karşılaştırma sayısının maksimum değerine hiçbir genelleştirilmiş kural çıkarılamaması durumunda ulaşılır.

### 3.5.1.1. Örnek uygulama

Tablo 3.9’da verilen örnekler seti kullanılarak RULES-1 algoritmasını ile gerekli kurallar çıkarılacaktır. Bu örnek ve çözümü Aksoy tarafından RULES-1 algoritması için kullanılmıştır[95].

Tablo 3.9. Mevsimler için örnek seti

| Örnek No | Hava     | Ağaçlar   | Sıcaklık | Mevsim   |
|----------|----------|-----------|----------|----------|
| 1        | Yağmurlu | Yapraksız | Normal   | Sonbahar |
| 2        | Yağmurlu | Yeşil     | Normal   | İlkbahar |
| 3        | Güneşli  | Yeşil     | Normal   | İlkbahar |
| 4        | Yağmurlu | Yapraksız | Düşük    | Kış      |
| 5        | Karlı    | Yapraksız | Düşük    | Kış      |
| 6        | Güneşli  | Yapraksız | Düşük    | Kış      |
| 7        | Yağmurlu | Yeşil     | Yüksek   | Yaz      |
| 8        | Yağmurlu | Sarı      | Normal   | Sonbahar |
| 9        | Güneşli  | Yeşil     | Yüksek   | Yaz      |
| 10       | Güneşli  | Sarı      | Normal   | Sonbahar |
| 11       | Karlı    | Yeşil     | Düşük    | Kış      |

Birinci aşamada Tablo 3.9’daki örnek setinde kullanılan tüm karakteristik ve değerler bulunur. Karakteristiklere ait değerler aşağıda verilmiştir:

| Karakteristik | Değerler                 |
|---------------|--------------------------|
| Hava          | Yağmurlu, Güneşli, Karlı |
| Ağaçlar       | Yeşil, Sarı, Yapraksız   |
| Sıcaklık      | Yüksek, Normal, Düşük    |

Bu değerlerin hangi sınıfta yer aldıkları Tablo 3.10’da gösterilmiştir. Bu değerlerin tek başlarına kural oluşturabilmesi için sadece bir sınıfta geçmeleri gerekmektedir. Tablo 3.10’a göre sadece bir mevsimde geçen *Karlı* (*Kış*), *Sarı* (*Sonbahar*), *Düşük* (*Kış*) ve *Yüksek* (*Yaz*) değerleri tek başlarına birer kural oluşturabilirler. Diğer

değerler birden fazla sınıfta geçtiklerinden dolayı tek başlarına kural oluşturamazlar. Tekli değerler ile tek şarttan oluşan kurallar çıkarılır (Tablo 3.11).

Tablo 3.10. Birinci aşama için karakteristik değerlerinin yer aldığı sınıflar

| Değer     | Sınıflar (Mevsimler)         |
|-----------|------------------------------|
| Yağmurlu  | İlkbahar, Yaz, Sonbahar, Kış |
| Güneşli   | İlkbahar, Yaz, Sonbahar, Kış |
| Karlı     | Kış                          |
| Yeşil     | İlkbahar, Yaz, Kış           |
| Sarı      | Sonbahar                     |
| Yapraksız | Sonbahar, Kış                |
| Düşük     | Kış                          |
| Normal    | Sonbahar, İlkbahar           |
| Yüksek    | Yaz                          |

Tablo 3.11. Birinci aşama sonunda elde edilen kurallar

| Kural No | Kural Tanımı                          |
|----------|---------------------------------------|
| 1        | EĞER Hava Karlı İSE Mevsim KIŞ        |
| 2        | EĞER Ağaçlar Sarı İSE Mevsim Sonbahar |
| 3        | Eğer Sıcaklık Düşük İSE Mevsim KIŞ    |
| 4        | EĞER Sıcaklık Yüksek İSE Mevsim YAZ   |

Bu kurallar Tablo 3.9'daki örnek setine uygulandığında, Kural-1 ile 5 ve 11 nolu örnekler, Kural-2 ile 8 ve 10 nolu örnekler, Kural-3 ile 4, 5, 6, ve 11 nolu örnekler ve Kural-4 ile 7 ve 9 nolu örnekler sınıflandırılabilir.

Birinci aşama sonunda sınıflandırılmayan örnekler Tablo 3.12'de verilmiştir.

Tablo 3.12. Birinci aşama sonunda sınıflandırılmayan örnek listesi

| Örnek No | Hava     | Ağaçlar   | Sıcaklık | Mevsim   |
|----------|----------|-----------|----------|----------|
| 1        | Yağmurlu | Yapraksız | Normal   | Sonbahar |
| 2        | Yağmurlu | Yeşil     | Normal   | İlkbahar |
| 3        | Güneşli  | Yeşil     | Normal   | İlkbahar |

İkinci aşamada Tablo 3.10'da kullanılan karakteristik değerlerinden tek başlarına hiçbir kural oluşturamayanlar Tablo 3.13'te verilmiştir.

Tablo 3.13. Birinci aşamadan sonra kural çıkarılamayan değerler listesi

| Karakteristik | Değer             |
|---------------|-------------------|
| Hava          | Yağmurlu, Güneşli |
| Ağaçlar       | Yeşil, Yapraksız  |
| Sıcaklık      | Normal            |

İkinci aşamada kural çıkaramayan değerlerin ikili kombinasyonları dikkate alınır. Bu amaçla Tablo 3.13'teki tüm değerlerin ikili kombinasyonları; {Yağmurlu, Yeşil}, {Yağmurlu, Yapraksız}, {Yağmurlu, Normal}, {Güneşli, Yeşil}, {Güneşli, Yapraksız}, {Güneşli, Normal}, {Yeşil, Normal} ve {Yapraksız, Normal} şeklindedir.

Bu kombinasyonların Tablo 3.9'daki örnekler setine uygulanması sonucunda {Güneşli, Yapraksız} sadece Kış mevsiminde, {Yeşil, Normal} sadece İlkbahar mevsiminde ve {Yapraksız, Normal} ise sadece Sonbahar mevsiminde geçmektedir. Buna göre ikinci aşamada çıkarılan kurallar listesi Tablo 3.14'te verilmiştir.

Tablo 3.14. İkinci aşama ile elde edilen kurallar

| Kural No | Kural Tanımı                                                  |
|----------|---------------------------------------------------------------|
| 1        | EĞER Hava Güneşli VE Ağaçlar Yapraksız İSE Mevsim Kış         |
| 2        | EĞER Yapraklar Yeşil VE Sıcaklık Normal İSE Mevsim İlkbahar   |
| 3        | EĞER Ağaçlar Yapraksız VE Sıcaklık Normal İSE Mevsim Sonbahar |

Buna göre tüm örnekler sınıflandırılmış olmaktadır. Tablo 3.9'daki örnek seti kullanılarak oluşturulan tüm kurallar Tablo 3.15'te verilmiştir.

Tablo 3.15. Mevsimler eğitim seti için Rules-1 tarafından üretilen kurallar

| Kural No | Kural Tanımı                                                  |
|----------|---------------------------------------------------------------|
| 1        | EĞER Hava Karlı İSE Mevsim KİŞ                                |
| 2        | EĞER Ağaçlar Sarı İSE Mevsim Sonbahar                         |
| 3        | Eğer Sıcaklık Düşük İSE Mevsim KİŞ                            |
| 4        | EĞER Sıcaklık Yüksek İSE Mevsim YAZ                           |
| 5        | EĞER Hava Güneşli VE Ağaçlar Yapraksız İSE Mevsim Kış         |
| 6        | EĞER Yapraklar Yeşil VE Sıcaklık Normal İSE Mevsim İlkbahar   |
| 7        | EĞER Ağaçlar Yapraksız VE Sıcaklık Normal İSE Mevsim Sonbahar |

### 3.5.2. RULES-2 algoritması

RULES-2 algoritması, RULES-1 algoritmasının gelişmiş halidir. Kural çıkarma prosedürünün RULES-1'den farkı; RULES-2 her aşamada sınıflandırılmamış tüm örneklerdeki karakteristik değerlerini göz önüne almak yerine, sınıflandırılmamış örnekler setinden rasgele seçilen bir örneğin karakteristik değerlerini göz önüne alır. RULES-2 algoritmasının RULES-1 algoritmasına göre üstünlükleri aşağıda verilmiştir[59]:

#### a) Daha kısa işlem süresi

RULES-1'de kural çıkarma işlemi RULES-2'ye göre daha uzundur. Bunun temel sebebi RULES-1'in her aşamada değerler dizisindeki tüm değerleri ve bu değerlerin tüm kombinasyonlarını göz önüne almasındandır. Karakteristik sayısının fazla ve bunlara ait değerlerin de çok sayıda olması durumunda kural çıkarma için gerekli işlem süresi uzayacaktır. RULES-2'de bu problemi aşmak için her defasında sadece seçilen örnekte yer alan değerler ve bu değerlerin kombinasyonları dikkate alınır.

#### b) Gereksiz şartların önlenmesi

RULES-2'de her defasında sadece bir örnek dikkate alındığından, çıkarılacak hiçbir kural gereksiz şart içermeyecektir. RULES-1'de ise her defasında sınıflandırılmamış tüm örneklerle ait değerler ele alındığından gereksiz şartların kullanılması kaçınılmaz olmaktadır. RULES-1 bu hatayı gidermek için çıkarılan kuralları tekrar kontrol etmektedir. Ancak RULES-2'de bu kontrol işlemine gerek kalmamaktadır.

#### c) Eksik değerli örnekleri işleyebilmesi

Gerçek problemlerde, bazı karakteristiklere ait değerler belli olmayabilir (bilinmeyebilir veya olmayabilir). Bu durumda RULES-1 algoritması bu örnekler için kural çıkarmaz. RULES-2 ise bilinmeyen karakteristik değerlerini kural çıkarma işlemi sırasında göz önüne almayarak bu örnekler için de kural çıkarılmasını sağlar.

#### **d) Sayısal değer içeren karakteristikleri işleyebilme**

RULES-2, bir karakteristiğin alabildiği değerlerin tümü sayısal ise bu karakteristikleri sayısal karakteristik olarak sınıflandırır. Özellikle mühendislik problemlerinde bu sayısal karakteristikleri kullanmak kaçınılmazdır. RULES-2 bu tür karakteristikleri de işleyerek kural çıkarabilir. Şayet bir örnekler setinde sayısal karakteristikler varsa bu değerlerin en küçük ve en büyük olanlarını bularak, kullanıcının bu karakteristik için tanımladığı aralık sayısına göre örnekler setini yeniden düzenler. Bu durumda elde edilen kurallar daha genel olacaktır.

#### **3.5.3. RULES-3 algoritması**

Rules-3 algoritması[59,62], Rules-2 algoritmasının gelişmiş bir versiyonudur. Rules-3'ün kural çıkarma prosedürü Rules-2 ile aynıdır. Bu algoritmanın Rules-2'ye göre iki üstünlüğü vardır:

##### **1) Daha genel kurallar elde etme**

Rules-2 algoritması çıkarılan kuralların daha genel olup olmadıklarını ve çıkarılabilecek daha genel kuralların bulunup bulunmadığını kontrol etmez. Bu eksikliği gidermek için Rules-3 algoritması elde edilen kuralları tek tek kontrol ederek en genel olanlarını seçerek daha az kural çıkarılmasını sağlar. Bu işlemi eldeki örnekleri sınıflandırabilen kurallar çıkarıldıktan sonra bunların her birinin verilen örnekler setinde aynı sınıfta geçen kaç örneği sınıflandırdığına bakarak en fazla örneği sınıflandırabilen kuralı seçerek gerçekleştirir. Bu şekilde aynı örnekleri sınıflandırabilen kurallardan daha az örneği sınıflandıran kural veya kuralları iptal ederek en genel olanını seçer [59].

##### **2) Her örnek için çıkarılabilecek kural sayısının belirlenmesi**

Rules-2 algoritmasında çıkarılacak kurallar üzerinde bir kontrol söz konusu değildir. Örneğin, bir örnek için çıkarılabilecek kural sayısı belirlenemez. Algoritma daima çıkarılabilecek maksimum sayıda kural çıkarır. Rules-3 algoritması kullanıcının

tanımlayacağı sayıda kural çıkarabilir. Bu şekilde daha fazla kural çıkarılması engellenmekte ve işlem zamanı da kısalmaktadır. Her bir örnek için en az bir en fazla belirtilen sayı kadar kural çıkarılmaktadır. Özellikle karakteristik sayısının ve bunlara ait değerlerin fazla sayıda olması durumunda yararlı bir işlemdir [59].

Rules-3 algoritması kullanılarak Tablo 3.9’da verilen örnek seti için Tablo 3.16’daki kurallar elde edilir.

Tablo 3.16. Mevsimler eğitim seti için Rules-3 tarafından üretilen kurallar

| Kural No | Kural Tanımı                                                  |
|----------|---------------------------------------------------------------|
| 1        | EĞER Ağaçlar Yapraksız VE Sıcaklık Normal İSE Mevsim SONBAHAR |
| 2        | EĞER Ağaçlar Yeşil VE Sıcaklık Normal İSE Mevsim İLKBAHAR     |
| 3        | EĞER Sıcaklık Düşük İSE Mevsim KIŞ                            |
| 4        | EĞER Sıcaklık Yüksek İSE Mevsim YAZ                           |
| 5        | EĞER Ağaçlar Sarı İSE Mevsim SONBAHAR                         |

#### 3.5.4. RULES-3 PLUS ve RULES-4 algoritmaları

Rules-3 Plus algoritması[96], RULES ailesi temel öğrenme algoritmalarının yeni bir versiyonudur. Rules-3 Plus’ın üstün olan yeni iki özelliği vardır. Birincisi, Rules-3 Plus, Rules-3’teki geniş arama yönteminin yerine, daha verimli bir kural arama yöntemi kullanır. İkincisi, genellik ve doğruluk derecesi daha yüksek olan kurallar için bir ölçü kullanır. Bu ölçüye göre kural adaylarını sıralama ve seçme işlemini gerçekleştirir. Rules-3 böyle bir ölçü kullanmaz. Rules-3’te örnekteki tüm değerlerin kombinasyonlarına bakılır.

Rules-3 Plus algoritmasında bir karakteristik-değer çifti bir şart meydana getirir. Karakteristiklerin sayısı  $n_a$  ise, bir kural 1 ile  $n_a$  arasında şart içerebilir. Birden fazla şartın bir kuralda birleşmesine izin verilir, ancak bir kuraldaki karakteristik-değer çiftleri birbirinden farklıdır. Bu yüzden kural, birden daha fazla şart içerirse, bu durumda karakteristikler farklı olmalıdır. Ele alınan her bir örnek için kural oluşturma prosedürü en fazla  $n_a$  kadar sayıda işlem yapar. Birincisinde tek şartlı kural üretilir. İkincisinde iki şartlı kural üretilir ve işlem tüm karakteristikler için kural üretilinceye kadar devam eder.

RULES-3 Plus, ele alınan örnekteki karakteristik değerlerinin kümesini SETAV (SET of Attribute-Value) olarak adlandırır. SETAV ele alınan örnekteki karakteristik-değer çiftlerinden oluşur. SETAV'daki elemanların toplam sayısı, örnekteki karakteristiklerin sayısına eşittir. İlk iterasyonda kümenin her bir elemanı koşul olarak bir kural oluşturup oluşturmadığına karar vermek için tüm örnek seti ile karşılaştırılarak kontrol edilir. Örnek seti için verilen bir eleman sadece bir sınıfa ait ise, bu durumda bu eleman bir kural oluşturur. Eğer eleman birden daha fazla sınıfta geçiyorsa PRSET (Partial Rules SET) olarak adlandırılan kümeye ilave edilir. PRSET'teki ifadelerin maksimum sayısına kullanıcı tarafından karar verilir. PRSET'e eklenecek yeni ifadelerin mevcut ifadelerden farklı olması gerekir.

Rules-3 Plus kural oluşturma işlemi sırasında ifadelerin bilgi içeriğini değerlendirmek için  $H$  ölçüsü olarak adlandırılan (3.1) ifadesini kullanır.

$$H = \sqrt{\frac{E^c}{E}} \left[ 2 - 2\sqrt{\frac{E_i^c E_i}{E^c E}} - 2\sqrt{\left(1 - \frac{E_i^c}{E^c}\right) \left(1 - \frac{E_i}{E}\right)} \right] \quad (3.1)$$

Burada;

$E^c$  İfade tarafından kapsanan örneklerin sayısı (verilen kural tarafından doğru veya yanlış sınıflandırılmış örneklerin toplam sayısı)

$E_i^c$  Sınıf  $i$ 'ye ait olan ve ifade tarafından kapsanan örneklerin toplam sayısı (verilen kural tarafından doğru olarak sınıflandırılmış örneklerin sayısı).

$E_i$  Hedef sınıfına ait olan eğitim setindeki örneklerin toplam sayısıdır.

(3.1) denkleminde verilen ilk terim kuralın genelliği ile ilgilidir (3.2).

$$G = \sqrt{\frac{E^c}{E}} \quad (3.2)$$

(3.1) denkleminde verilen ikinci terim ise kuralın doğruluğu ile ilgilidir (3.3).

$$A = 2 - 2\sqrt{\frac{E_i^c E_i}{E^c E}} - 2\sqrt{\left(1 - \frac{E_i^c}{E^c}\right)\left(1 - \frac{E_i}{E}\right)} \quad (3.3)$$

Kural oluşturma işlemi boyunca PRSET'teki ifadeler  $H$  ölçüsüne göre sıralanırlar. Verilen iterasyonda çıkarılan kuralların sayısı birden fazla ise  $H$  değeri en yüksek olan kural seçilir ve diğerleri atılır. Yeni bir örneği sınıflandırırken Rules-3 Plus tarafından belirlenen kuralı kullanırken mümkün üç sonuç vardır. Bunlar:

- Sadece bir kural yeni örneği kapsar ve örnek kural tarafından sınıflandırılır.
- Örneği birden fazla kural kapsar. Bu durumda  $H$  değeri en yüksek olan kural örneği sınıflandırmak için kullanılır.
- Hiçbir kural örneği sınıflandırmaz veya örnek yanlış sınıflandırılır. Bu durumda örnek eğitim setine eklenmeli ve endüktif süreç tekrar başlatılmalıdır.

Rules-3 Plus algoritmasının kural çıkarma prosedürü adım adım Şekil 3.11'te gösterilmiştir[96].

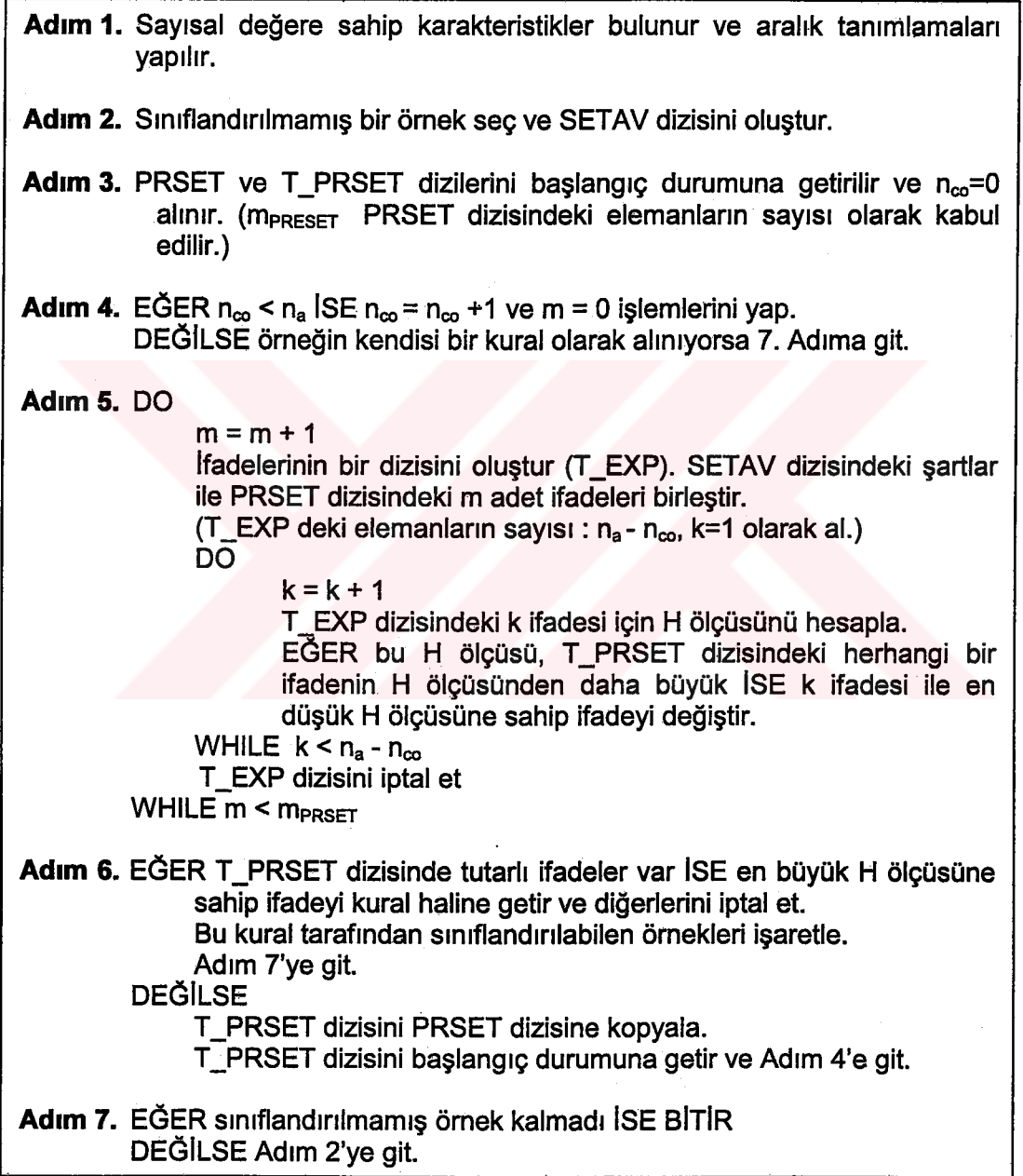
Bu algoritma RULES-2 ve RULES-3'te olduğu gibi sayısal karakteristikleri sınıflandırma özelliğine sahiptir. RULES-3 ile Rules-3 Plus arasındaki en temel fark; her bir iterasyonda şartların mümkün olan tüm kombinasyonlarını dikkate almak yerine, sadece PRSET'te tanımlanmış olan ifadeleri dikkate almasıdır. Rules-3 ve bundan önceki algoritma olan Rules-2'de her bir örnek için karakteristik-değer çiftlerinin mümkün olan bütün kombinasyonlarının sayısı denklem (3.4) ile bulunur.

$$N_{\max} = \sum_{i=1}^{n_a} \frac{n_a!}{(n_a - i)! i!} \quad (3.4)$$

Rules-3 Plus algoritmasında bu sayı denklem (3.5) ile bulunur.

$$N'_{\max} = n_a + m_{PRSET} \sum_{i=1}^{n_a-1} (n_a - i) \quad (3.5)$$

Görüldüğü gibi  $N_{\max}$  değeri  $N'_{\max}$  değerinden daha büyüktür. Ayrıca bu değerler, her bir iterasyonda en uygun kuralı bulmak için dikkate alınacak ifadelerin sayısından çok daha büyüktür.



Şekil 3.11. Rules-3 Plus algoritması

Algoritmada kullanılan bazı sembollerin görevleri aşağıda verilmiştir:

|             |                                                              |
|-------------|--------------------------------------------------------------|
| $n_{co}$    | : Şartların sayısı,                                          |
| $n_a$       | : Karakteristiklerin sayısı                                  |
| $m_{PRSET}$ | : PRSET dizisindeki elemanların sayısı.                      |
| $T\_PRSET$  | : PRSET gibi aynı boyuta sahip, geçici bir dizi değişkendir. |
| $T\_EXP$    | : İfadelerin oluşturduğu geçici dizi değişken.               |

Rules-4 algoritmasının Rules-3 Plus'tan farkı; kurallar üretildikten sonra eğitim setinde az sayıda örnek için üretilen kuralların iptal edilmesidir. Bundan dolayı eğitim setindeki örnekler için doğruluk oranı düşmektedir. Kabul edilebilir bir hata oranında kural çıkarılmaktadır. Bu özellik C.45Rules ve C4.5Prune algoritmalarında da kullanılmaktadır.

### 3.6. AQ ailesi algoritmaları

Bir endüktif öğrenme algoritması olan AQ ilk olarak Michalski tarafından geliştirilmiştir[6, 7, 10]. AQ'de karar kuralları EĞER-İSE şeklinde olmayıp *Değişken-değerli Mantık Sistemi 1 (VL1)* formunda ifadeler olarak gösterilir. VL1'de bir seçici, bir değer veya ilişki operatörlerinden ( $<$ ,  $\neq$ ,  $\leq$ ,  $\geq$ ,  $>$ ,  $=$ ) birini kullanarak değerleri ayırtmak için bir değişken veya karakteristik ile bağlantı kurar. Seçicilerin birleşimi karmaşık bir yapıdadır. Aşağıda iki seçicinin karmaşık yapısı görülmektedir. Bu yapıda *havanın yağmurlu* veya *bulutlu* olması ve *sıcaklığın 28* derecenin altında olması durumu görülmektedir [99].

(Hava = Yağmurlu veya Bulutlu) ve (Sıcaklık<28)

Herhangi bir sınıf için eğitim setinde ilgili sınıfı içeren örnekler pozitif set ve diğer tüm örneklerin oluşturduğu set ise negatif set olarak kabul edilir. Bu şekilde her örnek pozitif veya negatif sette olacak şekilde ayrıştırılır. AQ'de her bir sınıfı ve sınıfın karar kuralı ile ilgili şart bölümünü ayırmak için bir form kullanılır. Aşağıdaki ifade karar kuralına bir örnektir:

(Hava = Güneşli veya Bulutlu) ve (Sıcaklık > 28)  
veya (Rüzgar = Kuvvetli) ve (Sıcaklık > 33) => Sınıf (İYİ)

AQ ailesi algoritmalarından olan AQ15 algoritması[12], AQ algoritmasının gelişmiş bir şekli olup endüktif öğrenmenin STAR[100] metodunu kullanır. STAR metodu pozitif sınıfta olup negatif sınıfta olmayan karakteristik-değer çiftlerini gruplamak için kullanılır.

AQ algoritmalarında kullanılan bazı kavramlar aşağıda verilmiştir:

**Seçici (Selector):** Bir veya birkaç karakteristik ile bunların alabilecekleri değerleri gösterir. Örneğin, [*Hava=Yağmurlu,Karlı,Güneşli*], *Hava* karakteristiğini ve bunun alabileceği değerleri gösteren bir seçicidir.

**Birleşik Kavram (Conjunctive concept):** Seçici gruplarına verilen isimdir. Burada iki seçici aynı karakteristiğe ait olamaz.

**Tanım (Description):** Bir karakteristik-değer grubuna verilen isimdir.

**Tam tanım (Complete description):** Pozitif sınıfa ait olan tüm örnekleri tanımlayan karakteristik-değer gruplarıdır. Her defasında eldeki örneğin ait olduğu sınıf *pozitif sınıf* diğerleri ise *negatif sınıf* olarak değerlendirilir.

**Tutarlı tanım (Consistent description):** Negatif sınıfa ait hiçbir sınıfta geçmeyen karakteristik-değer gruplarıdır.

**Yıldız (Star):** Seçilen pozitif sınıftaki örneği tanımlayan fakat negatif sınıfa ait hiçbir örneği tanımlamayan karakteristik-değer gruplarıdır.

**Çekirdek (Seed):** Kural çıkarmak için seçilmiş örnektir.

**Karşıtı geliştir (Extend against):** AQ ailesi algoritmaların kullandığı bir prosedür olup, seçilen örnekte yer alan bir karakteristik değerine karşı, negatif sette yer alan

bir örnekte kullanılmış olan aynı karakteristiğe ait değerin olası karşıt değerini bulur. Şayet değer, seçilen örnek ve negatif sınıftaki bir örnek için aynı ise karşıt değer geliştirilemez, aksi halde negatif sınıfta yer alan değerin dışında söz konusu karakteristiğin alabileceği tüm olası değerler karşıt değer olarak kabul edilir.

AQ algoritmasının temel kural çıkarma mantığı Şekil 3.12'de adım adım gösterilmiştir.

|                |                                                                                                                                                                                                                                                                                                                  |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Adım-1.</b> | Örnekler setinden bir örnek seç. Bu örneğin ait olduğu sınıf <i>Pozitif sınıf</i> ve diğer tüm sınıflar ise <i>Negatif sınıf</i> olarak kabul et.                                                                                                                                                                |
| <b>Adım-2.</b> | Karşıtını geliştir prosedürü ile bir <i>Yıldız</i> bul.                                                                                                                                                                                                                                                          |
| <b>Adım-3.</b> | <i>Yıldız'daki</i> tanımlar (karakteristik-değer grupları) için <i>Tutarlı Tanım</i> ve <i>Tam Tanım</i> değerlerini bul.                                                                                                                                                                                        |
| <b>Adım-4.</b> | Eğer en az bir <i>Tanım</i> için <i>Tutarlı Tanım</i> değeri sıfır ise Adım-5' git. Şayet hiçbir <i>Tanım</i> için bu değer sıfır değilse, bu tanımların kombinasyonlarını bul ve aynı değerleri hesapla. Şayet mümkün olan tüm kombinasyonlar için de <i>Tutarlı Tanım</i> değeri sıfır olmuyorsa Adım-1'e git. |
| <b>Adım-5.</b> | Tüm tanımları VL haline getir. Bu kuralların kapsadığı tüm örnekleri örnek setinden çıkar.                                                                                                                                                                                                                       |
| <b>Adım-6.</b> | Eğer geriye örnek kalmadıysa Bitir aksi halde Adım-1'e git.                                                                                                                                                                                                                                                      |

Şekil 3.12. AQ algoritması

AQ algoritması kullanılarak Tablo 3.9'da verilen örnek seti için Tablo 3.17'de verilen kurallar elde edilir.

Tablo 3.17. AQ tarafından elde edilen kurallar (Mevsimler eğitim seti)

| <b>Kural</b> | <b>Kural Tanımı</b>                                                              |
|--------------|----------------------------------------------------------------------------------|
| 1            | EĞER Sıcaklık=Normal VE Ağaçlar=Yeşil VEYA Ağaçlar=Yapraksız İSE Mevsim=Sonbahar |
| 2            | EĞER Sıcaklık=Düşük İSE Mevsim=Kış                                               |
| 3            | EĞER Sıcaklık=Yüksek İSE Mevsim=Yaz                                              |
| 4            | EĞER Sıcaklık=Normal VE Ağaçlar=Yeşil İSE Mevsim=Yaz                             |

Tablo 3.17'deki birinci kuraldan da görüldüğü gibi, AQ ailesi algoritmaları VL1 formunda kurallar üretirler. Üretilen kurallarda VE, VEYA operatörleri ile beraber  $\neq$  (Farklı) gibi ilişki operatörleri de bulunabilir. Ancak diğer tüm algoritmalar sadece VE mantık operatörünü kullanarak kural ürettiklerinden dolayı bu algoritmaları diğer algoritmalarla, kural sayılarına göre test etmek güçtür.

AQ algoritmasının geliştirilmiş birçok versiyonu vardır. Michalski ve Larson AQ ile aynı prosedürü kullanan, ancak yeni örnekler verildiğinde daha önce çıkarılmış olan kuralları düzelterek yeni kural seti oluşturabilen AQ11[102] algoritmasını geliştirmişlerdir. Daha sonra AQ14-NT[101] ve CN2[10,11] algoritmaları geliştirilmiştir.

Hong, Mozatic, Michalski ve Lavrac daha sonra AQ15[12] algoritmasını geliştirmişlerdir. Q15, AQ11 algoritmasının gelişmiş bir versiyonudur. En önemli özelliği gürültülü ve çakışan verileri de işleyebilmesi ve 58'den fazla karakteristik içeren örnek setlerini de çözebilmesidir. Önceki algoritmalar Pascal dili kullanılarak geliştirilmişlerdir. Pascal'ın küme tanımlamalarından kaynaklanan eksiklikleri ANCI-C kullanılarak giderilmiştir. Bu amaçla AQ15c programı geliştirilmiştir.

1993 yılında, genetik algoritmalar tekniğini kullanarak kural üreten AQ15-GA algoritması geliştirilmiştir[102].

AQ17 ailesinin diğer algoritmaları olarak AQ17-DCI, AQ17-FCLS, AQ17-HCI, AQ17 [103] ve AQ17-PRE[104] sayılabilir. AQ17-FCLS algoritmasının en büyük özelliği sayısal verileri de işleyebilmesidir. Diğer algoritmalarda bu özellik bulunmamaktadır[105]. AQ ailesinin son algoritmaları ise AQ18[105, 106], AQ20 [107]'dir.

### 3.7. C4.5 algoritması

C4.5 endüktif öğrenme algoritması Quinlan tarafından geliştirilmiştir [17]. ID3 algoritması gibi yukarıdan-aşağıya doğru karar-ağacı üretir. Günümüzde en çok bilinen ve en çok kullanılan algoritma olarak kabul edilmektedir. ID3 algoritması

karar ağacının her düğümü için bilgi kazancı değerlerini hesaplar. C4.5 ise bilgi kazancı ile beraber alt sette yer alan karakteristiklerin bilgi kazanç oranlarını da hesaplayarak bilgi kazanç oranı en yüksek olan karakteristiği düğüm noktası olarak seçer. Karar ağacının her dalı sadece bir tek sınıfa karşılık gelinceye kadar işlemleri sürdürür. C4.5 algoritması karar ağacı oluşturma prosedürü, düğüm noktalarının belirlenmesi dışında ID3 ile aynıdır. Daha sonra karar ağacını kural setine dönüştürür.

C4.5'in C4.5Rules[108] ve C4.5Prune[108] versiyonları oluşturulan karar ağacında özel kural oluşturacak olan dalları iptal ederek daha az sayıda ve genel kurallar elde edilmesini sağlar. Bu özelliğinden dolayı eğitim setini belli bir hata oranına göre sınıflandırır.

### 3.8. J-Pruned algoritması

J-Pruned algoritması[109] 2002 yılında geliştirilmiştir. ID3 ve C4.5 algoritmalarının temel kural çıkarma mantığı üzerine geliştirilen bir algoritmadır. Kural çıkarma işlemi sırasında entropi ve bilgi kazancı yerine *J ölçüsü* olarak adlandırılan *çapraz-entropi* ölçüsünü kullanır. Çapraz-entropi (3.6) denklemi yardımıyla hesaplanır [110,111].

$$j(X;Y=y) = p(x|y) \log_2 \left( \frac{p(x|y)}{p(x)} \right) + (1 - p(x|y)) \log_2 \left( \frac{(1 - p(x|y))}{(1 - p(x))} \right) \quad (3.6)$$

Düğüm noktalarını en yüksek *j* değerine sahip karakteristiklere göre seçer.

### 3.9. CART ve OC1 algoritmaları

1984 yılında Breiman tarafından CART (Classification and Regression Trees) isimli algoritma geliştirilmiştir [18]. CART sayısal verileri işlemek üzere tasarlanmıştır. Bu algoritma Böl-ve-Fethet endüktif öğrenme metoduna göre işlem yapmakta ve sadece ikili karar ağacı oluşturmaktaydı. Bu algoritmada bir  $\wedge$  düğüm noktasında örnek setini ayırtmak için bir  $i(\wedge)$  ölçüsü kullanılmaktadır.  $i(\wedge)$  ölçüsü;

$$i(\wedge) = \sum_{i \neq j} p(j | S) p(k | S)$$

şeklinde tanımlanmaktadır. Burada,  $S \wedge$  düğümündeki seti,  $j$  ve  $k$ ,  $S$  setindeki sınıfları,  $p(j | S)$  ise  $S$  setindeki  $j$  sınıfı örneklerinin olasılığını gösterir.

$S$  sayısal değerlerden oluşan bir örnek seti,  $C = \{C_1, C_2, \dots, C_n\}$  karakteristikler seti ve  $n$  karakteristiklerin sayısını gösterebilir. CART algoritmasının öğrenme prosedürü aşağıdaki gibi açıklanabilir:

CART,  $\wedge$  düğümünde tüm karakteristikleri birer birer araştırır. Her bir karakteristik için, en çok katkıyı sağlayan en iyi ayrıştırmayı bulur.  $n$  adet aday içerisinde en iyi ayrıştırmayı seçer. Düğümdeki en iyi ayrıştırma  $i(\wedge)$  ölçüsüne bağlıdır. Seçilen karakteristiği bulunan nokta için düğüm olarak kabul eder.

Ardından, Crawford, CART algoritmasının bir uzantısı olan ve eğik ayrışmayı sağlayan OC1 [52]'i geliştirmiştir. Bu algoritma ile elde edilen karar ağacı koordinat eksenine paralel değil eğik bir şekildedir. Bu algoritma ile daha küçük karar ağaçlarının oluşturulması mümkün olmaktadır[3].

### 3.10. PRISM Algoritması

Doğrudan kural çıkaran bir algoritmadır. Hedef sınıf için olasılığı en yüksek olan karakteristik-değer çiftlerinin bir kombinasyonunu seçerek eğitim setinden doğrudan kural çıkarır. Bu algoritma Şekil 3.13'teki yapıya göre kural çıkarma işlemini gerçekleştirir[16, 119].

PRISM algoritması, ilk önce sınıflardan herhangi birisini seçer. Seçilen bu sınıfa karşılık gelen örneklerden bir alt set oluşturur. Bu alt sette yer alan karakteristik-değer çiftlerinin olasılıklarını hesaplar. En yüksek olasılığa sahip karakteristik-değer çifti seçilir. Eğer bu seçilen karakteristik-değer çifti sadece bir tek sınıfta geçiyorsa kural oluşturur, aksi halde bu değeri içeren örneklerden yeni bir alt set oluşturulur.

Bu alt sette yer alan karakteristik-değer çiftlerinin olasılıkları hesaplanır ve en yüksek olasılık değerine sahip karakteristik-değer çifti seçilir. Bu aşamaya kadar seçilen iki karakteristik-değer çifti sadece bir tek sınıfta geçiyorsa bunlar kural haline getirilir. Aksi halde işlem bu şekilde devam eder[16, 119].

1'den  $n$ 'ye kadar olan sınıflardan her defasında biri ( $i$ ) seçilir. ( $n$  : Sınıf sayısı)

**Adım-1 :**  $i$ . sınıftaki her bir karakteristik-değer çifti için olasılıklar hesaplanır.

**Adım-2 :** Olasılığı en yüksek olan karakteristik-değer çifti seçilir ve seçilen kombinasyonu içeren (tüm sınıflar için) örneklerden oluşan bir alt set oluşturulur.

**Adım-3 :** 1 ve 2 nolu altset sadece  $i$ . sınıfın örneklerini içerinceye kadar adımlar tekrarlanır. Bu altseti oluşturmada seçilmiş olan bütün karakteristik-değer çiftleri birleştirilerek kural oluşturur.

**Adım-4 :** Eğitim setinden bu kural tarafından sınıflandırılan tüm örnekler silinir. 1'den 4'e kadar olan adımlar,  $i$ . sınıfın bütün örnekleri silininceye kadar tekrarlanır.

**Not :** Her bir sınıf için eğitim seti yenilenir.

Şekil 3.13. PRISM algoritması

İlk kural oluşturulduktan sonra bu kural ile sınıflandırılan örnekler işaretlenir. Geriye kalan örnekler için yeni kurallar araştırılır. Bu işlem alt setteki tüm örnekler işaretleninceye kadar devam eder. Tüm örnekler işaretlendikten sonra ikinci sınıf ele alınır. Bu sınıf için de aynı işlemler tekrarlanır. İşlemler tüm sınıflar için yapıldıktan sonra kural çıkarma işlemi son bulur[16, 119].

### 3.11. Sonuç

Karar ağacı üreten algoritmalar, ağacın düğüm noktalarını belirlemek için çeşitli ölçüler kullanırlar. Örneğin, ID3 ailesi algoritmaları entropi ölçüsünü, C4.5 algoritmaları bilgi kazanç oranını, CART algoritması  $i(\wedge)$  ölçüsünü ve J-Pruned algoritması  $j$  ölçüsünü kullanır. Bu ölçüler sayesinde karar ağacının daha kısa ve genel bir yapıda oluşması sağlanır. Ancak karar ağacını kural setine dönüştürmenin ve çok sayıda karakteristik değer içeren örnek problemleri karar ağacı şeklinde göstermenin zorlukları açıktır.

Doğrudan kural üreten algoritmaların büyük bir kısmı arama metodlarına göre kural üretmektedir (Rules). Diğer bir kısmı ise değerlere ait olasılık değerlerine göre seçim yapmaktadır (Prism, AQ Ailesi, Rules-3 Plus, Rules-4).

Doğrudan kural çıkaran mevcut algoritmalar, örnek setindeki karakteristik ve değerlere ait bilgi değerini ölçmezler. Bundan dolayı üretilen kuralların sayısı daha fazla olmakta ve genel kurallar yerine özel kurallar üretilmektedir. Oysa doğrudan kural üreten algoritmalarda, entropi ve bilgi kazanç oranları kullanılarak daha genel ve doğruluk oranı daha yüksek olan algoritmalar hazırlanabilir.

Bu amaçla, üç yeni algoritma geliştirilmiştir: REX-1, REX-2 ve REX-3. REX-1 ve REX-2 algoritmaları temel olarak entropi ve bilgi kazanç ölçülerini kullanarak kural çıkarmaktadırlar. REX-3 algoritması ise sadece arama metodunu kullanmakta ancak en çok örneği sınıflandırabilen kuralları seçmektedir.

## BÖLÜM 4. REX-1 : ENDÜKTİF ÖĞRENME ALGORİTMASI

### 4.1. Giriş

Bu bölümde yeni geliştirilen bir endüktif öğrenme algoritması olan REX-1 (REX: **R**ule **E**xtraction) sunulacaktır. Bu algoritma, verilen örnek setinden genelleştirilmiş EĞER-İSE (IF-THEN) kuralları elde etmek için geliştirilmiştir. Kural çıkarma işlemini kapsama (doğrudan kural çıkarma) metodunu kullanarak yerine getirmektedir. Bu algoritma, RULES ailesi algoritmalarının zayıf yönlerini ortadan kaldırmaktadır. Rules-3, Rules-3 Plus ve Rules-4 algoritmalarında kullanılmayan *entropi* yardımıyla kural çıkarma prosedürü sırasında önem derecesi yüksek olan karakteristiklere öncelik verilmesi ve daha genel kurallar elde edilmesi amaçlanmaktadır. Gerçek hayatta kullanılan büyük veri setleri kullanılarak algoritmanın diğer algoritmalarla göre üstünlükleri anlatılacaktır.

Algoritma bir örnek problem yardımıyla adım adım anlatılacak ve ardından farklı algoritmalarla karşılaştırılacaktır.

### 4.2. Algoritmanın Tanımı

Geliştirilen algoritma karar ağacı ve doğrudan kural üreten algoritmaların bazı ortak yönlerini kullanarak örnek setinden kurallar elde edilmesini sağlar.

Karar ağacı üreten algoritmaların en iyi bilinenleri ID3 ve C4.5'tir. Bu algoritmalar verilen örnek setinden genelleştirilmiş kurallar çıkarmak için öncelikle her karakteristiğe ait değerlerin ve karakteristiğin entropisini hesaplayarak karar ağacı oluştururlar. Hesaplanan bu entropi değerlerine göre ayrı ayrı bilgi kazançları hesaplanır. Bilgi kazancı en yüksek olan karakteristik ağacın kökü olarak seçilir. Diğer karakteristikler buna göre yeniden düzenlenir. Aynı işlem diğer alt setler için

de tekrarlanır. En son aşamada oluşan karar ağacına göre kurallar çıkarılır. C4.5 ve ID3 gibi algoritmalarda olduğu gibi, her aşamada yeni bir karar ağacı oluşturmak ve örnekleri bu karar ağacının köküne göre alt setlere ayırmak çok zor bir işlemdir. Bu algoritmada işlemler daha basit bir kural çıkarma prosedürü kullanılarak yerine getirilmektedir. ID3 ve C4.5 algoritmaları karar ağacı oluşturduktan sonra, karar ağacını kural setine dönüştürürler. REX-1 algoritması entropi ölçüsünü kullanarak ve karar ağacı oluşturmada doğrudan kural üretmektedir.

RULES-3 algoritması verilen örnek setindeki örnekleri sırayla ele alır. Her örnekteki karakteristik değerlerinin ilk önce tekli kombinasyonlarını dikkate alır ve bu tekli değerlerden herhangi birisi bütün örnek setinde sadece bir sınıfta geçiyorsa bu değer kural haline getirilir. Aksi halde ikili kombinasyonlara bakılır. Bu kombinasyonlardan sadece bir sınıfta tekrar edilenleri ile kural oluşturulur. Bu işlem karakteristik sayısı kadar kombinasyona ulaşıncaya kadar tekrar edilir. Her aşamada çıkarılan kural ile sınıflandırılabilen örnekler ayıklanır. RULES-3 algoritması ile mümkün olan tüm kurallar çıkarılır. Ancak karakteristiklere ait bilgi değerleri göz önüne alınmadığından çıkarılan kural sayısı fazla olabilmekte ve kurallarda bilgi değeri küçük olan karakteristikler de yer alabilmektedir.

REX-1 algoritması ilk önce örnek setinde bulunan karakteristik ve bu karakteristiklere ait değerlerin entropilerini hesaplayarak, düzensizliği az olan karakteristiklere öncelik vererek kural çıkarma işlemini gerçekleştirir. Entropisi düşük olan karakteristiklerin bilgi kazancı yüksektir. Böylece örnek setinde bilgi değeri en büyük olan karakteristiklere öncelik verilir. Ancak RULES-3 algoritması örnek seti üzerinde böyle bir inceleme yapmadan ilk örnek ve ilk karakteristikten başlayarak kural çıkarma işlemini gerçekleştirir. Bunun birçok sakıncaları vardır. Örnek setinde önemi olmayan bir karakteristiğe göre kurallar üretilmesi kaçınılmaz olmaktadır.

#### **4.2.1. REX-1 algoritmasının özellikleri**

REX-1 algoritmasının temel özellikleri aşağıdaki gibi açıklanabilir:

- Entropi ve düzeltilmiş entropi değerlerini kullanma.
- Kısa işlem süresi.
- Eksik değerli örnekleri işleyebilme.
- Sayısal değer içeren karakteristikleri işleyebilme.

#### 4.2.1.1. Entropi ve düzeltilmiş entropi değerlerini kullanma

Karakteristik ve değerlere ait entropileri hesaplayarak bilgi kazancı daha yüksek olan değerlere göre kural çıkarılmaktadır. Örnek seti için çok düzensiz ve önem derecesi az olan karakteristikler ya çok az dikkate alınacak yada hiç dikkate alınmayacaktır. Başka ifadeyle, karakteristikler önem derecelerine göre kurallarda yer alacaktır. Bu ise üretilen kuralların daha genel olmasını sağlayacaktır.

Karakteristik ve değerlere ait entropinin nasıl hesaplanacağı Bölüm-2’de anlatılmıştır. Karakteristiklere ait düzeltilmiş entropi, her karakteristiğin entropisinin ilgili karakteristikteki değer sayısı ile çarpılması sonucu elde edilir. Bu işlem için kullanılacak denklem (4.1)’de verilmiştir.

$$DEntropi(A_i) = E(A_i) \times n_i \quad (4.1)$$

Burada kullanılan  $A_i$ ,  $i$ . karakteristiği ve  $n_i$  ise  $i$ . karakteristikteki değer sayısını göstermektedir. Sadece entropi kullanmak yerine, düzeltilmiş entropi kullanılmasının önemi bir örnek yardımıyla EK-A’da anlatılmıştır.

#### 4.2.1.2. Kısa işlem süresi

Eğitim seti düzeltilmiş entropi değerlerine göre yeniden sıralandıktan sonra kural çıkarma özelliği en iyi olan karakteristik değerleri öncelikle değerlendirecektir. Kural oluşturulduktan sonra diğer kombinasyonlar dikkate alınmayacak ve sınıflandırılan örnekler işaretlenecektir. İşaretlenen örnekler için kural çıkarma prosedürü tekrarlanmayacaktır. Özellikle karakteristik sayısının çok fazla olduğu problemlerde gereksiz karşılaştırma işlemleri yapılmayacağından dolayı kural çıkarma işlemleri daha kısa zamanda yapılacaktır.

#### 4.2.1.3. Eksik değerli örnekleri işleyebilme

Gerçek problemlerde bazı örnekler için karakteristik değerleri belirli olmayabilir. REX-1 algoritması bilinmeyen bu değerleri göz önünde bulundurmaz ve bu örnekler için de işlem yapılması sağlanır. Bilgi girişleri sırasında bu değerler boş olarak işlendiğinde bu değerler için kural çıkarılmaz.

#### 4.2.1.4. Sayısal değer içeren karakteristikleri işleyebilme

Sayısal değer içeren karakteristiklerde her değerın ayrı ayrı ele alınması mantıksız sonuçların elde edilmesini sağlar. Bu durum genel kurallar yerine özel kuralların çıkarılmasına sebep olur. IRIS veri setinde olduğu gibi şayet herhangi bir karakteristik sayısal değerler içeriyorsa bu karakteristik için aralık tanımlamalarının yapılması gerekir. Yapılan bu aralık tanımlamalarına göre kural çıkarma prosedürü kural çıkarma işlemini yerine getirir. Şayet aralık tanımlamalarının yapılmaması durumunda elde edilen kural kuralları özel kurallara olacaktır. Aralık tanımlamaları ile daha genel kuralları elde edilir.

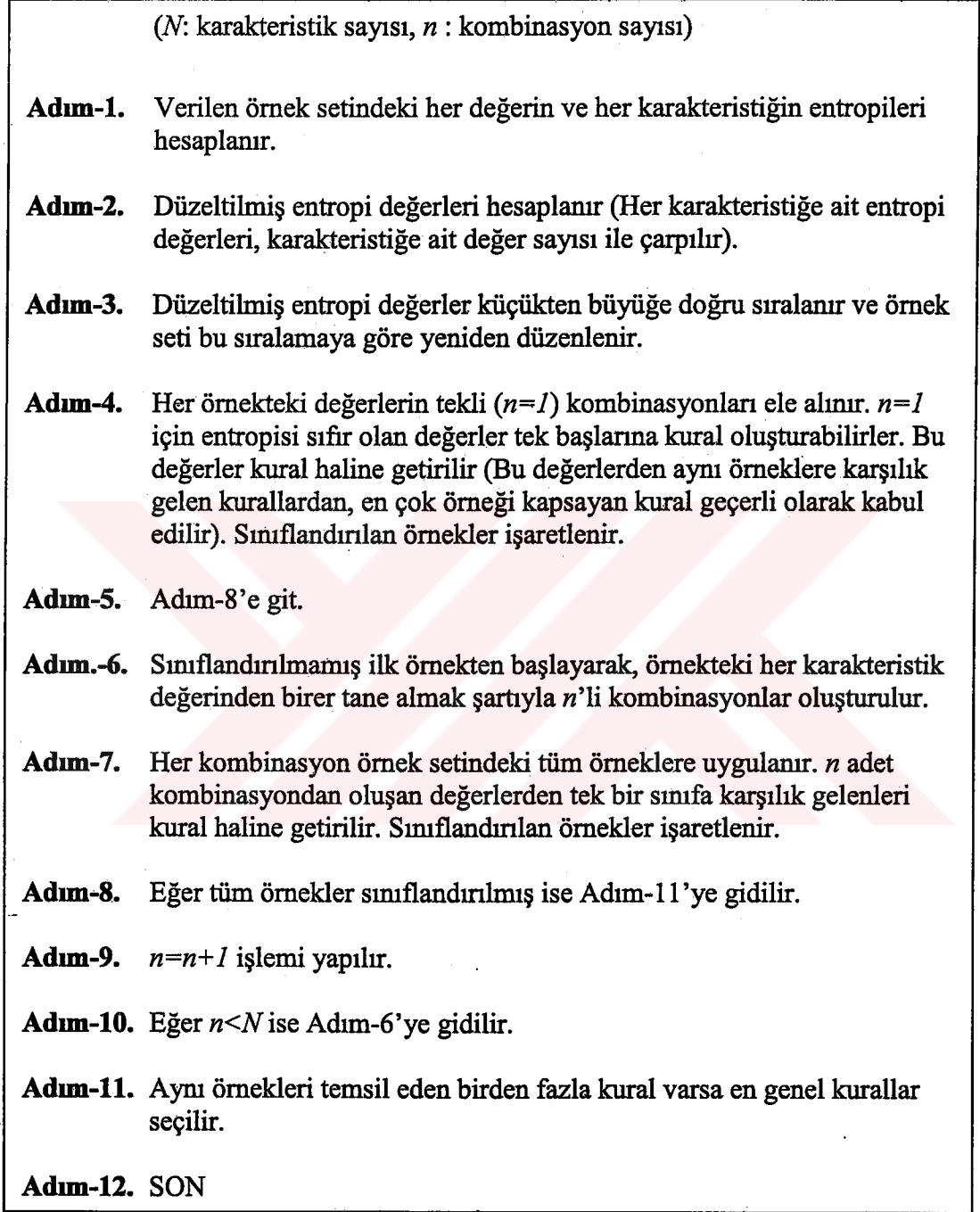
Aralık tanımlamaları her sayısal karakteristik için ayrı ayrı yapılabilir. Her aralığa 1'den başlayarak otomatik olarak sayılar verilir. Bu sayısal değerlere göre örnek seti yeniden düzenlenir, entropi ve düzeltilmiş entropi değerleri yeniden hesaplanır. Bu değerlere göre kuralları üretildikten sonra ilgili sayılar yerine gerçek aralık değerleri yerleştirilir.

#### 4.2.2. Genel ihtiyaçlar

- 1) Bir örnekler seti (öğrenme seti) bir tablo halinde listelenebilir olmalıdır. Her kolon karakteristik değerlerinden oluşur.
- 2) Bir örnek seti,  $m$  adet örnek, her örnek ise  $n$  adet karakteristik değerinden ve  $s$  adet sınıf değerinden oluşur.

#### 4.2.3. Algoritma

Geliştirilen REX-1 algoritması Şekil 4.1’de adım adım gösterilmiştir.



Şekil 4.1. REX-1 algoritması

#### 4.2.4. Algoritma ile uygulama

Geliştirilen bu algoritma, örnek setinden (öğrenme setinden) genel kurallar çıkaran oldukça pratik bir algoritmadır. Bu algoritma, Tablo 4.1’de verilen mevsimler örneği kullanılarak adım adım anlatılmıştır.

Tablo 4.1. Mevsimler eğitim seti

| Örnek No | Hava     | Ağaçlar   | Sıcaklık | Mevsim   |
|----------|----------|-----------|----------|----------|
| 1        | Yağmurlu | Yapraksız | Normal   | Sonbahar |
| 2        | Yağmurlu | Yapraksız | Düşük    | Kış      |
| 3        | Karlı    | Yapraksız | Düşük    | Kış      |
| 4        | Güneşli  | Yapraksız | Düşük    | Kış      |
| 5        | Yağmurlu | Sarı      | Normal   | Sonbahar |
| 6        | Yağmurlu | Yeşil     | Yüksek   | Yaz      |
| 7        | Yağmurlu | Yeşil     | Normal   | İlkbahar |
| 8        | Güneşli  | Yeşil     | Normal   | İlkbahar |
| 9        | Güneşli  | Yeşil     | Yüksek   | Yaz      |
| 10       | Güneşli  | Sarı      | Normal   | Sonbahar |
| 11       | Karlı    | Yeşil     | Düşük    | Kış      |

Tablo 4.1’de verilen örnek seti 11 örnekten ( $m = 11$ ), üç karakteristik (*Hava*, *Ağaçlar*, *Sıcaklık*) ve dört sınıftan (*İlkbahar*, *Yaz*, *Sonbahar*, *Kış*) oluşmaktadır. Örnek setinde yer alan karakteristikler ve bunlara ait değerler aşağıda verilmiştir:

| Karakteristik | Değerler                 |
|---------------|--------------------------|
| Hava          | Yağmurlu, Karlı, Güneşli |
| Ağaçlar       | Yapraksız, Sarı, Yeşil   |
| Sıcaklık      | Normal, Düşük, Yüksek    |

Bu bilgilere göre verilen örnek setini kullanarak algoritmanın adım adım uygulanması aşağıda açıklanmıştır:

##### Adım-1:

İlk olarak her karakteristik ve değer için entropi değerleri hesaplanacaktır. Buna göre, *Hava* karakteristiği *Yağmurlu*, *Karlı* ve *Güneşli* olmak üzere üç değerden oluşmaktadır.

*Hava* karakteristiğinin *Yağmurlu* değeri 5 örnekte geçmektedir. Bu örneklerin ikisi *Sonbahar*, biri *Yaz*, biri *Kış* ve biri de *İlkbahar* sınıflarına aittir. Bu bilgilere göre  $\{Hava, Yağmurlu\}$  için entropi değeri;

$$E_{Hava, Yağmurlu} = -\frac{2}{5} \log_2 \frac{2}{5} - \frac{1}{5} \log_2 \frac{1}{5} - \frac{1}{5} \log_2 \frac{1}{5} - \frac{1}{5} \log_2 \frac{1}{5}$$

$$E_{Hava, Yağmurlu} = 1.922 \text{ bit olur.}$$

*Hava* karakteristiğinin *Karlı* değeri 2 örnekte geçmektedir. Bu örneklerin her ikisi de *Kış* mevsimine aittir. Buna göre  $\{Hava, Karlı\}$  için entropi değeri;

$$E_{Hava, Karlı} = -\frac{2}{2} \log_2 \frac{2}{2}$$

$$E_{Hava, Karlı} = 0 \text{ bit olur.}$$

*Hava* karakteristiğinin *Güneşli* değeri dört örnekte geçmektedir. Her bir örnek farklı sınıflara aittir. Buna göre  $\{Hava, Güneşli\}$  için entropi değeri;

$$E_{Hava, Güneşli} = -\frac{1}{4} \log_2 \frac{1}{4} - \frac{1}{4} \log_2 \frac{1}{4} - \frac{1}{4} \log_2 \frac{1}{4} - \frac{1}{4} \log_2 \frac{1}{4}$$

$$E_{Hava, Güneşli} = 2.0 \text{ bit olarak hesaplanır.}$$

Bu bilgilere göre *Hava* karakteristiğinin entropi değeri aşağıdaki gibi olur:

$$E_{Hava} = \frac{5}{11} x E_{Hava, Yağmurlu} + \frac{2}{11} x E_{Hava, Karlı} + \frac{4}{11} x E_{Hava, Güneşli}$$

$$E_{Hava} = \frac{5}{11} x (1.922) + \frac{2}{11} x (0) + \frac{4}{11} x (2.0)$$

$$E_{Hava} = 1.601 \text{ bit}$$

İkinci karakteristik olan *Ağaçlar*,  $\{Yapraksız, Sarı, Yeşil\}$  değerlerinden oluşmaktadır.

Bu karakteristikteki her değer için ayrı ayrı entropi değerleri hesaplanacaktır.

*Ağaçlar* karakteristiğinin ilk değeri olan *Yapraksız* dört örnekte geçmektedir. Bu örneklerin biri *Sonbahar*, üç tanesi ise *Kış* sınıfına aittir. Bu bilgilere göre  $\{Ağaçlar, Yapraksız\}$  için entropi değeri;

$$E_{Ağaçlar, Yapraksız} = -\frac{1}{4} \log_2 \frac{1}{4} - \frac{3}{4} \log_2 \frac{3}{4}$$

$$E_{Ağaçlar, Yapraksız} = 0.811 \text{ bit olur.}$$

*Ağaçlar* karakteristiğinin ikinci değeri olan *Sarı*, iki örnekte geçmektedir. Bu örneklerin ikisi de *Sonbahar* sınıfına aittir. Bu bilgilere göre  $\{Ağaçlar, Sarı\}$  için entropi değeri;

$$E_{Ağaçlar, Sarı} = -\frac{2}{2} \log_2 \frac{2}{2}$$

$$E_{Ağaçlar, Sarı} = 0 \text{ bit olur.}$$

*Ağaçlar* karakteristiğinin üçüncü değeri olan *Yeşil*, 5 örnekte geçmektedir. Bu örneklerin ikisi *Yaz*, ikisi *İlkbahar* ve biri de *Kış* sınıfına aittir. Bu bilgilere göre  $\{Ağaçlar, Yeşil\}$  için entropi değeri;

$$E_{Ağaçlar, Yeşil} = -\frac{2}{5} \log_2 \frac{2}{5} - \frac{2}{5} \log_2 \frac{2}{5} - \frac{1}{5} \log_2 \frac{1}{5}$$

$$E_{Ağaçlar, Yeşil} = 1.522 \text{ bit olur.}$$

Bu bilgilere göre *Ağaçlar* karakteristiğinin entropisi aşağıdaki gibi hesaplanır:

$$E_{Ağaçlar} = \frac{4}{11} x E_{Ağaçlar, Yapraksız} + \frac{2}{11} x E_{Ağaçlar, Sarı} + \frac{5}{11} x E_{Ağaçlar, Yeşil}$$

$$E_{Ağaçlar} = \frac{4}{11} x (0.811) + \frac{2}{11} x (0) + \frac{5}{11} x (1.522)$$

$$E_{Ağaçlar} = 0.987 \text{ bit}$$

Üçüncü karakteristik olan *Sıcaklık*, {*Normal*, *Düşük*, *Yüksek*} değerlerinden oluşmaktadır. Bu değerlerin entropileri aşağıdaki gibi hesaplanır:

*Sıcaklık* karakteristiğinin *Normal* değeri 5 örnekte geçmektedir. Bunların ikisi *İlkbahar* ve üçü de *Sonbahar* sınıfına aittir. Buna göre {*Sıcaklık*, *Normal*} için entropi değeri;

$$E_{Sıcaklık,Normal} = -\frac{2}{5} \log_2 \frac{2}{5} - \frac{3}{5} \log_2 \frac{3}{5}$$

$$E_{Sıcaklık,Normal} = 0.971 \text{ bit olur.}$$

*Sıcaklık* karakteristiğinin ikinci değeri olan *Düşük*, 4 örnekte geçmektedir. Bu örneklerin hepsi *Kış* sınıfına aittir. Buna göre {*Sıcaklık*, *Düşük*} için entropi değeri;

$$E_{Sıcaklık,Düşük} = -\frac{4}{4} \log_2 \frac{4}{4}$$

$$E_{Sıcaklık,Düşük} = 0 \text{ bit olur.}$$

*Sıcaklık* karakteristiğinin son değeri olan *Yüksek*, 2 örnekte geçmektedir. Bu örneklerin ikisi de *Yaz* sınıfına aittir. Buna göre {*Sıcaklık*, *Yüksek*} için entropi değeri;

$$E_{Sıcaklık,Yüksek} = -\frac{2}{2} \log_2 \frac{2}{2}$$

$$E_{Sıcaklık,Yüksek} = 0 \text{ bit olur.}$$

Bu bilgilere göre *Sıcaklık* karakteristiğinin entropisi aşağıdaki gibi hesaplanır:

$$E_{Sıcaklık} = \frac{5}{11} \times E_{Sıcaklık,Normal} + \frac{4}{11} \times E_{Sıcaklık,Düşük} + \frac{2}{11} \times E_{Sıcaklık,Yüksek}$$

$$E_{Sıcaklık} = \frac{5}{11} \times (0.971) + \frac{4}{11} \times (0) + \frac{2}{11} \times (0)$$

$$E_{Sıcaklık} = 0.441 \text{ bit}$$

Hesaplanan entropi deęerleri Tablo 4.2’de toplu olarak verilmiřtir.

Tablo 4.2. Karakteristikler ve deęerlere ait entropi deęerleri

| Karakteristik | Entropi (bit) | Deęer     | Entropi (bit) |
|---------------|---------------|-----------|---------------|
| Hava          | 1.601         | Yaęmurlu  | 1.922         |
|               |               | Karlı     | 0             |
|               |               | Güneřli   | 2.0           |
| Aęaęlar       | 0.987         | Yapraksız | 0.189         |
|               |               | Sarı      | 0             |
|               |               | Yeřil     | 1.522         |
| Sıcaklık      | 0.441         | Normal    | 0.971         |
|               |               | Düşük     | 0             |
|               |               | Yüksek    | 0             |

#### Adım 2:

Tablo 4.2’deki entropi deęerlerine göre *Düzeltilmiş Entropi (DEntropi)* deęerleri ařaęıdaki gibi hesaplanır. *Hava*, *Aęaęlar* ve *Sıcaklık* karakteristikleri 3’er deęerden oluřmaktadır.

$$DEntropi(Hava) = 1.601 * 3 = 4.803 \text{ bit.}$$

$$DEntropi(Aęaęlar) = 0.987 * 3 = 2.961 \text{ bit.}$$

$$DEntropi(Sıcaklık) = 0.441 * 3 = 1.324 \text{ bit.}$$

#### Adım-3:

Düzeltilmiş entropi deęerlerine göre örnek seti ařaęıdaki karakteristik sırasına göre yeniden düzenlenir.

$$DEntropi(Sıcaklık) < DEntropi(Aęaęlar) < DEntropi(Hava)$$

Bu sıralamaya göre Tablo 4.1’deki örnekler seti yeniden düzenlenir. *Sıcaklık*, *Aęaęlar* ve *Hava* karakteristiklerine göre yeniden düzenlenen örnek seti Tablo 4.3’te verilmiřtir.

Tablo 4.3. Yeniden düzenlenmiş örnek seti (Tablo 4.1)

| Örnek No | Sıcaklık | Ağaçlar   | Hava     | Mevsim   |
|----------|----------|-----------|----------|----------|
| 1        | Normal   | Yapraksız | Yağmurlu | Sonbahar |
| 2        | Düşük    | Yapraksız | Yağmurlu | Kış      |
| 3        | Düşük    | Yapraksız | Karlı    | Kış      |
| 4        | Düşük    | Yapraksız | Güneşli  | Kış      |
| 5        | Normal   | Sarı      | Yağmurlu | Sonbahar |
| 6        | Yüksek   | Yeşil     | Yağmurlu | Yaz      |
| 7        | Normal   | Yeşil     | Yağmurlu | İlkbahar |
| 8        | Normal   | Yeşil     | Güneşli  | İlkbahar |
| 9        | Yüksek   | Yeşil     | Güneşli  | Yaz      |
| 10       | Normal   | Sarı      | Güneşli  | Sonbahar |
| 11       | Düşük    | Yeşil     | Karlı    | Kış      |

**Adım-4:**

Tablo 4.3'teki örnek setine bakıldığında entropisi sıfır olan değerler aşağıdadır.

Sıcaklık karakteristiği için; Düşük ve Yüksek  
Ağaçlar karakteristiği için; Sarı  
Hava karakteristiği için; Karlı

Bu değerler tekli kombinasyonlara göre aşağıdaki kuralları oluştururlar:

**Kural-1: EĞER Sıcaklık Düşük İSE Mevsim KIŞ**

Bu kural ile 2, 3, 4 ve 11 nolu örnekler sınıflandırılır. Bu örneklerin tümü sadece *KIŞ* sınıfına aittir.

**Kural-2: EĞER Sıcaklık Yüksek İSE Mevsim YAZ**

Bu kural ile 6 ve 9 nolu örnekler sınıflandırılır. Bu örnekler sadece *YAZ* sınıfına aittir.

**Kural-3: EĞER Ağaçlar Sarı İSE Mevsim SONBAHAR**

Bu kural ile 5 ve 10 nolu örnekler sınıflandırılır. Bu örnekler sadece *Sonbahar* sınıfına aittir.

**Kural-4: EĞER Hava Karlı İSE Mevsim KİŞ**

Bu kural ile 3 ve 11 nolu örnekler sınıflandırılır. Ancak 3 ve 11 nolu örnekler daha önce Kural-1 ile sınıflandırıldığı için bu örnekler için tekrar kural oluşturulmaz. Bu kural geçersiz olarak kabul edilir.

Bu işlemlerden sonra Adım-8'e gidilir. Tüm örnekler sınıflandırılmadığından dolayı Adım-9'daki  $n=n+1$  işlemi ile  $n$ 'nin değeri bir artırılarak iki olur ve Adım-6'ya gidilir.

**Adım-6:**

Sınıflandırılmayan örnekler olarak 1, 7 ve 8 nolu örnekler bulunmaktadır. Bu örnekler Tablo 4.4'te gösterilmiştir.

Tablo 4.4. Birinci aşama sonunda sınıflandırılmayan örnekler

| Örnek No | Sıcaklık | Ağaçlar   | Hava     | Mevsim   |
|----------|----------|-----------|----------|----------|
| 1        | Normal   | Yapraksız | Yağmurlu | Sonbahar |
| 7        | Normal   | Yeşil     | Yağmurlu | İlkbahar |
| 8        | Normal   | Yeşil     | Güneşli  | İlkbahar |

Birinci örnekteki değerlerin ikili kombinasyonları oluşturulur. Bu değerlerin ikili kombinasyonları;  $\{Normal, Yapraksız\}$ ,  $\{Normal, Yağmurlu\}$ ,  $\{Yapraksız, Yağmurlu\}$  şeklindedir.

**Adım-7:**

İkili kombinasyonların ilki olan  $\{Normal, Yapraksız\}$  değerleri için tüm örnek setine bakılır. Bu değerler ile sadece 1 nolu örnek sınıflandırılabilir.

**Kural-4 : EĞER Sıcaklık Normal VE Ağaçlar Yapraksız İSE Mevsim SONBAHAR**

Sınıflandırılmayan örnekler olarak 7 ve 8 nolu örnekler kalmaktadır. Yedinci örnekteki değerlerin ikili kombinasyonları oluşturulur. Bu değerlerin ikili kombinasyonları  $\{Normal, Yeşil\}$ ,  $\{Normal, Yağmurlu\}$  ve  $\{Normal, Yağmurlu\}$

şeklindedir. {Normal, Yeşil} değerleri 7 ve 8 nolu örneklerde sadece bir sınıfta geçtiklerinden kural haline getirilir.

#### **Kural-5 : EĞER Sıcaklık Normal VE Ağaçlar Yeşil İSE Mevsim İLKBAHAR**

Sınıflandırılacak başka örnek kalmadığından dolayı işlemler tamamlanır. Çıkarılan kurallar Tablo 4.5'te gösterilmiştir.

Tablo 4.5. Tablo 4.1'de verilen örnek seti için üretilen kurallar

| Kural No | Kural Tanımı                                                                       |
|----------|------------------------------------------------------------------------------------|
| 1        | <b>EĞER</b> Sıcaklık=Düşük <b>İSE</b> Mevsim=KIŞ                                   |
| 2        | <b>EĞER</b> Sıcaklık=Yüksek <b>İSE</b> Mevsim=YAZ                                  |
| 3        | <b>EĞER</b> Ağaçlar=Sarı <b>İSE</b> Mevsim=SONBAHAR                                |
| 4        | <b>EĞER</b> Sıcaklık=Normal <b>VE</b> Ağaçlar=Yeşil <b>İSE</b> Mevsim=İLKBAHAR     |
| 5        | <b>EĞER</b> Sıcaklık=Normal <b>VE</b> Ağaçlar=Yapraksız <b>İSE</b> Mevsim=SONBAHAR |

### **4.3. Diğer Algoritmalarla Karşılaştırma**

Geliştirilen algoritma ile diğer algoritmaları karşılaştırmak için Yatırım Projesi[117], Mevsimler örneği, Lens24 veri seti[113] ve IRIS veri seti [112] kullanılacaktır. Ayrıca en son geliştirilen algoritmalar olan GDT-NR, GDT-RS[114] algoritmalarıyla karşılaştırılacaktır.

#### **4.3.1. Yatırım Projesi örneğini kullanarak karşılaştırma**

Tablo 4.6'da verilen Yatırım Projesi eğitim seti ID3, C4.5, İLA, RULES-3 [117] ve REX-1 algoritmaları kullanılarak kurallar elde edilmiştir. Elde edilen bu kurallar Tablo 4.7'de gösterilmiştir. Tüm algoritmalar aynı sayıda (5) kural üretmişlerdir. Ancak ID3 ile elde edilen 3. kuraldaki şart sayısı diğer algoritmalara göre bir fazladır.

Tablo 4.6. Yatırım projesi sınıflandırma eğitim seti [117]

| Örnek No | Yatırım Seviyesi | Geri Dönüş Zamanı | Risk   | Karar |
|----------|------------------|-------------------|--------|-------|
| 1        | Yüksek           | Kısa              | Orta   | IP1   |
| 2        | Yüksek           | Orta              | Düşük  | IP2   |
| 3        | Orta             | Orta              | Düşük  | IP2   |
| 4        | Düşük            | Orta              | Düşük  | IP2   |
| 5        | Yüksek           | Orta              | Orta   | IP1   |
| 6        | Yüksek           | Uzun              | Yüksek | IP0   |
| 7        | Yüksek           | Uzun              | Orta   | IP3   |
| 8        | Düşük            | Uzun              | Orta   | IP3   |
| 9        | Düşük            | Uzun              | Yüksek | IP0   |
| 10       | Orta             | Uzun              | Düşük  | IP2   |
| 11       | Düşük            | Kısa              | Orta   | IP1   |

IP0 :Yatırım Projesinin Rededilmesi

IP1 :Kısa-Orta dönem, Orta Risk Projesi

IP2 :Orta-Uzun dönem, Küçük Risk Projesi

IP3 :Uzun dönem, Orta Risk Projesi

Tablo 4.7. ID3, C4.5, ILA, RULES-3 ve REX-1 ile üretilen kurallar

| Kural No | Algoritma                              | Kural Tanımı                                                                                                                                                                                                                                                                         |
|----------|----------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1        | ID3<br>C4.5<br>ILA<br>RULES-3<br>REX-1 | EĞER Risk=Düşük İSE Karar = IP2<br>EĞER Risk=Düşük İSE Karar = IP2<br>EĞER Risk=Düşük İSE Karar = IP2<br>EĞER Risk=Düşük İSE Karar = IP2<br>EĞER Risk=Düşük İSE Karar = IP2                                                                                                          |
| 2        | ID3<br>C4.5<br>ILA<br>RULES-3<br>REX-1 | EĞER Risk=Yüksek İSE Karar = IP0<br>EĞER Risk=Yüksek İSE Karar = IP0<br>EĞER Risk=Yüksek İSE Karar = IP0<br>EĞER Risk=Yüksek İSE Karar = IP0<br>EĞER Risk=Yüksek İSE Karar = IP0                                                                                                     |
| 3        | ID3<br>C4.5<br>ILA<br>RULES-3<br>REX-1 | EĞER Dönüş_Zaman=Kısa VE Risk = Orta İSE Karar = IP1<br>EĞER Dönüş_Zaman=Kısa İSE Karar = IP1<br>EĞER Dönüş_Zaman=Kısa İSE Karar = IP1<br>EĞER Dönüş_Zaman=Kısa İSE Karar = IP1<br>EĞER Dönüş_Zaman=Kısa İSE Karar = IP1                                                             |
| 4        | ID3<br>C4.5<br>ILA<br>RULES-3<br>REX-1 | EĞER Dönüş_Zaman=Orta VE Risk = Orta İSE Karar = IP1<br>EĞER Dönüş_Zaman=Orta VE Risk = Orta İSE Karar = IP1<br>EĞER Dönüş_Zaman=Orta VE Risk = Orta İSE Karar = IP1<br>EĞER Dönüş_Zaman=Orta VE Risk = Orta İSE Karar = IP1<br>EĞER Dönüş_Zaman=Orta VE Risk = Orta İSE Karar = IP1 |
| 5        | ID3<br>C4.5<br>ILA<br>RULES-3<br>REX-1 | EĞER Dönüş_Zaman=Uzun VE Risk = Orta İSE Karar = IP1<br>EĞER Dönüş_Zaman=Uzun VE Risk = Orta İSE Karar = IP1<br>EĞER Dönüş_Zaman=Uzun VE Risk = Orta İSE Karar = IP1<br>EĞER Dönüş_Zaman=Uzun VE Risk = Orta İSE Karar = IP1<br>EĞER Dönüş_Zaman=Uzun VE Risk = Orta İSE Karar = IP1 |

### 4.3.2. ID3 algoritması ile karşılaştırma

Entropi ölçüsünü kullanan algoritmalarından biri olan ID3 algoritması ile REX-1 algoritması mevsimler örnek seti kullanılarak karşılaştırılacaktır. Tablo 4.8’de, REX-1 algoritması kullanılarak elde edilen kurallar ile ID3 algoritması tarafından elde edilen kurallar gösterilmektedir. ID3 algoritması tarafından üretilen üçüncü kural iki şart içerirken, REX-1 algoritması tarafından üretilen üçüncü kural sadece bir şart içermektedir ve diğer kurallar ise aynıdır. Kural başına ortalama şart sayısı ID3 için 1.6 iken REX-1 için 1.4 olmaktadır.

Tablo 4.8. REX-1 ile ID3 algoritmasının karşılaştırılması

| Kural | Algoritma | Kural Tanımı                                                  |
|-------|-----------|---------------------------------------------------------------|
| 1     | ID3       | EĞER Sıcaklık Düşük İSE Mevsim KIŞ                            |
|       | REX-1     | EĞER Sıcaklık Düşük İSE Mevsim KIŞ                            |
| 2     | ID3       | EĞER Sıcaklık Yüksek İSE Mevsim YAZ                           |
|       | REX-1     | EĞER Sıcaklık Yüksek İSE Mevsim YAZ                           |
| 3     | ID3       | EĞER Sıcaklık Normal VE Ağaçlar Sarı İSE Mevsim SONBAHAR      |
|       | REX-1     | EĞER Ağaçlar Sarı İSE Mevsim SONBAHAR                         |
| 4     | ID3       | EĞER Sıcaklık Normal VE Ağaçlar Yeşil İSE Mevsim İLKBAHAR     |
|       | REX-1     | EĞER Sıcaklık Normal VE Ağaçlar Yeşil İSE Mevsim İLKBAHAR     |
| 5     | ID3       | EĞER Sıcaklık Normal VE Ağaçlar Yapraksız İSE Mevsim SONBAHAR |
|       | REX-1     | EĞER Sıcaklık Normal VE Ağaçlar Yapraksız İSE Mevsim SONBAHAR |

### 4.3.3. IRIS veri setini kullanarak karşılaştırma

IRIS (Tablo 4.9), 1936 yılında Fisher tarafından geliştirilen bir veri setidir [112]. Bu standart veri seti, genellikle örnek sınıflandırma algoritmalarının karşılaştırılmasında sıklıkla kullanılır. IRIS veri seti 150 adet örnek ve dört sayısal karakteristik değerinden oluşmaktadır: Sepal-Length (SL), Sepal-Width (SW), Petal-Length(PL) ve Petal-Width(PW) (Sepal : Taç yaprak, Petal : Çanak yaprak). Veri seti üç sınıftan oluşmaktadır: Iris-versicolor, Iris-Setosa ve Iris-virginica. Daha önce yapılmış olan uygulamalar göz önüne alınarak eğitim seti olarak ilk 70 ve test seti olarak ise son 80 adet örnek seçilmiştir.

Tablo 4.9. IRIS veri seti

| No | Sepal Length | Sepal Width | Petal Length | Petal Width | IRIS            | No  | Sepal Length | Sepal Width | Petal Length | Petal Width | IRIS            |
|----|--------------|-------------|--------------|-------------|-----------------|-----|--------------|-------------|--------------|-------------|-----------------|
| 1  | 6.3          | 3.3         | 4.7          | 1.6         | Iris-versicolor | 51  | 5.0          | 3.0         | 1.6          | 0.2         | Iris-setosa     |
| 2  | 5.8          | 4.0         | 1.2          | 0.2         | Iris-setosa     | 52  | 6.7          | 3.3         | 5.7          | 2.5         | Iris-virginica  |
| 3  | 6.7          | 3.3         | 5.7          | 2.1         | Iris-virginica  | 53  | 5.8          | 2.8         | 5.1          | 2.4         | Iris-virginica  |
| 4  | 6.1          | 2.8         | 4.0          | 1.3         | Iris-versicolor | 54  | 5.8          | 2.6         | 4.0          | 1.2         | Iris-versicolor |
| 5  | 5.0          | 3.4         | 1.5          | 0.2         | Iris-setosa     | 55  | 6.0          | 3.4         | 4.5          | 1.6         | Iris-versicolor |
| 6  | 6.2          | 2.2         | 4.5          | 1.5         | Iris-versicolor | 56  | 5.0          | 3.6         | 1.4          | 0.2         | Iris-setosa     |
| 7  | 7.2          | 3.6         | 6.1          | 2.5         | Iris-virginica  | 57  | 5.0          | 3.4         | 1.6          | 0.4         | Iris-setosa     |
| 8  | 4.4          | 2.9         | 1.4          | 0.2         | Iris-setosa     | 58  | 6.9          | 3.1         | 5.4          | 2.1         | Iris-virginica  |
| 9  | 5.0          | 3.5         | 1.3          | 0.3         | Iris-setosa     | 59  | 6.8          | 3.2         | 5.9          | 2.3         | Iris-virginica  |
| 10 | 5.4          | 3.4         | 1.5          | 0.4         | Iris-setosa     | 60  | 6.9          | 3.1         | 5.1          | 2.3         | Iris-virginica  |
| 11 | 6.3          | 2.8         | 5.1          | 1.5         | Iris-virginica  | 61  | 5.2          | 3.4         | 1.4          | 0.2         | Iris-setosa     |
| 12 | 5.5          | 4.2         | 1.4          | 0.2         | Iris-setosa     | 62  | 5.5          | 2.3         | 4.0          | 1.3         | Iris-versicolor |
| 13 | 5.5          | 2.6         | 4.4          | 1.2         | Iris-versicolor | 63  | 6.4          | 2.9         | 4.3          | 1.3         | Iris-versicolor |
| 14 | 5.5          | 2.4         | 3.7          | 1.0         | Iris-versicolor | 64  | 6.7          | 2.5         | 5.8          | 1.8         | Iris-virginica  |
| 15 | 7.7          | 3.0         | 6.1          | 2.3         | Iris-virginica  | 65  | 5.6          | 2.8         | 4.9          | 2.0         | Iris-virginica  |
| 16 | 4.6          | 3.1         | 1.5          | 0.2         | Iris-setosa     | 66  | 5.0          | 2.0         | 3.5          | 1.0         | Iris-versicolor |
| 17 | 6.8          | 2.8         | 4.8          | 1.4         | Iris-versicolor | 67  | 5.1          | 3.3         | 1.7          | 0.5         | Iris-setosa     |
| 18 | 5.8          | 2.7         | 4.1          | 1.0         | Iris-versicolor | 68  | 6.7          | 3.0         | 5.2          | 2.3         | Iris-virginica  |
| 19 | 4.7          | 3.2         | 1.6          | 0.2         | Iris-setosa     | 69  | 4.6          | 3.2         | 1.4          | 0.2         | Iris-setosa     |
| 20 | 5.7          | 3.8         | 1.7          | 0.3         | Iris-setosa     | 70  | 5.3          | 3.7         | 1.5          | 0.2         | Iris-setosa     |
| 21 | 6.5          | 3.2         | 5.1          | 2.0         | Iris-virginica  | 71  | 6.1          | 2.6         | 5.6          | 1.4         | Iris-virginica  |
| 22 | 5.1          | 3.5         | 1.4          | 0.3         | Iris-setosa     | 72  | 5.5          | 2.5         | 4.0          | 1.3         | Iris-Versicolor |
| 23 | 6.3          | 2.7         | 4.9          | 1.8         | Iris-virginica  | 73  | 6.1          | 3.0         | 4.6          | 1.4         | Iris-versicolor |
| 24 | 7.1          | 3.0         | 5.9          | 2.1         | Iris-virginica  | 74  | 4.8          | 3.0         | 1.4          | 0.3         | Iris-setosa     |
| 25 | 6.3          | 2.5         | 4.9          | 1.5         | Iris-versicolor | 75  | 6.0          | 2.2         | 4.0          | 1.0         | Iris-versicolor |
| 26 | 6.1          | 3.0         | 4.9          | 1.8         | Iris-virginica  | 76  | 4.9          | 2.5         | 4.5          | 1.7         | Iris-virginica  |
| 27 | 5.4          | 3.7         | 1.5          | 0.2         | Iris-setosa     | 77  | 7.9          | 3.8         | 6.4          | 2.0         | Iris-virginica. |
| 28 | 5.4          | 3.9         | 1.3          | 0.4         | Iris-setosa     | 78  | 5.5          | 2.4         | 3.8          | 1.1         | Iris-versicolor |
| 29 | 6.4          | 2.7         | 5.3          | 1.9         | Iris-virginica  | 79  | 5.4          | 3.9         | 1.7          | 0.4         | Iris-setosa     |
| 30 | 6.9          | 3.2         | 5.7          | 2.3         | Iris-virginica  | 80  | 5.9          | 3.0         | 4.2          | 1.5         | Iris-versicolor |
| 31 | 7.7          | 3.8         | 6.7          | 2.2         | Iris-virginica  | 81  | 4.8          | 3.1         | 1.6          | 0.2         | Iris-setosa     |
| 32 | 5.1          | 3.8         | 1.9          | 0.4         | Iris-setosa     | 82  | 5.2          | 2.7         | 3.9          | 1.4         | Iris-versicolor |
| 33 | 7.2          | 3.2         | 6.0          | 1.8         | Iris-virginica  | 83  | 4.4          | 3.2         | 1.3          | 0.2         | Iris-setosa     |
| 34 | 6.3          | 2.3         | 4.4          | 1.3         | Iris-versicolor | 84  | 6.2          | 3.4         | 5.4          | 2.3         | Iris-virginica  |
| 35 | 6.6          | 3.0         | 4.4          | 1.4         | Iris-versicolor | 85  | 5.5          | 3.5         | 1.3          | 0.2         | Iris-setosa     |
| 36 | 6.3          | 3.3         | 6.0          | 2.5         | Iris-virginica  | 86  | 5.1          | 3.7         | 1.5          | 0.4         | Iris-setosa     |
| 37 | 6.7          | 3.1         | 4.4          | 1.4         | Iris-versicolor | 87  | 4.9          | 3.1         | 1.5          | 0.1         | Iris-setosa.    |
| 38 | 7.7          | 2.8         | 6.7          | 2.0         | Iris-virginica  | 88  | 4.7          | 3.2         | 1.3          | 0.2         | Iris-setosa     |
| 39 | 5.7          | 3.0         | 4.2          | 1.2         | Iris-versicolor | 89  | 5.0          | 3.2         | 1.2          | 0.2         | Iris-setosa.    |
| 40 | 6.9          | 3.1         | 4.9          | 1.5         | Iris-versicolor | 90  | 5.6          | 2.9         | 3.6          | 1.3         | Iris-versicolor |
| 41 | 6.5          | 3.0         | 5.5          | 1.8         | Iris-virginica  | 91  | 7.3          | 2.9         | 6.3          | 1.8         | Iris-virginica  |
| 42 | 7.2          | 3.0         | 5.8          | 1.6         | Iris-virginica  | 92  | 5.7          | 4.4         | 1.5          | 0.4         | Iris-setosa     |
| 43 | 6.0          | 2.7         | 5.1          | 1.6         | Iris-versicolor | 93  | 4.5          | 2.3         | 1.3          | 0.3         | Iris-setosa     |
| 44 | 4.8          | 3.4         | 1.9          | 0.2         | Iris-setosa     | 94  | 6.5          | 2.8         | 4.6          | 1.5         | Iris-versicolor |
| 45 | 6.4          | 3.2         | 5.3          | 2.3         | Iris-virginica  | 95  | 5.7          | 2.8         | 4.1          | 1.3         | Iris-versicolor |
| 46 | 6.0          | 2.9         | 4.5          | 1.5         | Iris-versicolor | 96  | 4.9          | 3.1         | 1.5          | 0.1         | Iris-setosa     |
| 47 | 4.9          | 2.4         | 3.3          | 1.0         | Iris-versicolor | 97  | 6.1          | 2.9         | 4.7          | 1.4         | Iris-versicolor |
| 48 | 6.6          | 2.9         | 4.6          | 1.3         | Iris-versicolor | 98  | 6.3          | 3.4         | 5.6          | 2.4         | Iris-virginica  |
| 49 | 7.0          | 3.2         | 4.7          | 1.4         | Iris-versicolor | 99  | 4.6          | 3.4         | 1.4          | 0.3         | Iris-setosa     |
| 50 | 6.3          | 2.5         | 5.0          | 1.9         | Iris-virginica  | 100 | 4.9          | 3.1         | 1.5          | 0.1         | Iris-setosa     |

Tablo 4.9.(Devam) IRIS veri seti

| No  | Sepal<br>Length | Sepal<br>Width | Petal<br>Length | Petal<br>Width | IRIS            |
|-----|-----------------|----------------|-----------------|----------------|-----------------|
| 101 | 5.7             | 2.8            | 4.5             | 1.3            | Iris-versicolor |
| 102 | 6.5             | 3.0            | 5.2             | 2.0            | Iris-virginica  |
| 103 | 4.6             | 3.6            | 1.0             | 0.2            | Iris-setosa     |
| 104 | 5.7             | 2.9            | 4.2             | 1.3            | Iris-versicolor |
| 105 | 5.8             | 2.7            | 5.1             | 1.9            | Iris-virginica  |
| 106 | 5.0             | 3.5            | 1.6             | 0.6            | Iris-setosa     |
| 107 | 6.2             | 2.9            | 4.3             | 1.3            | Iris-versicolor |
| 108 | 5.6             | 2.5            | 3.9             | 1.1            | Iris-versicolor |
| 109 | 6.7             | 3.1            | 4.7             | 1.5            | Iris-versicolor |
| 110 | 5.1             | 3.8            | 1.6             | 0.2            | Iris-setosa     |
| 111 | 7.7             | 2.6            | 6.9             | 2.3            | Iris-virginica  |
| 112 | 4.9             | 3.0            | 1.4             | 0.2            | Iris-setosa     |
| 113 | 5.7             | 2.5            | 5.0             | 2.0            | Iris-virginica  |
| 114 | 5.2             | 3.5            | 1.5             | 0.2            | Iris-setosa     |
| 115 | 6.3             | 2.9            | 5.6             | 1.8            | Iris-virginica  |
| 116 | 5.2             | 4.1            | 1.5             | 0.1            | Iris-setosa     |
| 117 | 4.8             | 3.0            | 1.4             | 0.1            | Iris-setosa     |
| 118 | 7.4             | 2.8            | 6.1             | 1.9            | Iris-virginica  |
| 119 | 5.4             | 3.0            | 4.5             | 1.5            | Iris-versicolor |
| 120 | 4.3             | 3.0            | 1.1             | 0.1            | Iris-setosa     |
| 121 | 5.8             | 2.7            | 5.1             | 1.9            | Iris-virginica  |
| 122 | 5.1             | 3.8            | 1.5             | 0.3            | Iris-setosa     |
| 123 | 4.4             | 3.0            | 1.3             | 0.2            | Iris-setosa     |
| 124 | 6.7             | 3.0            | 5.0             | 1.7            | Iris-versicolor |
| 125 | 5.7             | 2.6            | 3.5             | 1.0            | Iris-versicolor |
| 126 | 6.7             | 3.1            | 5.6             | 2.4            | Iris-virginica  |
| 127 | 5.4             | 3.4            | 1.7             | 0.2            | Iris-setosa     |
| 128 | 6.8             | 3.0            | 5.5             | 2.1            | Iris-virginica  |
| 129 | 5.0             | 3.3            | 1.4             | 0.2            | Iris-setosa     |
| 130 | 6.4             | 2.8            | 5.6             | 2.2            | Iris-virginica  |
| 131 | 6.4             | 3.2            | 4.5             | 1.5            | Iris-versicolor |
| 132 | 4.8             | 3.4            | 1.6             | 0.2            | Iris-setosa.    |
| 133 | 7.6             | 3.0            | 6.6             | 2.1            | Iris-virginica. |
| 134 | 5.9             | 3.0            | 5.1             | 1.8            | Iris-virginica  |
| 135 | 5.1             | 2.5            | 3.0             | 1.1            | Iris-versicolor |
| 136 | 5.6             | 3.0            | 4.5             | 1.5            | Iris-versicolor |
| 137 | 5.1             | 3.4            | 1.5             | 0.2            | Iris-setosa     |
| 138 | 5.6             | 3.0            | 4.1             | 1.3            | Iris-versicolor |
| 139 | 6.2             | 2.8            | 4.8             | 1.8            | Iris-virginica  |
| 140 | 5.1             | 3.5            | 1.4             | 0.2            | Iris-setosa     |
| 141 | 5.0             | 2.3            | 3.3             | 1.0            | Iris-versicolor |
| 142 | 5.8             | 2.7            | 3.9             | 1.2            | Iris-versicolor |
| 143 | 5.6             | 2.7            | 4.2             | 1.3            | Iris-versicolor |
| 144 | 6.5             | 3.0            | 5.8             | 2.2            | Iris-virginica. |
| 145 | 6.0             | 2.2            | 5.0             | 1.5            | Iris-virginica. |
| 146 | 6.4             | 2.8            | 5.6             | 2.1            | Iris-virginica. |
| 147 | 6.0             | 3.0            | 4.8             | 1.8            | Iris-virginica. |
| 148 | 6.4             | 3.1            | 5.5             | 1.8            | Iris-virginica. |
| 149 | 6.1             | 2.8            | 4.7             | 1.2            | Iris-versicolor |
| 150 | 5.9             | 3.2            | 4.8             | 1.8            | Iris-versicolor |

Bu veri setinin seçilmesindeki amaç; verilerin gerçek olması, gürültülü veri içermesi, sayısal karakteristik bulundurması, örnek tanıma literatüründe çok yaygın olarak kullanılması ve endüktif öğrenme algoritmalarında performans karşılaştırmak amacıyla sıklıkla kullanılmasıdır. Rules-3, ID3, Rules-3 Plus ve REX-1 algoritmaları kullanılarak elde edilen kural setleri sırasıyla Tablo 4.10a, 4.10b, 4.10c, 4.10d’de verilmiştir.

Tablo 4.10a. RULES-3 tarafından elde edilen kural seti (Iris veri seti) [96]

| Kural | Kural Tanımı                                                                 |
|-------|------------------------------------------------------------------------------|
| 1     | EĞER $6.56 \leq SL < 7.13$ VE $3.95 \leq PL < 4.93$ İSE IRIS=Iris-versicolor |
| 2     | EĞER $5.91 \leq PL < 6.9$ İSE IRIS =Iris-virginica                           |
| 3     | EĞER $0.9 \leq PW < 1.3$ İSE IRIS =Iris-versicolor                           |
| 4     | EĞER $4.93 \leq PL < 5.91$ İSE IRIS =Iris-virginica                          |
| 5     | EĞER $6 \leq SL < 6.56$ VE $3.95 \leq PL < 4.93$ İSE IRIS =Iris-versicolor   |
| 6     | EĞER $4.86 \leq SL < 5.43$ VE $3.95 \leq PL < 4.93$ İSE IRIS =Iris-virginica |
| 7     | EĞER $1 \leq PL < 1.98$ İSE IRIS =Iris-setosa                                |
| 8     | EĞER $2.96 \leq PL < 3.95$ İSE IRIS =Iris-versicolor                         |
| 9     | EĞER $5.43 \leq SL < 6$ VE $1.3 \leq PW < 1.7$ İSE IRIS =Iris-versicolor     |
| 10    | EĞER $5.43 \leq SL < 6$ VE $3.2 \leq SW < 3.6$ İSE IRIS =Iris-versicolor     |
| 11    | EĞER $2.8 \leq SW < 3.2$ VE $1.7 \leq PW < 2.1$ İSE IRIS =Iris-virginica     |

Tablo 4.10b. ID3 tarafından elde edilen kural seti (Iris veri seti) [96]

| Kural | Kural Tanımı                                                                                   |
|-------|------------------------------------------------------------------------------------------------|
| 1     | EĞER $1 \leq PL < 1.98$ İSE IRIS =Iris-setosa                                                  |
| 2     | EĞER $4.93 \leq PL < 5.91$ İSE IRIS =Iris-virginica                                            |
| 3     | EĞER $5.91 \leq PL < 6.9$ İSE IRIS =Iris-virginica                                             |
| 4     | EĞER $3.95 \leq PL < 4.93$ VE $1.3 \leq PW < 1.7$ İSE IRIS =Iris-versicolor                    |
| 5     | EĞER $3.95 \leq PL < 4.93$ VE $0.9 \leq PW < 1.3$ İSE IRIS =Iris-versicolor                    |
| 6     | EĞER $3.95 \leq PL < 4.93$ VE $1.7 \leq PW < 2.1$ VE $2.4 \leq SW < 2.8$ İSE IRIS =Iris-virg.  |
| 7     | EĞER $3.95 \leq PL < 4.93$ VE $1.7 \leq PW < 2.1$ VE $3.2 \leq SW < 3.6$ İSE IRIS =Iris-versi. |
| 8     | EĞER $2.96 \leq PL < 3.95$ İSE IRIS=Iris-versicolor                                            |

Tablo 4.10c. Rules-3 Plus tarafından elde edilen kural seti (Iris veri seti) [96]

| Kural | Kural Tanımı                                                                |
|-------|-----------------------------------------------------------------------------|
| 1     | EĞER $1 \leq PL < 1.98$ İSE IRIS=Iris-setosa                                |
| 2     | EĞER $3.95 \leq PL < 4.93$ VE $1.3 \leq PW < 1.7$ İSE IRIS =Iris-versicolor |
| 3     | EĞER $5.91 \leq PL < 6.9$ İSE IRIS =Iris-virginica                          |
| 4     | EĞER $4.93 \leq PL < 5.91$ İSE IRIS =Iris-virginica                         |
| 5     | EĞER $2.4 \leq SW < 2.8$ VE $1.7 \leq PW < 2.1$ İSE IRIS =Iris-virginica    |
| 6     | EĞER $2.96 \leq PL < 3.95$ İSE IRIS =Iris-versicolor                        |
| 7     | EĞER $0.9 \leq PW < 1.3$ İSE IRIS =Iris-versicolor                          |
| 8     | EĞER $2.1 \leq PW < 1.5$ İSE IRIS =Iris-virginica                           |
| 9     | EĞER $3.2 \leq SW < 3.6$ VE $3.95 \leq PL < 4.93$ İSE IRIS =Iris-versicolor |
| 10    | EĞER $2.8 \leq SW < 3.2$ VE $1.7 \leq PW < 2.1$ İSE IRIS =Iris-virginica    |

Tablo 4.10d. REX-1 tarafından elde edilen kural seti (Iris veri seti)

| Kural | Kural Tanımı                                                                |
|-------|-----------------------------------------------------------------------------|
| 1     | EĞER $1.3 \leq PW < 1.7$ VE $3.95 \leq PL < 4.93$ İSE IRIS =Iris-versicolor |
| 2     | EĞER $0 \leq PW < 0.5$ İSE IRIS =Iris-setosa                                |
| 3     | EĞER $1.7 \leq PW < 2.1$ İSE IRIS =Iris-virginica                           |
| 4     | EĞER $0.9 \leq PW < 1.3$ İSE IRIS =Iris-versicolor                          |
| 5     | EĞER $2.1 \leq PW < 2.5$ İSE IRIS =Iris-virginica                           |
| 6     | EĞER $4.93 \leq PL < 5.91$ VE $2.8 \leq SW < 3.2$ İSE IRIS =Iris-virginica  |
| 7     | EĞER $2.1 \leq PW < 2.5$ İSE IRIS =Iris-virginica                           |
| 8     | EĞER $1.3 \leq PW < 1.7$ VE $2.4 \leq SW < 2.8$ İSE IRIS =Iris-versicolor   |

Rules-3, ID3, Rules-3 Plus ve REX-1 algoritmaları tarafından üretilen kural sayıları ve kural başına ortalama şart sayıları Tablo 4.11’de verilmiştir. Kural sayısı ve şartların ortalamasına göre REX-1 algoritmasının sonuçları diğer üç algoritmaya göre daha iyidir. Kural sayısı bakımından ID3 ve REX-1 algoritmaları 8 adet kural üretmişlerdir. Ancak ID3 algoritması tarafından üretilen kurallardaki toplam şart sayısı 14 iken, bu sayı REX-1 algoritmasında 11’dir.

Tablo 4.11. Kural sayısı ve kural başına şartların ortalaması (Iris veri seti)

| Algoritma    | Kural Sayısı | Şart Sayısı | Kural Başına Şart Ortalaması |
|--------------|--------------|-------------|------------------------------|
| RULES-3      | 11           | 17          | 1.545                        |
| ID3          | 8            | 14          | 1.75                         |
| RULES-3 PLUS | 10           | 14          | 1.4                          |
| REX-1        | 8            | 11          | 1.375                        |

#### 4.3.4. Lens24 veri setini kullanarak karşılaştırma

Lens24 veri seti, bir hastaya hangi lensin takılıp takılmayacağına karar vermek için kullanılan bir karar verme veritabanıdır [16, 113]. Lens24 veri seti Tablo 4.12’de verilmiştir. A, B, C ve D sembolleri karakteristikleri, E ise verilecek kararı göstermektedir. Gözlükçünün kararının ne olacağını gösteren E aşağıdaki değerleri almaktadır:

- 1 : Hastaya sert lensin uygun olduğunu gösterir.
- 2 : Hastaya yumuşak lensin uygun olduğunu gösterir.
- 3 : Hastaya lens takmanın gereksiz olduğunu gösterir.

Aşağıda verilen karakteristik değerleri ise hasta ile ilgili bazı bilgileri içermektedir:

A : Yaş ( 1 : genç, 2 : uzağı göremeyen, 2 : uzağı görebilen)

B : Görüş ( 1 : miyop, 2: hipermetrop)

C : Astigmat (1 : yok, 2 : var)

D : Gözyaşı oluşma oranı (1 : az, 2 : normal)

Tablo 4.12. Lens24 veri seti

| Örnek No | A | B | C | D | E | Örnek No | A | B | C | D | E |
|----------|---|---|---|---|---|----------|---|---|---|---|---|
| 1        | 1 | 1 | 1 | 1 | 3 | 13       | 2 | 2 | 1 | 1 | 3 |
| 2        | 1 | 1 | 1 | 2 | 2 | 14       | 2 | 2 | 1 | 2 | 2 |
| 3        | 1 | 1 | 2 | 1 | 3 | 15       | 2 | 2 | 2 | 1 | 3 |
| 4        | 1 | 1 | 2 | 2 | 1 | 16       | 2 | 2 | 2 | 2 | 3 |
| 5        | 1 | 2 | 1 | 1 | 3 | 17       | 3 | 1 | 1 | 1 | 3 |
| 6        | 1 | 2 | 1 | 2 | 2 | 18       | 3 | 1 | 1 | 2 | 3 |
| 7        | 1 | 2 | 2 | 1 | 3 | 19       | 3 | 1 | 2 | 1 | 3 |
| 8        | 1 | 2 | 2 | 2 | 1 | 20       | 3 | 1 | 2 | 2 | 1 |
| 9        | 2 | 1 | 1 | 1 | 3 | 21       | 3 | 2 | 1 | 1 | 3 |
| 10       | 2 | 1 | 1 | 2 | 2 | 22       | 3 | 2 | 1 | 2 | 2 |
| 11       | 2 | 1 | 2 | 1 | 3 | 23       | 3 | 2 | 2 | 1 | 3 |
| 12       | 2 | 1 | 2 | 2 | 1 | 24       | 3 | 2 | 2 | 2 | 3 |

Tablo 4.13'te, GDT-NR, GDT-RS ve REX-1 algoritmaları tarafından üretilen kurallar gösterilmektedir. Bu algoritmaların üçü de aynı kuralları üretmişlerdir. Tablo 4.14'de ise C4.5Prune tarafından üretilen kurallar gösterilmiştir[114].

Tablo 4.13. GDT-NR, GDT-RS ve REX-1 tarafından üretilen kurallar

| Kural No | Kural Tanımı                   | Örnek Sayısı | Doğruluk Oranı (%) |
|----------|--------------------------------|--------------|--------------------|
| 1        | EĞER D=1 İSE KARAR=3           | 12           | 100                |
| 2        | EĞER D=2 VE C=1 VE A=1 İSE E=2 | 2            | 100                |
| 3        | EĞER D=2 VE C=2 VE B=1 İSE E=1 | 3            | 100                |
| 4        | EĞER D=2 VE C=2 VE A=1 İSE E=1 | 2            | 100                |
| 5        | EĞER D=2 VE C=1 VE A=2 İSE E=2 | 2            | 100                |
| 6        | EĞER C=2 VE B=2 VE A=2 İSE E=3 | 2            | 100                |
| 7        | EĞER C=1 VE B=1 VE A=3 İSE E=3 | 2            | 100                |
| 8        | EĞER D=2 VE C=1 VE B=2 İSE E=2 | 3            | 100                |
| 9        | EĞER C=2 VE B=2 VE A=3 İSE E=3 | 2            | 100                |

GDT-NR ve GDT-RS algoritmaları kaba-set yöntemini kullanarak kural üreten algoritmalar. GDT-RS küçük veri setleri ve GDT-NR ise büyük veri setleri için tasarlanmıştır[114]. Bu algoritmalar örnek setini bir matris tablosu şeklinde alırlar.

Örnek setinin büyüklüğü kullanılacak bilgisayarın hafızasıyla sınırlı olacaktır. Çok büyük örnek setlerinin çözümü mümkün olamamaktadır.

Günümüzde bilgisayarla öğrenme algoritmalarının en iyisi olarak kabul edilen C4.5Prune ile elde edilen kurallar ve doğruluk oranları Tablo 4.14'te verilmiştir [114].

Tablo 4.14. C4.5prune tarafından elde edilen kurallar

| Kural No | Kural                          | Örnek Sayısı | Doğruluk Oranı (%) |
|----------|--------------------------------|--------------|--------------------|
| 1        | EĞER D=1 İSE KARAR=3           | 12           | 100                |
| 2        | EĞER C=1 VE D=2 İSE E=2        | 5            | 83.3               |
| 3        | EĞER B=1 VE C=2 VE D=1 İSE E=1 | 3            | 100                |

Tablo 4.13'te GDT-NR, GDT-RS ve REX-1 algoritmaları tarafından üretilen toplam kural sayısı dokuz ve tüm kurallar için doğruluk oranları ise %100'dür. Yani, üretilen kurallar tüm örnek seti için %100 doğruluğa sahiptir (REX-1 algoritması, her zaman örnek setlerinde %100 doğrulukla kural üretir).

Tablo 4.14'de C4.5Prune ile elde edilen kurallar görülmektedir. Bu kurallarla sınıflandırılabilen toplam örnek sayısı ise 20'dir. Elde edilen ikinci kural ile sınıflandırılan örnek sayısı beş'tir. Ancak (C=1 ve D=2) şartına uyan örnek sayısı altı'dır. Bunlar; 2, 6, 10, 14, 18 ve 22 nolu örneklerdir. Ancak 18 nolu örnek 3 (E=3) sınıfına karşılık gelirken diğer örnekler 2 (E=2) sınıfına karşılık gelmektedir. Dört örnek ise sınıflandırılamamaktadır. Tüm örnek seti için doğruluk oranı (20/24) %83.33 olmaktadır.

#### 4.4. Sonuçlar

REX-1 algoritması ile elde edilen kurallar, entropi ve düzeltilmiş entropi ölçüleri kullanılarak elde edilmektedir. Bundan dolayı düzeltilmiş entropiye göre değeri düşük olan karakteristiklere öncelik verilecektir. Örnek setindeki karakteristikler *düzeltilmiş entropi* değerlerine göre küçükten büyüğe doğru sıralandıklarından dolayı entropisi yüksek olan karakteristikler en son dikkate alınır.

Mevsimler örneğinde olduğu gibi, entropisi en yüksek olan *Hava* karakteristiği üretilen kurallarda hiç yer almamaktadır. Çünkü *Hava* karakteristiğinin *Güneşli* ve *Yağmurlu* değerleri her dört sınıfta (mevsim) da geçmektedir. Bu da düzensizlik derecesinin yüksek olduğu anlamına gelmektedir.

REX-1 algoritmasının bu özelliği, üretilen kurallarda önem derecesi yüksek olan karakteristiklerin yer almasını, daha az sayıda ve daha genel kurallar elde edilmesini sağlamaktadır. Yapılan karşılaştırma işlemlerinde REX-1 algoritması ID3, RULES-3 ve RULES-3 PLUS algoritmalarından daha iyi sonuçlar üretmiştir.

Endüktif öğrenme algoritmaları ile ilgili gerçek hayattan alınan veri setleri kullanılarak yapılan performans karşılaştırmaları Bölüm-7’de verilmiştir.



## BÖLÜM 5. REX-2 : ENDÜKTİF ÖĞRENME ALGORİTMASI

### 5.1. Giriş

Bu bölümde, Bölüm-4'te tarif edilen REX-1 algoritmasının yeni bir versiyonu anlatılacaktır. Bu algoritma REX-1 algoritmasının tüm özelliklerini taşımaktadır (Sayısal değer içeren karakteristikleri ve eksik değer içeren örnekleri işleyebilme, hızlı işlem süresi). REX-1 algoritmasına göre iki farklı özelliği bulunmaktadır:

1. REX-1 algoritması *entropi* ve *düzeltilmiş entropi* değerlerini kullanarak, önce *düzeltilmiş entropi* değerlerini küçükten büyüğe sıralayıp örnek setini bu sıralamaya göre yeniden düzenleyerek kural çıkarma işlemini gerçekleştirmekteydi. Ancak REX-2 algoritması, örnek setini yeniden sıralama işlemi için *entropi*'ye ek olarak her karakteristiğin *bilgi kazancını* ve *kısmi bilgi* değerlerine göre *bilgi kazanç oranlarını* hesaplar. Hesaplanan bu *bilgi kazanç oranlarını* büyükten küçüğe doğru sıralayarak, örnek setini bu sıralamaya göre yeniden düzenler. Düzenlenen bu örnek setini kullanarak kural çıkarma işlemini gerçekleştirir. Bilgi kazanç oranı, *düzeltilmiş entropiye* göre bilgi değerinin tespit edilmesi bakımından daha verimli sonuç üretmektedir. Yapılan uygulamalar bunun doğruluğunu ortaya koymuştur.
2. REX-1 algoritması düzenlenen örnek setindeki örnekleri ele alırken her örnek için ilk önce tekli, sonra ikili,... kombinasyonları dikkate alarak ilk kural çıkarılıncaya kadar işlemi sürdürür. Ancak REX-2 algoritması ilk önce tüm örneklerde tekli kuralları çıkarmaya başlar. Ardından kalan örneklerde ikili, geriye kalan örneklerde üçlü,... kombinasyonları kullanarak kural çıkarma işlemini gerçekleştirir. Bu özellik yardımıyla daha az şart içeren kuralların çıkarılması sağlanmaktadır.

## 5.2. Algoritmanın Tanımı

Bu algoritmada örnek setini yeniden düzenlemek için her karakteristiğin *Bilgi Kazancı* hesaplanır. Hesaplanan bilgi kazancı karakteristiklerin *Kısmi Entropisine* bölünerek *Bilgi Kazanç Oranları* bulunur. Bu oranlar büyükten küçüğe doğru sıralanarak, örnek seti bu sıralamaya göre yeniden düzenlenir. Bu şekilde bilgi kazanç oranı en yüksek olan karakteristiklere öncelik verilmesi sağlanır. REX-1 algoritmasından bir diğer farkı ise kural çıkarma işlemi sırasında önce tek şarttan oluşan kurallar daha sonra iki şarttan oluşan kurallar... üretilir. Her aşama sonunda tüm örneklerin sınıflandırılıp sınıflandırılmadığına bakılarak işleme devam edip edilmeyeceği belirlenir.

Entropi ve bilgi kazancının nasıl hesaplanacağı Bölüm-2'de anlatılmıştır. Örnek setinin bilgi değeri (5.1)'deki denklem yardımıyla hesaplanır.

$$Bilgi(S) = - \sum_{i=1}^m \frac{S_i}{|S|} \cdot \log_2 \left( \frac{S_i}{|S|} \right) \quad (5.1)$$

Buradaki  $m$  örnek setindeki sınıf sayısını,  $|S|$  örnek setindeki örnek sayısını,  $S_i$  ise  $i$ . sınıfa ait örneklerin sayısını göstermektedir.

Hesaplamalarda bir sınıfa karşılık gelen değerler için ayrı ayrı entropiler hesaplanacaktır. Buna göre  $T_1, T_2, \dots, T_n$  bir değer geçtiği örneklerin oluşturduğu alt setleri gösterebilir.  $k$  alt setteki değer sayısını ve  $freq(C_k, T)$ ,  $T$  alt setindeki  $C_k$  sınıfına ait olan örneklerin sayısını ve  $|T|$  ise alt setteki örneklerin toplam sayısını göstermektedir. Buna göre her bir değer için entropi (5.2)'deki denklem yardımıyla hesaplanır [115].

$$E(T) = - \sum_{k=1}^K \frac{freq(C_k, T)}{|T|} \cdot \log_2 \frac{freq(C_k, T)}{|T|} \quad bit \quad (5.2)$$

Bir karakteristiğe ait entropi; karakteristikteki değerlerin entropilerinin ilgili değerlerin olasılıkları ile çarpımlarının toplamına eşittir (5.3) [115].

$$E(A) = \sum_{i=1}^n \frac{T_i}{|T|} \cdot E(T_i) \quad (5.3)$$

Buradaki,  $A$  bir karakteristiği,  $n$  karakteristikteki değer sayısını ve  $E(T_i)$  ise  $i$ . değer entropisini göstermektedir.

Bir karakteristiğe ait bilgi kazancı, örnek setinin bilgi değeri ile karakteristiğe ait entropi değerleri arasındaki farka eşittir.  $S$  örnek setindeki  $A$  karakteristiğine ait bilgi kazancı denklem (5.4)'teki gibi hesaplanır.  $Bilgi(S)$  tüm örnek setinin bilgi değeri olduğundan tüm karakteristikler için aynıdır.

$$Kazanç(S, A) = Bilgi(S) - E(A) \quad (5.4)$$

Her karakteristik için *kısmi bilgi* denklem (5.5) yardımıyla hesaplanır.

$$KısmiBilgi(S, A) = - \sum_{i=1}^{A_n} \frac{S_i}{S} \log_2 \frac{S_i}{S} \quad (5.5)$$

Burada  $S$  örnek setindeki  $A$  karakteristiğinin kısmi bilgisi hesaplanmaktadır.

$A_n$  :  $A$  karakteristiğinin değer sayısıdır.

$S_i$  :  $A$  karakteristiğindeki  $i$ . değer geçtiği örnek sayısıdır.

$S$  : Örnek setindeki toplam örnek sayısıdır.

Her karakteristik için bilgi kazanç oranları ise (5.6) yardımıyla hesaplanır.

$$KazançOranı(S, A) = \frac{Kazanç(S, A)}{KısmiBilgi(S, A)} \quad (5.6)$$

Bilgi kazanç oranlarının büyükten küçüğe doğru sıralanmasından sonra örnek seti bu sıralamaya göre yeniden düzenlenir.

### 5.2.1. Algoritma

Geliştirilen REX-2 algoritması Şekil 5.1'de adım adım gösterilmiştir.

( $N$ : karakteristik sayısı,  $n$  : kombinasyon sayısı)

- Adım-1.** Sayısal karakteristikler için aralık tanımlamalarının yapılması.
- Adım-2.** Verilen örnek setindeki her değer ve her karakteristiğin entropileri ile her karakteristiğin bilgi kazançları, kısmi bilgi ve bilgi kazanç oranları hesaplanır.
- Adım-3.** Karakteristikler *bilgi kazanç oranlarına* göre büyükten küçüğe doğru sıralanır ve örnek seti bu sıralamaya göre yeniden düzenlenir.
- Adım-4.**  $n=1$
- Adım-5. DO**  
     ÇIKIŞ=FALSE;  
     For I=1 To ÖrnekSayısı  
         EĞER I. örnek sınıflandırılmamış İSE  
             Örnekteki her karakteristik değerinden birer tane almak şartıyla  $n$ 'li kombinasyonlar oluştur  
             (KombSay : Kombinasyon sayısı).  
              $K=0$ ;  
             **DO**  
                  $K=K+1$ ;  
                 EVET=FALSE;  
                 K. kombinasyon örnek setindeki tüm örneklere uygulanır.  
                 EĞER  $n$  adet değerden oluşan K. Kombinasyon tüm örneklerde tek bir sınıfa karşılık geliyor İSE  
                     K. Kombinasyon kural haline getirilir.  
                     Sınıflandırılan örnekler işaretlenir. Evet=TRUE;  
             **WHILE**  $K \leq \text{KombSay}$  **OR** NOT EVET;  
         END FOR  
          $n=n+1$ ;  
         EĞER tüm örnekler sınıflandırılmış İSE ÇIKIŞ=TRUE;  
     **WHILE**  $n \leq N$  **OR** NOT ÇIKIŞ;
- Adım-6.** Aynı örnekleri temsil eden birden fazla kural varsa en genel kurallar seçilir.
- Adım-7. SON**

Şekil 5.1. REX-2 algoritması

### 5.2.2. Algoritmanın uygulanması

Geliştirilen bu algoritma, örnek setinden (öğrenme setinden) genel kurallar çıkaran verimli bir algoritmadır. Bu algoritma, Tablo 5.1’de verilen Golf Oynama örneği kullanılarak adım adım anlatılmıştır.

Tablo 5.1. Golf oynama problemi eğitim seti

| Örnek No | Hava     | Sıcaklık | Nem    | Rüzgar   | Karar  |
|----------|----------|----------|--------|----------|--------|
| 1        | Güneşli  | Yüksek   | Yüksek | Hafif    | Oynama |
| 2        | Güneşli  | Yüksek   | Yüksek | Kuvvetli | Oynama |
| 3        | Bulutlu  | Yüksek   | Yüksek | Hafif    | Oyna   |
| 4        | Yağmurlu | Orta     | Yüksek | Hafif    | Oyna   |
| 5        | Yağmurlu | Düşük    | Normal | Hafif    | Oyna   |
| 6        | Yağmurlu | Düşük    | Normal | Kuvvetli | Oynama |
| 7        | Bulutlu  | Düşük    | Normal | Kuvvetli | Oyna   |
| 8        | Güneşli  | Orta     | Yüksek | Hafif    | Oynama |
| 9        | Güneşli  | Düşük    | Normal | Hafif    | Oyna   |
| 10       | Yağmurlu | Orta     | Normal | Hafif    | Oyna   |
| 11       | Güneşli  | Orta     | Normal | Kuvvetli | Oyna   |
| 12       | Bulutlu  | Orta     | Yüksek | Kuvvetli | Oyna   |
| 13       | Bulutlu  | Yüksek   | Normal | Hafif    | Oyna   |
| 14       | Yağmurlu | Orta     | Yüksek | Kuvvetli | Oynama |

Tablo 5.1’de verilen örnek seti ondört örnekten ( $m = 14$ ), dört karakteristik (*Hava*, *Sıcaklık*, *Nem*, *Rüzgar*) ve iki sınıftan (*Oyna*, *Oynama*) oluşmaktadır. Örnekte yer alan karakteristikler ve bunlara ait değerler aşağıda verilmiştir.

| Karakteristik | Değerler                   |
|---------------|----------------------------|
| Hava          | Yağmurlu, Güneşli, Bulutlu |
| Sıcaklık      | Yüksek, Orta, Düşük        |
| Nem           | Normal, Yüksek             |
| Rüzgar        | Hafif, Kuvvetli            |

Bu bilgilere göre verilen örnek setini kullanarak algoritmanın adım adım uygulanması aşağıda açıklanmıştır:

#### Adım-1:

Her karakteristik ve değer için entropi değerleri hesaplanacaktır. *Hava* karakteristiği *Yağmurlu*, *Güneşli* ve *Bulutlu* olmak üzere üç değerden oluşmaktadır.

*Hava* karakteristiğinin *Yağmurlu* değeri 5 örnekte geçmektedir. Bu örneklerin üçü *Oyna*, ikisi *Oynama* sınıflarına aittir. Bu bilgilere göre  $\{Hava, Yağmurlu\}$  için entropi değeri;

$$E_{Hava, Yağmurlu} = -\frac{2}{5} \log_2 \frac{2}{5} - \frac{3}{5} \log_2 \frac{3}{5}$$

$$E_{Hava, Yağmurlu} = 0.971 \text{ bit olur.}$$

*Hava* karakteristiğinin *Güneşli* değeri 5 örnekte geçmektedir. Bu örneklerin üçü *Oynama* ve ikisi ise *Oyna* sınıfına aittir. Buna göre  $\{Hava, Güneşli\}$  için entropi değeri;

$$E_{Hava, Güneşli} = -\frac{2}{5} \log_2 \frac{2}{5} - \frac{3}{5} \log_2 \frac{3}{5}$$

$$E_{Hava, Güneşli} = 0.971 \text{ bit olur.}$$

*Hava* karakteristiğinin *Bulutlu* değeri 4 örnekte geçmektedir. Bütün örnekler *Oyna* sınıfına aittir. Buna göre  $\{Hava, Bulutlu\}$  için entropi değeri;

$$E_{Hava, Bulutlu} = -\frac{4}{4} \log_2 \frac{4}{4}$$

$$E_{Hava, Bulutlu} = 0 \text{ bit olarak hesaplanır.}$$

Bu bilgilere göre *Hava* karakteristiğinin entropi değeri aşağıdaki gibi olur:

$$E_{Hava} = \frac{5}{14} x E_{Hava, Yağmurlu} + \frac{5}{14} x E_{Hava, Güneşli} + \frac{4}{14} x E_{Hava, Bulutlu}$$

$$E_{Hava} = \frac{5}{14} x (0.971) + \frac{5}{14} x (0.971) + \frac{4}{14} x (0)$$

$$E_{Hava} = 0.694 \text{ bit}$$

İkinci karakteristik olan *Sıcaklık*, {*Yüksek*, *Düşük*, *Orta*} değerlerinden oluşmaktadır. Bu değerlerin entropileri aşağıdaki gibi hesaplanır:

*Sıcaklık* karakteristiğinin *Yüksek* değeri 4 örnekte geçmektedir. Bunların ikisi *Oyna* ve ikisi de *Oynama* sınıfına aittir. Buna göre {*Sıcaklık*, *Yüksek*} için entropi değeri;

$$E_{Sıcaklık,Yüksek} = -\frac{2}{4}\log_2 \frac{2}{4} - \frac{2}{4}\log_2 \frac{2}{4}$$

$$E_{Sıcaklık,Yüksek} = 1 \text{ bit olur.}$$

*Sıcaklık* karakteristiğinin ikinci değeri olan *Düşük*, 4 örnekte geçmektedir. Bu örneklerin üçü *Oyna* ve birisi de *Oynama* sınıfına aittir. Buna göre {*Sıcaklık*, *Düşük*} için entropi değeri;

$$E_{Sıcaklık,Düşük} = -\frac{3}{4}\log_2 \frac{3}{4} - \frac{1}{4}\log_2 \frac{1}{4}$$

$$E_{Sıcaklık,Düşük} = 0.811 \text{ bit olur.}$$

*Sıcaklık* karakteristiğinin son değeri olan *Orta*, 6 örnekte geçmektedir. Bu örneklerin ikisi *Oynama* ve dört tanesi ise *Oyna* sınıfına aittir. Buna göre {*Sıcaklık*, *Orta*} için entropi değeri;

$$E_{Sıcaklık,Orta} = -\frac{4}{6}\log_2 \frac{4}{6} - \frac{2}{6}\log_2 \frac{2}{6}$$

$$E_{Sıcaklık,Orta} = 0.918 \text{ bit olur.}$$

Bu bilgilere göre *Sıcaklık* karakteristiğinin entropisi aşağıdaki gibi hesaplanır:

$$E_{Sıcaklık} = \frac{4}{14}xE_{Sıcaklık,Yüksek} + \frac{4}{14}xE_{Sıcaklık,Düşük} + \frac{6}{14}xE_{Sıcaklık,Orta}$$

$$E_{Sıcaklık} = \frac{4}{14}x(1) + \frac{4}{14}x(0.811) + \frac{6}{14}x(0.918)$$

$$E_{Sıcaklık} = 0.911 \text{ bit}$$

Üçüncü karakteristik olan *Nem*,  $\{Yüksek, Normal\}$  değerlerinden oluşmaktadır. Bu karakteristikteki her değer için entropi değerleri hesaplanacaktır.

*Nem* karakteristiğinin ilk değeri olan *Yüksek* yedi örnekte geçmektedir. Bu örneklerin dört tanesi *Oynama* ve üç tanesi ise *Oyna* sınıfına aittir. Bu bilgilere göre  $\{Nem, Yüksek\}$  için entropi değeri;

$$E_{Nem, Yüksek} = -\frac{4}{7} \log_2 \frac{4}{7} - \frac{3}{7} \log_2 \frac{3}{7}$$

$$E_{Nem, Yüksek} = 0.985 \text{ bit olur.}$$

*Nem* karakteristiğinin ikinci değeri olan *Normal*, yedi örnekte geçmektedir. Bu örneklerin altı tanesi *Oyna* ve bir tanesi ise *Oynama* sınıfına aittir. Bu bilgilere göre  $\{Nem, Normal\}$  için entropi değeri;

$$E_{Nem, Normal} = -\frac{6}{7} \log_2 \frac{6}{7} - \frac{1}{7} \log_2 \frac{1}{7}$$

$$E_{Nem, Normal} = 0.592 \text{ bit olur.}$$

Bu bilgilere göre *Nem* karakteristiğinin entropisi aşağıdaki gibi hesaplanır:

$$E_{Nem} = \frac{7}{14} x E_{Nem, Yüksek} + \frac{7}{14} x E_{Nem, Normal}$$

$$E_{Nem} = \frac{7}{14} x (0.985) + \frac{7}{14} x (0.592)$$

$$E_{Nem} = 0.788 \text{ bit}$$

Dördüncü karakteristik olan *Rüzgar*,  $\{Hafif, Kuvvetli\}$  değerlerinden oluşmaktadır. Bu karakteristikteki her değer için entropi değerleri hesaplanacaktır.

*Rüzgar* karakteristiğinin ilk değeri olan *Hafif* sekiz örnekte geçmektedir. Bu örneklerin iki tanesi *Oynama* ve altı tanesi ise *Oyna* sınıfına aittir. Bu bilgilere göre  $\{Rüzgar, Hafif\}$  için entropi değeri;

$$E_{Rüzgar,Hafif} = -\frac{2}{8}\log_2 \frac{2}{8} - \frac{6}{8}\log_2 \frac{6}{8}$$

$$E_{Rüzgar,Hafif} = 0.811 \text{ bit olur.}$$

*Nem* karakteristiğinin ikinci değeri olan *Kuvvetli*, altı örnekte geçmektedir. Bu örneklerin üç tanesi *Oyna* ve üç tanesi ise *Oynama* sınıfına aittir. Bu bilgilere göre  $\{Rüzgar,Kuvvetli\}$  için entropi değeri;

$$E_{Rüzgar,Kuvvetli} = -\frac{3}{6}\log_2 \frac{3}{6} - \frac{3}{6}\log_2 \frac{3}{6}$$

$$E_{Rüzgar,Kuvvetli} = 1 \text{ bit olur.}$$

Bu bilgilere göre *Rüzgar* karakteristiğinin entropisi aşağıdaki gibi hesaplanır:

$$E_{Rüzgar} = \frac{8}{14} \times E_{Rüzgar,Hafif} + \frac{6}{14} \times E_{Rüzgar,Kuvvetli}$$

$$E_{Rüzgar} = \frac{8}{14} \times (0.811) + \frac{6}{14} \times (1)$$

$$E_{Rüzgar} = 0.892 \text{ bit}$$

Hesaplanan entropi değerleri Tablo 5.2’de toplu olarak verilmiştir.

Tablo 5.2. Karakteristikler ve değerlere ait entropi değerleri

| Karakteristik | Entropi (bit) | Değer    | Entropi (bit) |
|---------------|---------------|----------|---------------|
| Hava          | 0.694         | Yağmurlu | 0.971         |
|               |               | Güneşli  | 0.971         |
|               |               | Bulutlu  | 0             |
| Sıcaklık      | 0.911         | Yüksek   | 1             |
|               |               | Düşük    | 0.811         |
|               |               | Orta     | 0.918         |
| Nem           | 0.788         | Yüksek   | 0.985         |
|               |               | Normal   | 0.592         |
| Rüzgar        | 0.892         | Hafif    | 0.811         |
|               |               | Kuvvetli | 1             |

**Adım 2:**

Tablo 5.2'deki entropi değerlerine göre her karakteristik için *kısmi bilgi* değerleri hesaplanacaktır. Buna göre kısmi bilgiler aşağıdaki gibi hesaplanır:

$$Kısmi\ Bilgi(Hava) = -\frac{5}{14}\log_2 \frac{5}{14} - \frac{5}{14}\log_2 \frac{5}{14} - \frac{4}{14}\log_2 \frac{4}{14}$$

$$Kısmi\ Bilgi(Hava) = 1.577 \text{ bit}$$

$$Kısmi\ Bilgi(Sıcaklık) = -\frac{4}{14}\log_2 \frac{4}{14} - \frac{4}{14}\log_2 \frac{4}{14} - \frac{6}{14}\log_2 \frac{6}{14}$$

$$Kısmi\ Bilgi(Sıcaklık) = 1.577 \text{ bit}$$

$$Kısmi\ Bilgi(Nem) = -\frac{7}{14}\log_2 \frac{7}{14} - \frac{7}{14}\log_2 \frac{7}{14}$$

$$Kısmi\ Bilgi(Nem) = 1 \text{ bit}$$

$$Kısmi\ Bilgi(Rüzgar) = -\frac{8}{14}\log_2 \frac{8}{14} - \frac{6}{14}\log_2 \frac{6}{14}$$

$$Kısmi\ Bilgi(Rüzgar) = 0.985 \text{ bit}$$

Örnek setinin (S) bilgi ölçüsü aşağıdaki hesaplanır:

$$I(S) = -\frac{9}{14}\log_2 \frac{9}{14} - \frac{5}{14}\log_2 \frac{5}{14}$$

$$I(S) = 0.940 \text{ bit}$$

Bir  $A_x$  karakteristiği için bilgi kazancı ise aşağıdaki gibi hesaplanır:

$$Kazanç(A_x) = I(S) - E(A_x)$$

Buna göre *Hava* karakteristiği için bilgi kazanç değerleri aşağıdaki gibi hesaplanır:

$$Kazanç(Hava) = I(S) - E(Hava)$$

$$Kazanç(Hava) = 0.94 - 0.694$$

$$Kazanç(Hava) = 0.246 \text{ bit}$$

*Sıcaklık* karakteristiği için bilgi kazancı:

$$Kazanç(Sıcaklık) = 0.94 - 0.911$$

$$Kazanç(Sıcaklık) = 0.029 \text{ bit}$$

*Nem* karakteristiği için bilgi kazancı:

$$Kazanç(Nem) = 0.94 - 0.788$$

$$Kazanç(Nem) = 0.152 \text{ bit}$$

*Rüzgar* karakteristiği için bilgi kazancı:

$$Kazanç(Rüzgar) = 0.94 - 0.892$$

$$Kazanç(Rüzgar) = 0.048 \text{ bit}$$

Bir  $x$  karakteristiği için *Bilgi Kazanç Oranı* aşağıdaki gibi hesaplanır:

$$KazançOran(x) = \frac{Kazanç(x)}{KısmiBilgi(x)}$$

Buna göre *Hava* karakteristiği için *Bilgi Kazanç Oranı*:

$$KazançOran(Hava) = \frac{Kazanç(Hava)}{KısmiBilgi(Hava)}$$

$$KazançOran(Hava) = \frac{0.246}{1.577}$$

$$KazançOran(Hava) = 0.156 \text{ bit} \quad \text{olur.}$$

*Sıcaklık* karakteristiği için *Bilgi Kazanç Oranı*:

$$KazançOran(Sıcaklık) = \frac{0.029}{1.577}$$

$$KazançOran(Sıcaklık) = 0.018 \text{ bit olur.}$$

*Nem karakteristiği için Bilgi Kazanç Oranı:*

$$KazançOran(Nem) = \frac{0.152}{1}$$

$$KazançOran(Nem) = 0.152 \text{ bit olur.}$$

*Rüzgar karakteristiği için Bilgi Kazanç Oranı:*

$$KazançOran(Rüzgar) = \frac{0.048}{0.985}$$

$$KazançOran(Rüzgar) = 0.049 \text{ bit olur.}$$

### **Adım-3:**

Adım-2’de hesaplanan bilgilere göre *Bilgi Kazanç Oranları* aşağıdaki gibi büyükten küçüğe doğru sıralanır:

$$\text{Hava (0.156)} > \text{Nem (0.152)} > \text{Rüzgar (0.049)} > \text{Sıcaklık (0.018)}$$

Bu sıralamaya göre Tablo 5.1’deki örnekler seti yeniden düzenlenir. *Hava*, *Nem*, *Rüzgar* ve *Sıcaklık* karakteristiklerine göre yeniden düzenlenen örnek seti Tablo 5.3’te verilmiştir.

### **Adım-4:**

Birinci Aşama : n=1 olarak alınır.

### **Adım-5:**

Örnekte bulunan değerler ile tekli kombinasyonlar oluşturulur. Bu kombinasyonlar {Hava:Güneşli}, {Nem:Yüksek}, {Rüzgar:Hafif} ve {Sıcaklık:Yüksek} şeklindedir.

Tablo 5.3. Yeniden düzenlenmiş örnek seti

| Örnek No | Hava     | Nem    | Rüzgar   | Sıcaklık | Karar  |
|----------|----------|--------|----------|----------|--------|
| 1        | Güneşli  | Yüksek | Hafif    | Yüksek   | Oynama |
| 2        | Güneşli  | Yüksek | Kuvvetli | Yüksek   | Oynama |
| 3        | Bulutlu  | Yüksek | Hafif    | Yüksek   | Oyna   |
| 4        | Yağmurlu | Yüksek | Hafif    | Orta     | Oyna   |
| 5        | Yağmurlu | Normal | Hafif    | Düşük    | Oyna   |
| 6        | Yağmurlu | Normal | Kuvvetli | Düşük    | Oynama |
| 7        | Bulutlu  | Normal | Kuvvetli | Düşük    | Oyna   |
| 8        | Güneşli  | Yüksek | Hafif    | Orta     | Oynama |
| 9        | Güneşli  | Normal | Hafif    | Düşük    | Oyna   |
| 10       | Yağmurlu | Normal | Hafif    | Orta     | Oyna   |
| 11       | Güneşli  | Normal | Kuvvetli | Orta     | Oyna   |
| 12       | Bulutlu  | Yüksek | Kuvvetli | Orta     | Oyna   |
| 13       | Bulutlu  | Normal | Hafif    | Yüksek   | Oyna   |
| 14       | Yağmurlu | Yüksek | Kuvvetli | Orta     | Oynama |

**Adım-7:**

Bu kombinasyonlardan ilki olan {Hava:*Güneşli*} tüm örneklerde sadece bir sınıfta geçmediğinden tek başına kural oluşturamaz. Diğer kombinasyonlar da sadece bir sınıfta geçmediklerinden dolayı kural oluşturamazlar.

İkinci örnek seçilir. Örnekte bulunan değerler ile tekli kombinasyonlar oluşturulur. Bu kombinasyonlar {Hava:*Güneşli*}, {Nem:*Yüksek*}, {Rüzgar:*Kuvvetli*} ve {Sıcaklık:*Yüksek*} şeklindedir. Bu kombinasyonlardan hiçbiri tek başına kural oluşturamaz.

Üçüncü örnek seçilir. Örnekte bulunan değerler ile tekli kombinasyonlar oluşturulur. Bu kombinasyonlar {Hava:*Bulutlu*}, {Nem:*Yüksek*}, {Rüzgar:*Hafif*} ve {Sıcaklık:*Yüksek*} şeklindedir. Bu kombinasyonlardan ilki olan {Hava:*Bulutlu*} sadece bir sınıfta geçtiğinden dolayı kural oluşturur.

**Kural 1. EĞER Hava=Bulutlu İSE Karar=Oyna**

Bu kural ile 3, 7, 12 ve 13. örnekler sınıflandırılır ve bu örnekler daha sonra dikkate alınmamak üzere işaretlenir.

Sınıflandırılmayan diğer örnekler bakıldığında tekli kombinasyonlarla başka kural çıkarılamayacağından dolayı (entropisi sıfır olan tek değer Hava:*Bulutlu* değeri idi.) ilk aşama tamamlanır ve ikinci aşamaya geçilir. İkinci aşamada sadece sınıflandırılmamış örnekler bakılır.

#### **Adım-4**

İkinci Aşama:  $n=2$  olarak alınır. Bu aşamada sınıflandırılmayan örneklerin ikili kombinasyonlarına göre kural çıkarma işlemi yapılacaktır.

Sınıflandırılmayan ilk örnek olan ikinci örnek seçilir. Örnekte bulunan değerler ile ikili kombinasyonlar oluşturulur. Bu kombinasyonların ilki olan {Hava:*Güneşli*, Nem:*Yüksek*} kombinasyonu 1, 2 ve 8 nolu örneklerde ve sadece bir sınıfta geçmektedir. Bundan dolayı kural haline getirilir.

#### ***Kural 2 : EĞER Hava=Güneşli VE Nem=Yüksek İSE Karar=Oynama***

Bu kural ile 1, 2 ve 8. örnekler sınıflandırılır ve bu örnekler daha sonra dikkate alınmamak üzere işaretlenir.

Sınıflandırılmayan bir sonraki örnek, 1, 2 ve 3 nolu örnekler sınıflandırıldığından dolayı 4 nolu örnektir. Bu örnekteki değerlerin ikili kombinasyonları oluşturulur. Bu kombinasyonlardan {Hava:*Yağmurlu*, Rüzgar:*Hafif*} tüm örnek setinde sadece bir sınıfta geçtiklerinden dolayı kural oluşturur.

#### ***Kural 3: EĞER Hava=Yağmurlu VE Rüzgar=Hafif İSE Karar=Oyna***

Bu kural ile 4, 5 ve 10 nolu örnekler sınıflandırılır ve bu örnekler daha sonra dikkate alınmamak üzere işaretlenir.

Sınıflandırılmayan bir sonraki örnek olarak 6 nolu örnek seçilir. Bu örnekteki değerlerin ikili kombinasyonları oluşturulur. Bu kombinasyonlardan {Hava:*Yağmurlu*, Rüzgar:*Kuvvetli*} örnek setinde sadece bir sınıfta geçtiğinden dolayı kural oluşturur.

**Kural 4: EĞER Hava=Yağmurlu VE Rüzgar=Kuvvetli İSE Karar=Oynama**

Bu kural ile 6 ve 14 nolu örnekler sınıflandırılır ve bu örnekler işaretlenir.

Sınıflandırılmayan bir sonraki örnek ise 9 nolu örnektir. Bu örnekteki değerlerin ikili kombinasyonları oluşturulur. Bu kombinasyonlardan {Hava:Güneşli, Nem:Normal} örnek setinde sadece bir sınıfta geçtiğinden kural oluşturur.

**Kural 5: EĞER Hava=Güneşli VE Nem=Normal İSE Karar=Oyna**

Bu kural ile 9 ve 11 nolu örnekler sınıflandırılır ve bu örneklerde işaretlenir.

Sınıflandırılacak başka örnek kalmadığından dolayı işlem tamamlanır. REX-2 algoritması kullanılarak elde edilen kurallar Tablo 5.4'te gösterilmiştir.

Tablo 5.4. REX-2 algoritması ile üretilen kurallar (Golf problemi)

| Kural | Kural Tanımı                                           |
|-------|--------------------------------------------------------|
| 1     | EĞER Hava=Bulutlu İSE Karar=Oyna                       |
| 2     | EĞER Hava=Güneşli VE Nem=Yüksek İSE Karar=Oynama       |
| 3     | EĞER Hava=Yağmurlu VE Rüzgar=Hafif İSE Karar=Oyna      |
| 4     | EĞER Hava=Yağmurlu VE Rüzgar=Kuvvetli İSE Karar=Oynama |
| 5     | EĞER Hava=Güneşli VE Nem=Normal İSE Karar=Oyna         |

### 5.3. Diğer Algoritmalarla Karşılaştırma

REX-2 algoritması, REX-1, C4.5, ID3, RULES-3, RULES-3 Plus ve RULES-4 algoritmalarıyla karşılaştırılacaktır.

#### 5.3.1. REX-1 algoritması ile karşılaştırma

Golf oynama problemini kullanarak, REX-1 algoritması Tablo 5.5'te gösterilen kuralları üretir. Her iki algoritmanın ürettiği kurallar ve kural sayıları aynıdır. Ancak veri setinin sıralama şekline göre aynı kural içerisindeki şartların yerleri değişebilmektedir.

Tablo 5.5. REX-1 algoritması ile üretilen kurallar (Golf problemi)

| Kural | Kural Tanımı                                           |
|-------|--------------------------------------------------------|
| 1     | EĞER Hava=Bulutlu İSE Karar=Oyna                       |
| 2     | EĞER Nem=Yüksek VE Hava=Güneşli İSE Karar=Oynama       |
| 3     | EĞER Rüzgar=Hafif VE Hava=Yağmurlu İSE Karar=Oyna      |
| 4     | EĞER Rüzgar=Kuvvetli VE Hava=Yağmurlu İSE Karar=Oynama |
| 5     | EĞER Nem=Normal VE Hava=Güneşli İSE Karar=Oyna         |

### 5.3.2. C4.5 algoritması ile karşılaştırma

Golf problemi için C4.5 algoritması tarafından üretilen kurallar Tablo 5.6'da verilmiştir. Buradan da görüldüğü gibi REX-2 ve C4.5 algoritmaları aynı kuralları üretmişlerdir.

Tablo 5.6. C4.5 algoritması ile üretilen kurallar (Golf Problemi)

| Kural | Kural Tanımı                                           |
|-------|--------------------------------------------------------|
| 1     | EĞER Hava=Bulutlu İSE Karar=Oyna                       |
| 2     | EĞER Hava=Güneşli VE Nem=Yüksek İSE Karar=Oynama       |
| 3     | EĞER Hava=Yağmurlu VE Rüzgar=Hafif İSE Karar=Oyna      |
| 4     | EĞER Hava=Yağmurlu VE Rüzgar=Kuvvetli İSE Karar=Oynama |
| 5     | EĞER Hava=Güneşli VE Nem=Normal İSE Karar=Oyna         |

### 5.3.3. IRIS veri setini kullanarak diğer algoritmalarla karşılaştırma

REX-2 algoritmasının IRIS veri setine uygulanması sonucunda elde edilen kurallar Tablo 5.7'de gösterilmiştir.

Tablo 5.7. REX-2 tarafından elde edilen kurallar (IRIS veri seti)

| Kural | Kural Tanımı                                                                |
|-------|-----------------------------------------------------------------------------|
| 1     | EĞER $1 \leq PL < 1.98$ İSE KARAR=Iris-setosa                               |
| 2     | EĞER $1.7 \leq PW < 2.1$ İSE KARAR=Iris-virginica                           |
| 3     | EĞER $0.9 \leq PW < 1.3$ İSE KARAR=Iris-versicolor                          |
| 4     | EĞER $2.1 \leq PW < 2.5$ İSE KARAR=Iris-virginica                           |
| 5     | EĞER $3.95 \leq PL < 4.93$ VE $1.3 \leq PW < 1.7$ İSE KARAR=Iris-versicolor |
| 6     | EĞER $4.93 \leq PL < 5.91$ VE $2.8 \leq SW < 3.2$ İSE KARAR=Iris-virginica  |
| 7     | EĞER $1.3 \leq PW < 1.7$ VE $2.4 \leq SW < 2.8$ İSE KARAR=Iris-versicolor   |

REX-1 algoritması tarafından üretilen kural sayısı sekiz ve şart sayısı 11 iken (Bkz. Bölüm-4), REX-2 algoritması tarafından üretilen kural sayısı 7'ye ve şart sayısı ise

10'a düşmüştür. Buna karşın ID3 tarafından üretilen kural sayısı 8 ve şart sayısı ise 14'tür. ID3, Rules-3, Rules-3 Plus, Rules-4 ve REX-1 ile yapılan karşılaştırmalar Tablo 5.8'de verilmiştir.

Rules-4 algoritması tarafından üretilen kural sayısı 9 ve şart sayısı ise 12'dir. REX-1 ve REX-2 algoritmalarının ürettiği kural sayıları ve kurallardaki şart sayıları diğer RULES ailesi algoritmalarından daha azdır. Ayrıca, IRIS test veri seti kullanıldığında; Rules-4 algoritması %93.60, REX-1 algoritması %93.75 ve REX-2 algoritması ise %100 doğruluk oranında sınıflandırma yapabilmektedir [97, 98, 118]. Bu bilgilere göre REX-2 algoritması, hem az sayıda kural çıkarmakta hem de yüksek doğruluk oranında sınıflandırma yapabilmektedir.

Tablo 5.8. Kural sayısı ve kural başına şartların ortalaması (IRIS veri seti)

| Algoritma    | Kural Sayısı | Şart Sayısı | Kural Başına Ortalama Şart Sayısı |
|--------------|--------------|-------------|-----------------------------------|
| RULES-3      | 11           | 17          | 1.545                             |
| ID3          | 8            | 14          | 1.75                              |
| RULES-3 PLUS | 10           | 14          | 1.4                               |
| RULES-4      | 9            | 12          | 1.333                             |
| REX-1        | 8            | 11          | 1.375                             |
| REX-2        | 7            | 10          | 1.429                             |

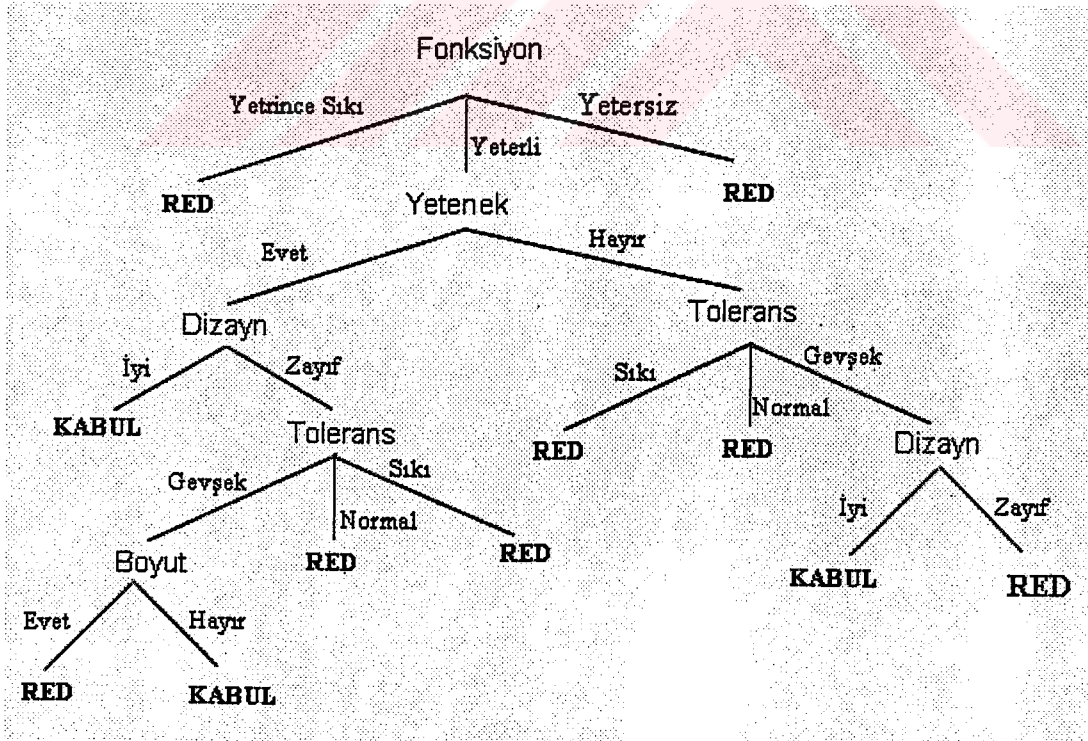
#### 5.3.4. ID3 Algoritması ile karşılaştırma

Tablo 5.9'da enjeksiyon-kalıp parçaları için tolerans dizaynı ile ilgili bir örnek seti verilmiştir[116]. Örnek setinde kullanılan karakteristikler: Tolerans tanımı (Tolerans), parçanın fonksiyonel gerektirmeleri karşılama derecesi (Fonksiyon), boyut ayırma özelliği (Boyut), kalıp dizaynı (Dizayn) ve makine yeteneği (Yetenek).

ID3 tarafından üretilen karar ağacından (Şekil 5.2) elde edilen kurallar (ağacın her dalı bir kural oluşturur) Tablo 5.10'da verilmiştir.

Tablo 5.9. Tolerans dizaynı için örnek seti [116]

| Örnek No | Tolerans | Fonksiyon  | Boyut | Dizayn | Yetenek | Karar |
|----------|----------|------------|-------|--------|---------|-------|
| 1        | Sıkı     | Yeterli    | Hayır | İyi    | Evet    | Kabul |
| 2        | Sıkı     | Yeterli    | Evet  | İyi    | Evet    | Kabul |
| 3        | Sıkı     | Yeterli    | Evet  | İyi    | Hayır   | Red   |
| 4        | Sıkı     | Yeterli    | Hayır | İyi    | Hayır   | Red   |
| 5        | Sıkı     | Aşırı Sıkı | Evet  | İyi    | Evet    | Red   |
| 6        | Sıkı     | Aşırı Sıkı | Hayır | İyi    | Hayır   | Red   |
| 7        | Sıkı     | Yeterli    | Hayır | Zayıf  | Evet    | Red   |
| 8        | Normal   | Yeterli    | Hayır | İyi    | Evet    | Kabul |
| 9        | Normal   | Yeterli    | Hayır | Zayıf  | Hayır   | Red   |
| 10       | Normal   | Yeterli    | Hayır | Zayıf  | Evet    | Red   |
| 11       | Normal   | Yeterli    | Evet  | Zayıf  | Hayır   | Red   |
| 12       | Normal   | Aşırı Sıkı | Hayır | İyi    | Evet    | Red   |
| 13       | Normal   | Yetersiz   | Hayır | İyi    | Evet    | Red   |
| 14       | Normal   | Yeterli    | Hayır | İyi    | Hayır   | Red   |
| 15       | Normal   | Yeterli    | Evet  | İyi    | Hayır   | Red   |
| 16       | Normal   | Yeterli    | Evet  | İyi    | Evet    | Kabul |
| 17       | Gevşek   | Yeterli    | Evet  | Zayıf  | Evet    | Red   |
| 18       | Gevşek   | Yeterli    | Evet  | İyi    | Evet    | Kabul |
| 19       | Gevşek   | Yeterli    | Evet  | Zayıf  | Hayır   | Red   |
| 20       | Gevşek   | Yeterli    | Hayır | Zayıf  | Hayır   | Red   |
| 21       | Gevşek   | Yeterli    | Hayır | İyi    | Hayır   | Kabul |
| 22       | Gevşek   | Yetersiz   | Hayır | İyi    | Evet    | Red   |
| 23       | Gevşek   | Yeterli    | Hayır | Zayıf  | Evet    | Kabul |
| 24       | Gevşek   | Yetersiz   | Evet  | İyi    | Evet    | Red   |



Şekil 5.2. ID3 tarafından üretilen karar ağacı [116]

Tablo 5.10. ID3 tarafından elde edilen kurallar (Şekil 5.2'deki karar ağacından)

| Kural No | Kural Tanımı                                                                                             |
|----------|----------------------------------------------------------------------------------------------------------|
| 1        | EĞER Fonksiyon=Yeterince Sıkı İSE Karar=RED                                                              |
| 2        | EĞER Fonksiyon=Yetersiz İSE Karar=RED                                                                    |
| 3        | EĞER Fonksiyon=Yeterli VE Yetenek=Evet VE Dizayn=İyi İSE Karar=KABUL                                     |
| 4        | EĞER Fonksiyon=Yeterli VE Yetenek=Evet VE Dizayn=Zayıf VE Tolerans=Gevşek VE Boyut=Evet İSE Karar=RED    |
| 5        | EĞER Fonksiyon=Yeterli VE Yetenek=Evet VE Dizayn=Zayıf VE Tolerans=Gevşek VE Boyut=Hayır İSE Karar=KABUL |
| 6        | EĞER Fonksiyon=Yeterli VE Yetenek=Evet VE Dizayn=Zayıf VE Tolerans=Normal İSE Karar=RED                  |
| 7        | EĞER Fonksiyon=Yeterli VE Yetenek=Evet VE Dizayn=Zayıf VE Tolerans=Sıkı İSE Karar=RED                    |
| 8        | EĞER Fonksiyon=Yeterli VE Yetenek=Hayır VE Tolerans=Sıkı İSE Karar=RED                                   |
| 9        | EĞER Fonksiyon=Yeterli VE Yetenek=Hayır VE Tolerans=Normal İSE Karar=RED                                 |
| 10       | EĞER Fonksiyon=Yeterli VE Yetenek=Hayır VE Tolerans=Gevşek VE Dizayn=İyi İSE Karar=KABUL                 |
| 11       | EĞER Fonksiyon=Yeterli VE Yetenek=Hayır VE Tolerans=Gevşek VE Dizayn=Zayıf İSE Karar=RED                 |

REX-2 algoritması tarafından elde edilen kurallar Tablo 5.11'de verilmiştir. ID3 algoritması tarafından elde edilen kural sayısı 11 ve toplam şart sayısı 36'dır. Buna karşın, REX-2 algoritması tarafından elde edilen kural sayısı 8 ve toplam şart sayısı ise sadece 15'tir.

Tablo 5.11. REX-2 tarafından elde edilen kurallar

| Kural No | Kural Tanımı                                                         |
|----------|----------------------------------------------------------------------|
| 1        | EĞER Fonksiyon=Aşırı Sıkı İSE Karar=RED                              |
| 2        | EĞER Fonksiyon=Yetersiz İSE Karar=RED                                |
| 3        | EĞER Fonksiyon=Yeterli İSE Karar=KABUL                               |
| 4        | EĞER Yetenek=Hayır VE Tolerans=Sıkı İSE Karar=RED                    |
| 5        | EĞER Fonksiyon=Yeterli VE Dizayn=Zayıf İSE Karar=RED                 |
| 6        | EĞER Yetenek=Hayır VE Tolerans=Normal İSE Karar=RED                  |
| 7        | EĞER Fonksiyon=Yeterli VE Yetenek=Evet VE Dizayn=İyi İSE Karar=KABUL |
| 8        | EĞER Yetenek=Hayır VE Dizayn=İyi VE Tolerans=Gevşek İSE Karar=KABUL  |

REX-2 algoritması tarafından elde edilen kuralların hangi örnekleri sınıflandırdığı Tablo 5.10'da verilmiştir. 3 ve 8 nolu kuralların sadece bir örneği sınıflandırdığı açıkça görülmektedir. Bu kurallar özel kural olarak adlandırılırlar. Amaç genel kurallar elde etmek olduğundan dolayı, kabul edilebilir bir hata oranına göre bu kurallar iptal edilebilir.

Tablo 5.12. Kurallar tarafından sınıflandırılan örnekler

| Kural No | Sınıflandırılan Örnekler       |
|----------|--------------------------------|
| 1        | 5 – 6 – 12                     |
| 2        | 13 – 22 – 24                   |
| 3        | 23                             |
| 4        | 3 – 4 – 6                      |
| 5        | 7 – 9 – 10 – 11 – 17 – 19 – 20 |
| 6        | 9 – 11 – 14 – 15               |
| 7        | 1 – 2 – 8 – 16 – 18            |
| 8        | 21                             |

#### 5.4. Sonuçlar

REX-2 algoritması sadece entropi değil aynı zamanda bilgi kazanç oranını da kullanarak, örnek setine ait bilgi değerinin daha iyi ölçülmesini sağlar ve bu özelliğe göre örnek setini düzenleyerek kural çıkarma işlemini gerçekleştirir. Yapılan test işlemlerinde (Iris veri setini kullanarak) REX-1 algoritmasına göre daha iyi sonuçlar elde edildiğini göstermektedir.

REX-2 algoritması Rules-3, Rules-3 Plus, Rules-4 ve ID3 algoritmalarından daha iyi sonuçlar üretmiştir. Algoritma, hem ürettiği kural sayısı hem de kurallardaki toplam şart sayısı bakımından diğer algoritmalara göre daha iyi sonuçlar vermiştir.

Endüktif öğrenme algoritmaları ile ilgili gerçek hayattan alınan veri setleri kullanılarak yapılan geniş karşılaştırmalar Bölüm-7’de verilmiştir.

## BÖLÜM 6. REX-3 : ENDÜKTİF ÖĞRENME ALGORİTMASI

### 6.1. Giriş

Bu bölümde, REX (Rule Extraction) ailesi algoritmalarının yeni bir versiyonu olarak geliştirilen REX-3 algoritması anlatılacaktır. Bu algoritmanın geliştirilmesindeki amaç, kural çıkarma işlemini hiçbir hesaplama (entropi, bilgi kazancı,...vb.) yapmadan gerçekleştirmektir.

Bölüm-4'te geliştirilen REX-1 algoritmasında her karakteristik için önce *entropi* değerleri ardında da entropi değerleri kullanılarak *düzeltilmiş entropi* değerleri hesaplanmaktaydı. Bölüm-5'te geliştirilen REX-2 algoritmasında ise öncelikle her karakteristiğin entropileri ile bilgi kazançları ve daha sonra her karakteristiğin kısmi bilgi değerleri ile bilgi kazanç oranları hesaplanmaktaydı. Her iki algoritmada da örnek seti, hesaplanan bu değerlere göre yeniden düzenlenmekte ve ardından kural çıkarma işlemi yapılmaktaydı. Bu algoritmada entropi ve bilgi kazancı gibi herhangi bir değer hesaplanmamakta ve örnek seti de orijinal şekliyle ele alınarak kural çıkarma işlemi gerçekleştirilmektedir.

REX-3 algoritması, REX-1 ve REX-2 algoritmalarında yapılan hesaplama (entropi, bilgi kazancı) işlemlerini ortadan kaldırmak ve daha çok örneği sınıflandırabilen kuralları seçmek için geliştirilmiştir. Daha çok örnek sınıflandıran kuralların seçilmesi durumunda üretilcek toplam kural sayısı da düşecektir. REX-3 algoritmasında entropi ve bilgi kazancı gibi değerlerin hesaplanmamasından dolayı algoritma daha az karmaşıklığa sahip olmaktadır.

Bu bölümde, algoritma bir örnek üzerinde adım adım anlatılacak ve IRIS veri seti kullanılarak elde edilen sonuçlar diğer algoritmalarla karşılaştırılacaktır.

## 6.2. Algoritmanın Tanımı

RULES ailesi algoritmaları (Rules-3, Rules-3 Plus, Rules-4) kural çıkarma işlemine ilk örnekten başlayarak, ilk önce aynı örnek üzerinde tekli kombinasyonlardan hangisinin kural çıkarabildiğine bakılmakta ve tüm örnek setinde sadece bir sınıfa ait olan karakteristik değeri kural haline getirilmekte idi. Şayet tekli karakteristik değerleri ile kural oluşturulamıyorsa bu durumda sırasıyla ikili, üçlü,... kombinasyonların kural oluşturup oluşturamayacağına bakılmaktadır. İlk kural oluşturulduktan sonra geriye kalan kombinasyonlara bakılmamaktadır.

Geliştirilen bu algorithmada ise geriye kalan diğer kombinasyonlara da bakılmaktadır. Şayet ilk bulunan kural tekli kombinasyonlardan birine ait ise geriye kalan tekli kombinasyonların da kural çıkarıp çıkarmadığı araştırılmaktadır. Şayet ilk olarak ikili kombinasyonlardan biri kural oluşturmuş ise bu durumda geriye kalan diğer ikili kombinasyonlardan hangilerinin kural oluşturabildiklerine de bakılmaktadır. Kural oluşturabilen kombinasyonlar arasından en çok örneği sınıflandıran kural seçilmekte ve diğer kurallar ise iptal edilmektedir. Böylece en genel kural seçimi yapılmış olmaktadır.

Algoritmanın bu özelliğinden dolayı entropi ve bilgi kazançlarının hesaplanmasına ve örnek setinin yeniden düzenlenmesine gerek kalmamaktadır. Çünkü, bilgi değeri yüksek olan karakteristikler aynı zamanda çok sayıda örnek için kural oluşturmaktadırlar.

### 6.2.1. Algoritmanın özellikleri

Bu algoritmanın özellikleri aşağıdaki gibi sayılabilir:

- Bu algoritma örnek seti üzerinde hiçbir istatistiksel işlem yapmamaktadır. İlk örnekten başlayarak doğrudan kural çıkarmaya başladığından dolayı çok basit bir algorithma sahiptir. Bu özellik programlama mantığı bakımından büyük kolaylık sağlamaktadır.

- Algoritma eksik değer içeren örneklere göre de işlem yapabilmektedir. Bazı örneklerin eksik değer içermesi kaçınılmaz bir durumdur. Bilgi girişi sırasında boş bırakılan değerler kural çıkarma işlemi sırasında dikkate alınmamaktadır.
- Sayısal değerler için aralık tanımlamalarının yapılmasına izin verir. Bu özellik yardımıyla daha genel kuralların elde edilmesi sağlanır.
- Bu algoritmanın diğer algoritmalara göre işlem süresi biraz daha uzundur. Çünkü kural oluşturan ilk kombinasyondan hemen sonra işlem son bulmamakta diğer kombinasyonlara da bakılmaktadır. Ancak diğer algoritmalarda yapılan hesaplamalar ve örnek setinin yeniden düzenlenmesi gibi işlemler burada yapılmamaktadır. Bu ise zaman bakımından bir avantajdır.

### 6.2.2. Algoritma

Geliştirilen REX-3 algoritması Şekil 6.1’de adım adım gösterilmiştir.

### 6.2.3. Algoritmanın uygulanması

Tablo 6.1’de bir yatırım projesinin uygulanıp uygulanmayacağına karar vermek için dört karakteristikten oluşan bir örnek seti verilmiştir. REX-3 algoritmasını kullanarak bu örnek setinden kurallar elde etmek için algoritma adım adım uygulanacaktır.

#### Adım 1.

Sayısal karakteristik olmadığından dolayı aralık tanımlama işlemi yapılmayacaktır.

#### Adım 2.

*Birinci Aşama:*  $n=1$  olarak alınır. Tekli kombinasyonlara göre kural çıkarma işlemi gerçekleştirilecektir. Bunun için sınıflandırılmayan örneklerin kural oluşturup oluşturmadıklarına bakmak için sırasıyla deneneceklerdir.

|                                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|---------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| (N: karakteristik sayısı, n : kombinasyon sayısı) |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>Adım-1.</b>                                    | Sayısal karakteristikler için aralık tanımlamalarının yapılması.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <b>Adım-2.</b>                                    | n=1 (Birinci Aşama)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <b>Adım-3.</b>                                    | <b>DO</b><br><b>ÇIKIŞ</b> =FALSE;<br><b>FOR</b> I=1 <b>TO</b> ÖrnekSayısı<br><b>EĞER</b> I. örnek sınıflandırılmamış <b>İSE</b><br>Örnekteki her karakteristik değerinden birer tane almak şartıyla n’li kombinasyonlar oluşturun<br>(KombSay : Kombinasyon sayısı).<br>K=0;<br><b>DO</b> // Tüm kombinasyonlara bakılıyor<br>K=K+1;<br>K. kombinasyon örnek setindeki tüm örneklerle uygulanır.<br><b>EĞER</b> n adet değerden oluşan K. Kombinasyon tüm örneklerde tek bir sınıfa karşılık geliyor <b>İSE</b><br>K. kombinasyon kural haline getirilir.<br>Sınıflandırılan örnekler işaretlenir.<br><br><b>WHILE</b> K<=KombSay;<br>En çok örnek sınıflandırabilen kural seçilir.<br><b>END FOR</b><br><b>EĞER</b> tüm örnekler sınıflandırılmış <b>İSE</b> <b>ÇIKIŞ</b> =TRUE;<br>n=n+1; (Aşama sayısı +1 kadar artırılır)<br><b>WHILE</b> n<N <b>OR NOT</b> <b>ÇIKIŞ</b> ; |
| <b>Adım-4.</b>                                    | Aynı örnekleri temsil eden birden fazla kural varsa en genel kurallar seçilir.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>Adım-5.</b>                                    | SON                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |

Şekil 6.1. REX-3 algoritması

Tablo 6.1. Yatırım Projesi-2 veri seti [117]

| Örnek No | Risk   | Kârlılık | Geri Dönüş Zamanı | Yatırım Seviyesi | Karar |
|----------|--------|----------|-------------------|------------------|-------|
| 1        | Yüksek | Önemli   | Uzun              | Yüksek           | RED   |
| 2        | Yüksek | Önemli   | Uzun              | Düşük            | RED   |
| 3        | Düşük  | Önemli   | Uzun              | Yüksek           | KABUL |
| 4        | Orta   | Orta     | Uzun              | Yüksek           | KABUL |
| 5        | Orta   | Küçük    | Kısa              | Yüksek           | KABUL |
| 6        | Orta   | Küçük    | Kısa              | Düşük            | RED   |
| 7        | Düşük  | Küçük    | Kısa              | Düşük            | KABUL |
| 8        | Yüksek | Orta     | Uzun              | Yüksek           | RED   |
| 9        | Yüksek | Küçük    | Kısa              | Yüksek           | KABUL |
| 10       | Orta   | Orta     | Kısa              | Yüksek           | KABUL |
| 11       | Yüksek | Orta     | Kısa              | Düşük            | KABUL |
| 12       | Düşük  | Orta     | Uzun              | Düşük            | KABUL |
| 13       | Düşük  | Önemli   | Kısa              | Yüksek           | KABUL |
| 14       | Orta   | Orta     | Uzun              | Düşük            | RED   |

(Karar=KABUL:Yatırım projesinin kabul edilmesi, Karar=RED:Yatırım projesinin kabul edilmemesi)

İlk olarak birinci örnek seçilir. Bu örnek {Risk:*Orta*, Kârlılık:*Orta*, Zaman:*Uzun*, Seviye:*Yüksek*} karakteristik-değerlerinden oluşmaktadır. Bu örnekteki değerlerin tekli kombinasyonları tek tek denendiğinde her değer örnek setinde birden fazla sınıfa karşılık geldiğinden dolayı tek başına kural oluşturamaz. Sonraki örneğe geçilir.

İkinci örnek, {Risk:*Yüksek*, Kârlılık:*Önemli*, Zaman:*Uzun*, Seviye:*Düşük*} karakteristik-değerlerinden oluşmaktadır. Bu değerler de tek başlarına kural oluşturamazlar. Sonraki örneğe geçilir.

Üçüncü örnek, {Risk:*Düşük*, Kârlılık:*Önemli*, Zaman:*Uzun*, Seviye:*Yüksek*} karakteristik-değerlerinden oluşmaktadır. Bu değerlerin tümünün örnek setinde sadece tek sınıfa karşılık gelip gelmediklerine bakılır. İlk değer olan {Risk:*Düşük*} 3, 7, 12 ve 13 nolu örneklerde sadece *KABUL* sınıfına karşılık gelmektedir. Bu değer tek başına aşağıdaki kuralı oluşturur.

**Kural 1 : EĞER Risk=Düşük İSE Karar=KABUL**

Bu kural ile 3, 7, 12 ve 13 nolu sınıflandırılır ve bir daha sonra dikkate alınmamak üzere işaretlenir. Üçüncü örnekteki diğer değerlerin hiçbirisi tek başına kural oluşturamaz. Sınıflandırılmayan bir sonraki örneğe geçilir.

Dördüncü örnek {Risk:*Yüksek*, Kârlılık:*Önemli*, Zaman:*Uzun*, Seviye:*Düşük*} karakteristik-değerlerinden oluşmaktadır. Bu değerler de tek başlarına kural oluşturamazlar. Bir sonraki örnek olan beşinci örneğe geçilir.

Beşinci örnekte yer alan karakteristik değerlerinden hiçbirisi tek başına kural oluşturamaz. Bir sonraki örnek olan altıncı örnekteki karakteristik değerleri de tek başlarına kural oluşturamazlar. Yedinci örnek daha önce işaretlendiğinden (Kural-1) dolayı dikkate alınmaz. Bu işlemler geri kalan tüm örnekler için tekrarlandığında hiçbir değer tek başına kural oluşturamadığı görülür.

Birinci aşama sonunda bir kural elde edilir ve kural ile 2, 7, 12 ve 13 nolu örnekler sınıflandırılır.

*İkinci Aşama:*  $n=n+1$  işlemi ile  $n=2$  olur ve ikinci aşamaya geçilir.

İkinci aşamada sınıflandırılmayan örneklerdeki karakteristik-değerlerin ikili kombinasyonları oluşturulur ve bu kombinasyonların kural oluşturup oluşturamayacaklarına bakılır.

Sınıflandırılmayan ilk örnek olan birinci örnek ele alınır. Bu örnek {Risk:Yüksek, Kârlılık:Önemli, Zaman:Uzun, Seviye:Yüksek} karakteristik-değerlerinden oluşmaktadır. Bu değerlerin ikili kombinasyonları aşağıda verilmiştir:

{Risk:Yüksek,Kârlılık:Önemli}, {Risk:Yüksek,Zaman:Uzun}, {Risk:Yüksek,Seviye:Yüksek}  
 {Kârlılık:Önemli, Zaman:Uzun}, {Kârlılık:Önemli, Seviye:Yüksek},  
 {Zaman:Uzun, Seviye:Yüksek}

Bu kombinasyonlar örnek setine sırasıyla uygulanır. İlk karakteristik-değer kombinasyonu olan {Risk:Yüksek, Kârlılık:Önemli} birden fazla sınıfta geçmektedir. Bundan dolayı kural oluşturamaz. İkinci kombinasyon {Risk:Yüksek,Zaman:Uzun} kombinasyonu sadece RED sınıfında geçmektedir. Bundan dolayı aşağıdaki kuralı oluşturur:

**Kural 2: EĞER Risk=Yüksek VE Zaman=Uzun İSE Karar=RED**

Bu kural ile 1, 2 ve 8 nolu örnekler sınıflandırılmaktadır. Bu örnekler daha sonra dikkate alınmamak üzere işaretlenirler. Geriye kalan {Risk:Yüksek,Seviye:Yüksek}, {Kârlılık:Önemli, Zaman:Uzun}, {Kârlılık:Önemli, Seviye:Yüksek}, {Zaman:Uzun, Seviye:Yüksek} kombinasyonları birden fazla sınıfa karşılık geldiklerinden dolayı kural oluşturamazlar.

Sınıflandırılmamış bir sonraki örnek olan dördüncü örnek seçilir. Bu örnek {Risk:Orta, Kârlılık:Orta, Zaman:Uzun, Seviye:Yüksek} karakteristik-değerlerinden

oluşmaktadır. Bu karakteristik değer çiftlerinin ikili kombinasyonları aşağıda verilmiştir:

{Risk:Orta, Kârlılık:Orta}, { Risk:Orta, Zaman:Uzun}, {Risk:Orta, Seviye:Yüksek}  
 {Kârlılık:Orta, Zaman:Uzun}, { Kârlılık:Orta, Seviye:Yüksek}  
 {Zaman:Uzun, Seviye:Yüksek}

Bu kombinasyonların sadece bir sınıfta geçip geçmedikleri sırasıyla kontrol edilir. Birinci ve ikinci kombinasyon birden fazla sınıfta geçtiklerinden dolayı kural oluşturamazlar. Üçüncü kombinasyon olan {Risk:Orta, Seviye:Yüksek} sadece bir sınıfta geçmektedir. Bu kombinasyon aşağıdaki kuralı oluşturur:

**Kural 3: EĞER Risk=Orta VE Seviye=Yüksek İSE Karar=KABUL**

Bu kural ile 4, 5 ve 10 nolu örnekler sınıflandırılır ve bu örnekler işaretlenir. İlk üç kural ile sınıflandırılan örnekler ile sınıflandırılmamış örnekler Tablo 6.2a'da verilmiştir.

Tablo 6.2a. İlk üç kural ile sınıflandırılan örnekler

| Kural No | Sınıflandırılan Örnekler | Sınıflandırılmamış Örnekler |
|----------|--------------------------|-----------------------------|
| 1        | 3 – 7 – 12 – 13          | 1-2-4-5-6-8-9-10-11-14      |
| 2        | 1 – 2 – 8                | 4-5-6-8-9-10-11-14          |
| 3        | 4 – 5 – 8                | 6-9-10-11-14                |

Sınıflandırılmamış ilk örnek olan altıncı örnek ele alınır. Bu örnek {Risk:Orta, Kârlılık:Küçük, Zaman:Kısa, Seviye:Düşük} karakteristik-değerlerinden oluşmaktadır. Bu değerlerin ikili kombinasyonları aşağıda verilmiştir:

{Risk:Orta, Kârlılık:Küçük}, {Risk:Orta, Zaman:Kısa}, {Risk:Orta, Seviye:Düşük}  
 {Kârlılık:Küçük, Zaman:Kısa}, {Kârlılık:Küçük, Seviye:Düşük}  
 {Zaman:Kısa, Seviye:Düşük}

Kombinasyonların tüm örnek setinde sadece bir sınıfta geçip geçmediklerine bakılır. Bu kombinasyonlardan sadece üçüncüsü {Risk:Orta, Seviye:Düşük} bir sınıfta geçtiğinden dolayı kural oluşturur. Elde edilen kural aşağıda verilmiştir:

**Kural 4: EĞER Risk=Orta VE Seviye=Düşük İSE Karar=RED**

Bu kural ile 6 ve 14 nolu örnekler sınıflandırılır ve bu örnekler bir daha dikkate alınmamak üzere işaretlenirler.

Bu aşamaya kadar sınıflandırılmamış olarak 9, 10 ve 11 nolu örnekler kalmaktadır. Bu örneklerden ilki olan dokuzuncu örnek {Risk:Yüksek, Kârlılık:Küçük, Zaman:Kısa, Seviye:Yüksek} karakteristik-değerlerinden oluşmaktadır. Bu değerlerin ikili kombinasyonları aşağıda verilmiştir:

{Risk:Yüksek,Kârlılık:Küçük}, {Risk:Yüksek,Zaman:Kısa}, {Risk:Yüksek,Seviye:Yüksek}  
 {Kârlılık:Küçük, Zaman:Kısa}, {Kârlılık:Küçük, Seviye:Yüksek}  
 {Zaman:Kısa, Seviye:Yüksek}

Bu kombinasyonlardan sadece {Zaman:Kısa, Seviye:Yüksek} tüm örnek setinde sadece bir sınıfta geçtiğinden dolayı kural oluşturur. Diğer kombinasyonlardan hiçbirisi sadece bir sınıfta geçmediklerinden dolayı kural oluşturamazlar. Elde edilen kural aşağıda verilmiştir:

**Kural 5: EĞER Zaman=Kısa VE Seviye=Yüksek İSE Karar=KABUL**

Bu kural ile 5, 9, 10 ve 13 nolu örnekler sınıflandırılır.

Bu aşamaya kadar sınıflandırılmamış örnek olarak 11 nolu örnek kalmaktadır. Bu örnek {Risk:Yüksek, Kârlılık:Orta, Zaman:Kısa, Seviye:Düşük} karakteristik-değerlerinden oluşmaktadır. Bu değerler ile elde edilen kombinasyonlar aşağıda verilmiştir:

{Risk:Yüksek, Kârlılık:Orta}, {Risk:Yüksek, Zaman:Kısa}, {Risk:Yüksek,Seviye:Düşük}  
 {Kârlılık:Orta, Zaman:Kısa}, {Kârlılık:Orta,Seviye:Düşük}, {Zaman:Kısa,Seviye:Düşük}

Bu kombinasyonlar örnek setine tek tek uygulandığında {Risk:Yüksek, Zaman:Kısa} kombinasyonunun sadece bir sınıfta geçtiği ve diğer kombinasyonların ise birden

fazla sınıfta geçtikleri görülmektedir. {Risk:Yüksek, Zaman:Kısa} kombinasyonu ile aşağıdaki kural elde edilir:

**Kural 6: EĞER Risk=Yüksek VE Zaman=Kısa İSE Karar=KABUL**

Bu kural yardımıyla 9 ve 11 nolu örnekler sınıflandırılır.

Bu aşamaya kadar altı kural üretildi ve bu kurallar ile tüm örnekler sınıflandırıldı. Hangi kural tarafından hangi örneklerin sınıflandırıldığı Tablo 6.2b’de verilmiştir.

Tablo 6.2b. Elde edilen kurallar ile sınıflandırılan örnekler

| Kural No | Sınıflandırılan Örnekler | Sınıflandırılmamış Örnekler              |
|----------|--------------------------|------------------------------------------|
| 1        | 3 – 7 – 12 – 13          | 1 – 2 – 4 – 5 – 6 – 8 – 9 – 10 – 11 – 14 |
| 2        | 1 – 2 – 8                | 4 – 5 – 6 – 8 – 9 – 10 – 11 – 14         |
| 3        | 4 – 5 – 8                | 6 – 9 – 10 – 11 – 14                     |
| 4        | 6 – 14                   | 9 – 10 – 11                              |
| 5        | 5 – 9 – 10 – 13          | 11                                       |
| 6        | 9 – 11                   | –                                        |

#### Adım-4:

Ancak 5 nolu kuralın sınıflandırdığı örnekler diğer kurallar tarafından da sınıflandırılmaktadır. 5 nolu kural 5, 9, 10 ve 13 nolu örnekleri sınıflandırmaktadır. Bu örneklerden 5 ve 10 nolu örnekler Kural-3, 9 nolu örnek Kural-6, 13 nolu örnek ise Kural-1 tarafından sınıflandırıldığından dolayı Kural-5 iptal edilir. Bu işlemler sonunda elde edilen kurallar seti Tablo 6.3’te verilmiştir.

Tablo 6.3. REX-3 ile Yatırım Projesi-2 kullanılarak elde edilen kurallar

| Kural No | Kural Tanımı                                           |
|----------|--------------------------------------------------------|
| 1        | <b>EĞER Risk=Düşük İSE Karar=KABUL</b>                 |
| 2        | <b>EĞER Risk=Yüksek VE Zaman=Uzun İSE Karar=RED</b>    |
| 3        | <b>EĞER Risk=Orta VE Seviye=Yüksek İSE Karar=KABUL</b> |
| 4        | <b>EĞER Risk=Orta VE Seviye=Düşük İSE Karar=RED</b>    |
| 5        | <b>EĞER Risk=Yüksek VE Zaman=Kısa İSE Karar=KABUL</b>  |

### 6.3. Diğer Algoritmalarla Karşılaştırma

REX-3 algoritması İLA ve C4.5 algoritmalarıyla ile Yatırım Projesi-2 örneği kullanılarak karşılaştırılacaktır. Ardından IRIS veri seti ve IRIS test seti kullanılarak

ID3 ve RULES ailesinin diğer algoritmaları olan Rules-3, Rules-3 Plus, Rules-4 ile karşılaştırılacaktır. Ardından sekiz adet veri seti kullanılarak Rules-3 algoritması ile karşılaştırma işlemi yapılacaktır.

### 6.3.1. ILA ve C4.5 ile karşılaştırma

Yatırım Projesi-2 örneğinden, ILA ve C4.5 algoritmaları kullanılarak kural çıkarılması durumunda REX-3 ile elde edilen kuralların aynılarının elde edildiği görülmüştür[117].

### 6.3.2. IRIS veri setini kullanarak karşılaştırma

BÖLÜM-4'te verilen IRIS veri seti kullanılarak REX-3 algoritması ile elde edilen kurallar Tablo 6.4'te verilmiştir. REX-3 ile elde kuralların sayısı ve kurallardaki toplam şart sayısı ile diğer algoritmalar tarafından elde edilen kurallar sayıları Tablo 6.5'te verilmiştir.

Tablo 6.4. REX-3 tarafından elde edilen kurallar (IRIS veri seti)

| Kural | Kural Tanımı                                                                |
|-------|-----------------------------------------------------------------------------|
| 1     | EĞER $1 \leq PL < 1.98$ İSE KARAR=Iris-setosa                               |
| 2     | EĞER $1.7 \leq PW < 2.1$ İSE KARAR=Iris-virginica                           |
| 3     | EĞER $0.9 \leq PW < 1.3$ İSE KARAR=Iris-versicolor                          |
| 4     | EĞER $2.1 \leq PW < 2.5$ İSE KARAR=Iris-virginica                           |
| 5     | EĞER $3.95 \leq PL < 4.93$ VE $1.3 \leq PW < 1.7$ İSE KARAR=Iris-versicolor |
| 6     | EĞER $2.8 \leq SW < 3.2$ VE $4.93 \leq PL < 5.91$ İSE KARAR=Iris-virginica  |
| 7     | EĞER $2.4 \leq SW < 2.8$ VE $1.3 \leq PW < 1.7$ İSE KARAR=Iris-versicolor   |

Tablo 6.5 REX-3 tarafından elde edilen kural ve kural başına şartların ortalaması (IRIS veri seti)

| Algoritma    | Kural Sayısı | Şart Sayısı | Kural Başına Ortalama Şart Sayısı |
|--------------|--------------|-------------|-----------------------------------|
| RULES-3      | 11           | 17          | 1.545                             |
| ID3          | 8            | 14          | 1.750                             |
| RULES-3 PLUS | 10           | 14          | 1.400                             |
| RULES-4      | 9            | 12          | 1.333                             |
| REX-1        | 8            | 11          | 1.375                             |
| REX-2        | 7            | 10          | 1.429                             |
| REX-3        | 7            | 10          | 1.429                             |

### 6.3.3. RULES-3 algoritması ile karşılaştırma

Endüktif öğrenme algoritmaları tarafından kullanılan bazı veri setleri kullanılarak RULES-3 ve REX-3 algoritmaları tarafından elde edilen kurallar Tablo 6.6'da verilmiştir. Kullanılan veri setleri ile ilgili detaylı bilgiler Bölüm-7'de verilmiştir.

Kullanılan sekiz veri setinden, Rules-3 sadece Monk1 ve Lenses veri setleri için REX-3 ile aynı sayıda kural üretmiştir. Diğer altı veri seti için REX-3 algoritması ile elde edilen kural sayıları daha düşüktür. Monk1, Monk2 ve Monk3 veri setleri için her iki algoritma tarafından üretilen kural sayılarının birbirine yakın çıkmasının nedeni, bu veri setlerindeki karakteristik sayılarının (6 adet) az olmasındandır. Karakteristik sayılarının çok fazla olduğu Vote, Promoter ve Parity5+5 örnek setleri için REX-3 algoritması çok daha az sayıda kural üretmiştir. Sekiz veri seti kullanılarak Rules-3 ile elde edilen toplam kural sayısı 259 iken, REX-3 ile elde edilen toplam kural sayısı 223'tür.

Tablo 6.6. REX-3 ve Rules-3 tarafından elde edilen kurallar

| Veritabanı<br>(Örnek Veri Seti) | Algoritma |       |
|---------------------------------|-----------|-------|
|                                 | RULES-3   | REX-3 |
| Vote                            | 36        | 25    |
| Monk1                           | 22        | 22    |
| Monk2                           | 83        | 79    |
| Monk3                           | 27        | 26    |
| Zoo                             | 12        | 9     |
| Lenses                          | 5         | 5     |
| Parity5+5                       | 47        | 40    |
| Promoter                        | 27        | 17    |

### 6.3.4. IRIS\_TEST veri setini kullanarak karşılaştırma

IRIS veri setinin son 80 örneği test verisi olarak kullanılmıştır. IRIS\_TEST veri seti Tablo 6.7'de verilmiştir. IRIS veri seti kullanılarak elde edilen kurallar test verilerine uygulanmış ve elde edilen sonuçlar Tablo 6.8'da gösterilmiştir [97, 98, 118].

Tablo 6.7. IRIS\_Test veri seti

| No | SL  | SW  | PL  | PW  | IRIS            |
|----|-----|-----|-----|-----|-----------------|
| 1  | 6.1 | 2.6 | 5.6 | 1.4 | Iris-virginica  |
| 2  | 5.5 | 2.5 | 4.0 | 1.3 | Iris-versicolor |
| 3  | 6.1 | 3.0 | 4.6 | 1.4 | Iris-versicolor |
| 4  | 4.8 | 3.0 | 1.4 | 0.3 | Iris-setosa     |
| 5  | 6.0 | 2.2 | 4.0 | 1.0 | Iris-versicolor |
| 6  | 4.9 | 2.5 | 4.5 | 1.7 | Iris-virginica  |
| 7  | 7.9 | 3.8 | 6.4 | 2.0 | Iris-virginica  |
| 8  | 5.5 | 2.4 | 3.8 | 1.1 | Iris-versicolor |
| 9  | 5.4 | 3.9 | 1.7 | 0.4 | Iris-setosa     |
| 10 | 5.9 | 3.0 | 4.2 | 1.5 | Iris-versicolor |
| 11 | 4.8 | 3.1 | 1.6 | 0.2 | Iris-setosa     |
| 12 | 5.2 | 2.7 | 3.9 | 1.4 | Iris-versicolor |
| 13 | 4.4 | 3.2 | 1.3 | 0.2 | Iris-setosa     |
| 14 | 6.2 | 3.4 | 5.4 | 2.3 | Iris-virginica  |
| 15 | 5.5 | 3.5 | 1.3 | 0.2 | Iris-setosa     |
| 16 | 5.1 | 3.7 | 1.5 | 0.4 | Iris-setosa     |
| 17 | 4.9 | 3.1 | 1.5 | 0.1 | Iris-setosa     |
| 18 | 4.7 | 3.2 | 1.3 | 0.2 | Iris-setosa     |
| 19 | 5.0 | 3.2 | 1.2 | 0.2 | Iris-setosa     |
| 20 | 5.6 | 2.9 | 3.6 | 1.3 | Iris-versicolor |
| 21 | 7.3 | 2.9 | 6.3 | 1.8 | Iris-virginica  |
| 22 | 5.7 | 4.4 | 1.5 | 0.4 | Iris-setosa     |
| 23 | 4.5 | 2.3 | 1.3 | 0.3 | Iris-setosa     |
| 24 | 6.5 | 2.8 | 4.6 | 1.5 | Iris-versicolor |
| 25 | 5.7 | 2.8 | 4.1 | 1.3 | Iris-versicolor |
| 26 | 4.9 | 3.1 | 1.5 | 0.1 | Iris-setosa     |
| 27 | 6.1 | 2.9 | 4.7 | 1.4 | Iris-versicolor |
| 28 | 6.3 | 3.4 | 5.6 | 2.4 | Iris-virginica  |
| 29 | 4.6 | 3.4 | 1.4 | 0.3 | Iris-setosa     |
| 30 | 4.9 | 3.1 | 1.5 | 0.1 | Iris-setosa     |
| 31 | 5.7 | 2.8 | 4.5 | 1.3 | Iris-versicolor |
| 32 | 6.5 | 3.0 | 5.2 | 2.0 | Iris-virginica  |
| 33 | 4.6 | 3.6 | 1.0 | 0.2 | Iris-setosa     |
| 34 | 5.7 | 2.9 | 4.2 | 1.3 | Iris-versicolor |
| 35 | 5.8 | 2.7 | 5.1 | 1.9 | Iris-virginica  |
| 36 | 5.0 | 3.5 | 1.6 | 0.6 | Iris-setosa     |
| 37 | 6.2 | 2.9 | 4.3 | 1.3 | Iris-versicolor |
| 38 | 5.6 | 2.5 | 3.9 | 1.1 | Iris-versicolor |
| 39 | 6.7 | 3.1 | 4.7 | 1.5 | Iris-versicolor |
| 40 | 5.1 | 3.8 | 1.6 | 0.2 | Iris-setosa     |
| 41 | 7.7 | 2.6 | 6.9 | 2.3 | Iris-virginica  |
| 42 | 4.9 | 3.0 | 1.4 | 0.2 | Iris-setosa     |
| 43 | 5.7 | 2.5 | 5.0 | 2.0 | Iris-virginica  |
| 44 | 5.2 | 3.5 | 1.5 | 0.2 | Iris-setosa     |
| 45 | 6.3 | 2.9 | 5.6 | 1.8 | Iris-virginica  |
| 46 | 5.2 | 4.1 | 1.5 | 0.1 | Iris-setosa     |
| 47 | 4.8 | 3.0 | 1.4 | 0.1 | Iris-setosa     |
| 48 | 7.4 | 2.8 | 6.1 | 1.9 | Iris-virginica  |
| 49 | 5.4 | 3.0 | 4.5 | 1.5 | Iris-versicolor |
| 50 | 4.3 | 3.0 | 1.1 | 0.1 | Iris-setosa     |
| 51 | 5.8 | 2.7 | 5.1 | 1.9 | Iris-virginica  |
| 52 | 5.1 | 3.8 | 1.5 | 0.3 | Iris-setosa     |
| 53 | 4.4 | 3.0 | 1.3 | 0.2 | Iris-setosa     |
| 54 | 6.7 | 3.0 | 5.0 | 1.7 | Iris-versicolor |
| 55 | 5.7 | 2.6 | 3.5 | 1.0 | Iris-versicolor |
| 56 | 6.7 | 3.1 | 5.6 | 2.4 | Iris-virginica  |
| 57 | 5.4 | 3.4 | 1.7 | 0.2 | Iris-setosa     |
| 58 | 6.8 | 3.0 | 5.5 | 2.1 | Iris-virginica  |
| 59 | 5.0 | 3.3 | 1.4 | 0.2 | Iris-setosa     |
| 60 | 6.4 | 2.8 | 5.6 | 2.2 | Iris-virginica  |
| 61 | 6.4 | 3.2 | 4.5 | 1.5 | Iris-versicolor |
| 62 | 4.8 | 3.4 | 1.6 | 0.2 | Iris-setosa     |
| 63 | 7.6 | 3.0 | 6.6 | 2.1 | Iris-virginica  |
| 64 | 5.9 | 3.0 | 5.1 | 1.8 | Iris-virginica  |
| 65 | 5.1 | 2.5 | 3.0 | 1.1 | Iris-versicolor |
| 66 | 5.6 | 3.0 | 4.5 | 1.5 | Iris-versicolor |
| 67 | 5.1 | 3.4 | 1.5 | 0.2 | Iris-setosa     |
| 68 | 5.6 | 3.0 | 4.1 | 1.3 | Iris-versicolor |
| 69 | 6.2 | 2.8 | 4.8 | 1.8 | Iris-virginica  |
| 70 | 5.1 | 3.5 | 1.4 | 0.2 | Iris-setosa     |
| 71 | 5.0 | 2.3 | 3.3 | 1.0 | Iris-versicolor |
| 72 | 5.8 | 2.7 | 3.9 | 1.2 | Iris-versicolor |
| 73 | 5.6 | 2.7 | 4.2 | 1.3 | Iris-versicolor |
| 74 | 6.5 | 3.0 | 5.8 | 2.2 | Iris-virginica  |
| 75 | 6.0 | 2.2 | 5.0 | 1.5 | Iris-virginica  |
| 76 | 6.4 | 2.8 | 5.6 | 2.1 | Iris-virginica  |
| 77 | 6.0 | 3.0 | 4.8 | 1.8 | Iris-virginica  |
| 78 | 6.4 | 3.1 | 5.5 | 1.8 | Iris-virginica  |
| 79 | 6.1 | 2.8 | 4.7 | 1.2 | Iris-versicolor |
| 80 | 5.9 | 3.2 | 4.8 | 1.8 | Iris-versicolor |

SL : Sepal Length

SW : Sepal Width

PL : Petal Length

PW : Petal Width

Tablo 6.8. IRIS test veri seti kullanılarak elde edilen doğruluk oranları

| Algoritma    | Kural Sayısı | Doğruluk Oranı |
|--------------|--------------|----------------|
| RULES-3      | 11           | %91.50         |
| ID3          | 8            | %92.50         |
| RULES-3 PLUS | 10           | %97.50         |
| RULES-4      | 9            | %93.60         |
| REX-1        | 8            | %93.75         |
| REX-2        | 7            | %100           |
| REX-3        | 7            | %100           |

IRIS test veri seti kullanılarak Tablo 6.8'daki sonuçlara bakıldığında REX-3 algoritması tarafından elde edilen sonuçların ID3 ve RULES ailesi (Rules-3, Rules-3 Plus, Rules-4) algoritmalarından daha iyi olduğu görülmektedir. Hem kural sayısı hem de doğruluk oranı bakımından çok daha iyi sonuçlar elde edilmiştir.

#### 6.4. Sonuçlar

REX-1 ve REX-2 algoritmaları ile REX-3 algoritması tarafından elde edilen kurallar karşılaştırıldıklarında REX-3 algoritması ile elde edilen kuralların daha az olduğu görülmüştür. REX-3 algoritması RULES ailesi (Rules-3, Rules-3 Plus, Rules-4) algoritmalarından daha az sayıda ve daha yüksek doğruluk oranında kurallar üretebilmektedir. Özellikle karakteristik sayısının fazla olduğu örnekler için çok daha az sayıda kural ürettiği görülmüştür.

Endüktif öğrenme algoritmaları ile ilgili gerçek hayattan alınan veri setleri kullanılarak yapılan performans karşılaştırmaları Bölüm-7'de verilmiştir.

## **BÖLÜM 7. SONUÇLAR**

### **7.1. Giriş**

Geliştirilen REX-1, REX-2 ve REX-3 algoritmaları sırasıyla Bölüm-4, Bölüm-5 ve Bölüm-6'da detaylı olarak anlatılmıştır. Her algoritmanın kural çıkarma prosedürü için ayrı ayrı örnekler kullanıldı. Her bölümde bazı örnek veri setleri kullanılarak diğer algoritmalarla testler yapıldı. Bu bölümde endüktif öğrenme ile ilgili performans karşılaştırmalarında en çok başvurulmuş veri setleri kullanılarak diğer algoritmalarla performans karşılaştırmaları yapılacaktır. Örnek setleri 127, 128 ve 129. kaynaklarda belirtilen internet ftp/web adreslerinden elde edilebilir.

### **7.2. Diğer Algoritmalarla Performans Karşılaştırma**

Performans karşılaştırmaları için en yaygın kullanılan algoritmalar ve toplam 11 adet veri seti seçilmiştir. Bu veri setleri günlük hayattan alınan gerçek veri setleridir. Bu veri setlerinin kural çıkarma (eğitim seti) ve çıkarılan kuralları test etmek (test seti) için iki farklı seti bulunmaktadır. Bu veri setleri ile ilgili bilgiler Tablo 7.1 ve Tablo 7.2'de verilmiştir.

Tablo 7.1. ve 7.2'de veriler kullanılarak, her veri seti için her algoritma tarafından elde edilen doğruluk oranları Tablo 7.3'te verilmiştir. Elde edilen bu değerler,  $\pm 0.1$  doğruluk oranlarına göre sıralandıklarında en yüksek doğruluk oranını sağlayan algoritma olarak REX-1 algoritması görülmektedir. Tablo 7.3'teki koyu (bold) yazılar en yüksek değerleri göstermektedir. Buna göre; C4.5Rules algoritması 3, C4.5Prune algoritması 2, ID3 algoritması 3, ILA algoritması 3, ILA-2 algoritması 4, OC1 algoritması 2, CN2 algoritması 3, REX-1 algoritması 6, REX-2 algoritması 4 ve REX-3 algoritması 3 örnekte en yüksek doğruluk oranlarında kural üretmiştir.

Tablo 7.1. Eğitime veri setlerine ait özellikler

| Veritabanı<br>(Örnek Seti) | Örnek<br>Sayısı | Karakteristik<br>Sayısı | Sınıflar    | Sınıflar     |           |
|----------------------------|-----------------|-------------------------|-------------|--------------|-----------|
|                            |                 |                         |             | Örnek Sayısı | Oranı (%) |
| Vote                       | 300             | 16                      | Republican  | 116          | 38.7      |
|                            |                 |                         | Democrat    | 184          | 61.3      |
| Monk1                      | 124             | 6                       | 0           | 62           | 50.0      |
|                            |                 |                         | 1           | 62           | 50.0      |
| Monk2                      | 169             | 6                       | 0           | 105          | 62.8      |
|                            |                 |                         | 1           | 64           | 37.2      |
| Monk3                      | 122             | 6                       | 0           | 62           | 50.8      |
|                            |                 |                         | 1           | 60           | 49.2      |
| Zoo                        | 67              | 16                      | 1           | 29           | 43.3      |
|                            |                 |                         | 2           | 14           | 20.9      |
|                            |                 |                         | 3           | 1            | 1.5       |
|                            |                 |                         | 4           | 9            | 13.4      |
|                            |                 |                         | 5           | 1            | 1.5       |
|                            |                 |                         | 6           | 6            | 9.0       |
|                            |                 |                         | 7           | 7            | 10.4      |
| Lenses                     | 16              | 4                       | 1           | 2            | 12.5      |
|                            |                 |                         | 2           | 2            | 12.5      |
|                            |                 |                         | 3           | 12           | 75.0      |
| Parity5+5                  | 100             | 10                      | 0           | 45           | 45.0      |
|                            |                 |                         | 1           | 55           | 55.0      |
| Tic-Tac-Toe                | 638             | 9                       | Pozitif     | 423          | 66.3      |
|                            |                 |                         | Negatif     | 215          | 33.7      |
| Splice                     | 700             | 60                      | EI          | 174          | 24.9      |
|                            |                 |                         | IE          | 159          | 22.7      |
|                            |                 |                         | None        | 367          | 52.4      |
| Promoter                   | 106             | 57                      | Promoter    | 53           | 50.0      |
|                            |                 |                         | Other       | 53           | 50.0      |
| Iris                       | 70              | 4                       | Iris-Versi. | 23           | 32.9      |
|                            |                 |                         | Setosa      | 21           | 30.0      |
|                            |                 |                         | Iris-Virg.  | 26           | 37.1      |

Tablo 7.2. Test veri setlerine ait özellikler

| Veritabanı<br>(Örnek Seti) | Örnek<br>Sayısı | Karakteristik<br>Sayısı | Sınıflar    | Test verisindeki<br>sınıf oranı (%) |
|----------------------------|-----------------|-------------------------|-------------|-------------------------------------|
| Vote                       | 135             | 16                      | Republican  | 38.7                                |
|                            |                 |                         | Democrat    | 61.3                                |
| Monky1                     | 432             | 6                       | 0           | 50.0                                |
|                            |                 |                         | 1           | 50.0                                |
| Monky2                     | 432             | 6                       | 0           | 62.8                                |
|                            |                 |                         | 1           | 37.2                                |
| Monky3                     | 432             | 6                       | 0           | 52.8                                |
|                            |                 |                         | 1           | 47.2                                |
| Zoo                        | 34              | 16                      | 1           | 43.3                                |
|                            |                 |                         | 2           | 20.9                                |
|                            |                 |                         | 3           | 1.5                                 |
|                            |                 |                         | 4           | 13.4                                |
|                            |                 |                         | 5           | 1.5                                 |
|                            |                 |                         | 6           | 9.0                                 |
|                            |                 |                         | 7           | 10.4                                |
|                            |                 |                         | 1           | 12.5                                |
| Lenses                     | 8               | 4                       | 2           | 12.5                                |
|                            |                 |                         | 3           | 75.0                                |
| Parity5+5                  | 1024            | 10                      | 0           | 50.0                                |
|                            |                 |                         | 1           | 50.0                                |
| Tic-Tac-Toe                | 320             | 9                       | Pozitif     | 63.4                                |
|                            |                 |                         | Negatif     | 36.6                                |
| Splice                     | 3170            | 60                      | EI          | 24.9                                |
|                            |                 |                         | IE          | 22.7                                |
|                            |                 |                         | None        | 52.4                                |
| Promoter                   | 40              | 57                      | Promoter    | 57.0                                |
|                            |                 |                         | Other       | 43.0                                |
| Iris                       | 80              | 4                       | Iris-Versi. | 33.8                                |
|                            |                 |                         | Setosa      | 36.2                                |
|                            |                 |                         | Iris-Virg.  | 30.0                                |

Tablo 7.3. Algoritmalar tarafından elde edilen doğruluk oranları [121, 123, 124, 125, 126]

| Veritabanı<br>(Örnek Seti) | C4.5<br>Rules | C4.5<br>Prune | ID3         | ILA          | ILA-2        | OC1          | CN2          | REX-1        | REX-2        | REX-3        |
|----------------------------|---------------|---------------|-------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| Vote                       | 95.6          | <b>97.0</b>   | 94.1        | 94.8         | <b>97.0</b>  | <b>96.3</b>  | 95.6         | <b>97.0</b>  | 89.6         | 89.6         |
| Parity5+5                  | 50.0          | 50.0          | 50.8        | 51.2         | 50.0         | 52.4         | 53.0         | <b>64.7</b>  | <b>63.8</b>  | <b>64.7</b>  |
| Monk1                      | 93.5          | 75.7          | 81.0        | <b>100</b>   | <b>100</b>   | 91.2         | 98.6         | 97.2         | 97.2         | 96.8         |
| Monk2                      | 66.2          | 65.0          | 69.9        | 78.5         | 59.7         | <b>96.3</b>  | 75.4         | 76.4         | 72.2         | 76.9         |
| Monk3                      | 96.3          | 97.2          | 91.7        | 88.2         | <b>100</b>   | 94.2         | 90.7         | 93.5         | 89.4         | 91.0         |
| Zoo                        | 85.3          | 85.3          | <b>97.1</b> | 91.2         | 88.2         | 73.5         | 82.4         | <b>97.1</b>  | 85.3         | 88.2         |
| Lenses                     | <b>62.5</b>   | <b>62.5</b>   | <b>62.5</b> | 50.0         | <b>62.5</b>  | 37.5         | <b>62.5</b>  | <b>62.5</b>  | <b>62.5</b>  | <b>62.5</b>  |
| Tic-Tac-Toe                | <b>98.1</b>   | 82.2          | 80.9        | <b>98.1</b>  | 84.1         | 85.6         | <b>98.0</b>  | <b>97.1</b>  | <b>97.1</b>  | <b>98.0</b>  |
| Splice                     | <b>92.7</b>   | 90.4          | 89.0        | 67.9         | 73.4         | 91.2         | 84.5         | 83.7         | 83.2         | 89.00        |
| Promoter                   | 97.5          | 95.0          | <b>100</b>  | <b>100</b>   | 97.5         | 87.5         | <b>100</b>   | <b>100</b>   | <b>100</b>   | 97.50        |
| <i>Ortalama</i>            | <i>83.77</i>  | <i>80.03</i>  | <i>81.7</i> | <i>81.99</i> | <i>81.24</i> | <i>80.57</i> | <i>84.07</i> | <i>86.92</i> | <i>84.03</i> | <i>85.42</i> |

Tablo 7.3'teki bütün örnek setlerinin ortalama doğruluk oranlarına göre en yüksek doğruluk oranı %86.92 ile REX-1, ikinci en yüksek doğruluk oranı %85.42 ile REX-3, üçüncü en yüksek doğruluk oranı %84.07 ile CN2 ve dördüncü en yüksek doğruluk oranı ise %84.03 ile REX-2 algoritmasına aittir.

### 7.3. BALLOONS ve BALANCE Veri Setlerini Kullanarak Performans Karşılaştırma

Balloons ve Balance veri setleri kullanılarak elde edilen sonuçlar ID3, AQ, ILA ve C4.5 algoritmaları ile karşılaştırılacaktır. Normal Balance veri seti dört karakteristik, üç sınıf ve toplam 625 örnekten oluşan bir veri setidir. Bu veri setinin küçültülmüş şekli ise toplam 416 örnekten oluşmaktadır. Karşılaştırma işleminin yapılacağı diğer algoritmalar da küçük veri setini kullandıklarından dolayı burada da bu set seçilmiştir. Bu setteki sınıflar, örnek sayıları ve oranları aşağıda verilmiştir:

B : 194 ( % 46.64 )  
L : 187 ( % 44.95 )  
R : 35 ( % 8.41 )

Balloons veri seti ise 4 karakteristik, iki sınıf ve toplam 16 örnekten oluşmaktadır. Örneklerin dağılımı birinci sınıf için %25 ve diğer sınıf için ise %75 oranındadır.

ID3, AQ, ILA ve C4.5 algoritmaları ile REX-1, REX-2 ve REX-3 algoritmaları tarafından elde edilen değerler Tablo 7.4'te verilmiştir [123].

Tablo 7.4. Balloons ve Balance veri setleri kullanılarak elde edilen sonuçlar

| Veri Seti | Algoritma | Kural Sayısı          | Kural Başına<br>Ortalama Şart Sayısı |
|-----------|-----------|-----------------------|--------------------------------------|
| Balloons  | ID3       | 3                     | 1.67                                 |
|           | AQ        | 3                     | 1.33                                 |
|           | ILA       | 3                     | 1.33                                 |
|           | C4.5      | 3 (Hata oranı:%0.0)   | 1.33                                 |
|           | REX-1     | 3                     | 1.33                                 |
|           | REX-2     | 3                     | 1.33                                 |
|           | REX-3     | 3                     | 1.33                                 |
| Balance   | ID3       | 410                   | 3.85                                 |
|           | AQ        | 312                   | 3.53                                 |
|           | ILA       | 303                   | 3.41                                 |
|           | C4.5      | 26 (Hata oranı:%16.8) | 2.07                                 |
|           | REX-1     | 217                   | 3.41                                 |
|           | REX-2     | 217                   | 3.41                                 |
|           | REX-3     | 217                   | 3.41                                 |

Balloons veri seti için ID3 dışındaki diğer algoritmalar aynı sayıda kurallar üretmişlerdir. Balance veri seti kullanılarak elde edilen sonuçlara göre geliştirilen her üç algoritmanın sonuçları ID3, AQ ve ILA algoritmalarına göre daha iyidir. C4.5Rules algoritması, Balance veri seti için %16.8 hata oranına göre 26 kural üretmiştir. Ancak diğer algoritmalarda hata oranı sıfır olup, örnek setinde verilen tüm örnekler için kural çıkarılmıştır.

#### 7.4. MONK Problemlerini Kullanarak Performans Karşılaştırma

Monk1, Monk2 ve Monk3 veri setleri (Bkz. Tablo 7.1, 7.2) kullanılarak ID3, C4.5, C4.5Rules, C4.5Prune ile yeni geliştirilen üç algoritma tarafından üretilen kurallar Tablo 7.5'te gösterilmiştir[121]. Yeni geliştirilen REX-1, REX-2 ve REX-3

algoritmaları kullanılarak elde edilen kural sayısı ve şart sayısı ile Tablo 7.6'da verilen doğruluk oranlarına göre yapılan karşılaştırmalar aşağıda verilmiştir:

Tablo 7.5 Monk problemleri için elde edilen kural ve şart sayıları [121, 122, 126]

| Algoritma  | Monk1        |             | Monk2        |             | Monk3        |             |
|------------|--------------|-------------|--------------|-------------|--------------|-------------|
|            | Kural Sayısı | Şart Sayısı | Kural Sayısı | Şart Sayısı | Kural Sayısı | Şart Sayısı |
| ID3        | 53           | 216         | 105          | 498         | 30           | 98          |
| C4.5       | 60           | 262         | 113          | 566         | 27           | 89          |
| C4.5 Prune | 9            | 31          | 55           | 353         | 20           | 102         |
| C4.5 Rules | 31           | 101         | 97           | 374         | 23           | 65          |
| REX-1      | 21           | 58          | 78           | 332         | 26           | 73          |
| REX-2      | 21           | 58          | 83           | 350         | 24           | 71          |
| REX-3      | 22           | 61          | 79           | 339         | 26           | 81          |

- Monk1 problemi için Tablo 7.5'deki bilgilere göre en az şartla en az kuralı üreten algoritmanın *C4.5Prune* olduğu görülmektedir. Ancak Tablo 7.6'daki doğruluk oranlarına bakıldığında bu algoritmanın doğruluk oranının %75.7 olduğu ve bu oranın diğer algoritmalarla göre daha düşük olduğu görülmektedir. Monk1 problemi için en yüksek doğruluk oranı değerleri yeni geliştirilen her üç algoritmaya aittir. Bu algoritmaların kural sayıları ve şart sayıları ise *C4.5prune* dışındaki diğer algoritmalarla çok daha düşüktür.
- Monk2 problemi için Tablo 7.5'deki bilgilere göre en az kural üreten *C4.5Prune* algoritmasıdır. Bu algoritma 353 adet şart kullanarak 55 adet kural üretmektedir. Oysa *REX-1* algoritması 332 şart ile 78, *REX-2* algoritması 350 şart ile 83 ve *REX-3* algoritması ise 339 şart ile 79 adet kural üretmiştir. Bu bilgiler *C4.5Prune* algoritması tarafından elde edilen kurallardaki ortalama şart sayısının 6.42 iken, bu oran *REX-1* için 4.26, *REX-2* için 4.22 ve *REX-3* için ise 4.29'dur. Bu *C4.5Prune* algoritması ile elde edilen kurallardaki şart sayısının yüksek olduğunu göstermektedir. Ayrıca Tablo 7.6'da *C4.5prune* %65.0 doğruluk oranına sahip iken, bu oran *REX-1* için %76.2, *REX-2* için %75.7 ve *REX-3* için ise %76.9'dur.

- Monk3 problemine göre en az kuralı üreten algoritma *C4.5Prune* iken, bu algoritmanın 20 kuralda kullandığı şart sayısı 120'dir. İkinci iyi sonuç ise *C4.5Rules* algoritması tarafından elde edilmiştir. Bu algoritma ile elde edilen kural sayısı 23 ve şart sayısı ise sadece 65'tir. Doğruluk oranı da göz önüne alındığında en iyi sonucu veren algoritma olarak *C4.5Rules* seçilebilir.

Tablo 7.6. Test veri setleri kullanılarak elde edilen doğruluk oranları [121, 122, 126]

| Algoritma  | Monk1        | Monk2        | Monk3        |
|------------|--------------|--------------|--------------|
| ID3        | %81.0        | %69.9        | %91.7        |
| C4.5       | %82.4        | %69.7        | %90.3        |
| C4.5 Prune | %75.7        | %65.0        | <b>%97.2</b> |
| C4.5 Rules | %93.5        | %66.2        | <b>%96.3</b> |
| REX-1      | <b>%97.2</b> | <b>%76.4</b> | %93.5        |
| REX-2      | <b>%97.2</b> | <b>%72.2</b> | %89.4        |
| REX-3      | <b>%96.8</b> | <b>%76.9</b> | %91.0        |

Endüktif öğrenme algoritmaları arasında en iyi algoritmalar olarak C4.5 algoritmaları kabul edildiğinden, bu sonuçlara göre yeni geliştirilen algoritmalar ile elde edilen sonuçların çok iyi olduğu rahatlıkla söylenebilir. Çünkü yeni geliştirilen bu algoritmalar verilen eğitim setinden daha az sayıda kurallar üretmekte ve test verilerine göre yüksek doğruluk oranında sonuç vermektedirler.

### 7.5. J-Pruned ile Performans Karşılaştırma

J-Pruned algoritması 2002 yılında, C4.5 algoritmasının kullandığı ağaç budama metoduna alternatif olarak geliştirilmiştir[109]. REX-1, REX-2 ve REX-3 algoritmaları ile *J-Pruned* algoritmasının performanslarını karşılaştırmak için Monk1, Monk2, Monk3, Lens24 (Lenses) ve Iris veri setleri kullanılacaktır. Kullanılan veri setleri ile ilgili bilgiler Tablo 7.1 ve Tablo 7.2'de verilmiştir. Algoritmalar ile yapılan karşılaştırmaların sonuçları Tablo 7.7'de verilmiştir.

Tablo 7.7'deki sonuçlara göre J-Pruned algoritması Monk1, Monk2 ve Monk3 veri setleri için daha az sayıda kural üretmesine rağmen bu kuralların test verilerini

sınıflandırma oranları bir örnek hariç (REX-2 algoritması Monk3 için %89.4) daha düşüktür. Ayrıca J-Pruned algoritması eğitim setini %100 doğruluk oranında sınıflandırmaz. Lenses ve Iris veri setleri için REX-1, REX-2 ve REX-3 algoritmaları hem daha az sayıda kural üretmiş hem de test verilerini daha yüksek doğruluk oranında sınıflandırmıştır. Sadece REX-1 algoritması Iris veri seti için daha düşük oranda sınıflandırma yapmıştır.

Tablo 7.7. *J-Pruned* ile performans karşılaştırma [109]

| Veri Seti | J-Pruned     |              | REX-1        |              | REX-2        |              | REX-3        |              |
|-----------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
|           | Kural Sayısı | Doğruluk (%) | Kural Sayısı | Doğruluk (%) | Kural Sayısı | Doğruluk (%) | Kural Sayısı | Doğruluk (%) |
| Monk1     | 15           | 67.8         | 21           | 97.2         | 21           | 97.2         | 22           | 96.8         |
| Monk2     | 22           | 55.7         | 78           | 76.4         | 83           | 72.2         | 79           | 76.9         |
| Monk3     | 13           | 90.9         | 26           | 93.5         | 24           | 89.4         | 26           | 91.0         |
| Lenses    | 7            | 70.0         | 5            | 100          | 5            | 100          | 5            | 100          |
| Iris      | 9            | 95.3         | 8            | 93.8         | 7            | 100          | 7            | 100          |

## 7.6. TDIDT ve PRISM Algoritmaları ile Performans Karşılaştırma

Geliştirilen yeni algoritmalar TDIDT ve PRISM algoritmaları ile test edilecektir. Bu amaçla Monk1, Monk2, Monk3 ve Soybean veri setleri ile bunlara ait test veri setleri kullanılacaktır. Gerçek hayattan alınan Monk veri setlerine ait özellikler Tablo 7.1 ve 7.2’de verilmiştir. Soybean [127, 128, 129] veri seti ise 683 örnek, 35 karakteristik ve 19 sınıftan oluşmaktadır. Monk1, Monk2, Monk3 ve Soybean veri setleri kullanılarak TDIDT, PRISM, REX-1, REX-2 ve REX-3 algoritmaları tarafından elde edilen bilgiler Tablo 7.8’de gösterilmiştir.

Kural başına ortalama şart sayısı, toplam şart sayısının kural sayısına bölümünden elde edilmektedir. Bu bilgilere göre TDIDT algoritmasının ürettiği kural sayılarının diğer algoritmalarla göre daha fazla olduğu açıkça görülmektedir. Algoritmalar tarafından üretilen toplam kural sayıları; TDIDT:270, PRISM:231, REX-1:223, REX-2:226 ve REX-3:218 şeklindedir.

Algoritmaları karşılaştırmak için kullanılan bir diğer ölçü ise test veri setlerini kullanmaktır. Test veri setleri, üretilen kuralları kullanarak bu verilerin hangi

doğruluk oranına göre sağlandığını tespit etmek için kullanılır. Yani, bilinmeyen bir örnek karşısında algoritmanın hangi doğruluk oranına göre sonuç üreteceğini test etmektir. Burada kullanılan test setleri orjinal test setlerinden elde edilmiştir. Algoritmalar tarafından elde edilen doğruluk oranları Tablo 7.9’da verilmiştir [119].

Tablo 7.8. Algoritmalar tarafından elde edilen sonuçlar

| Algoritma |                      | MONK1 | MONK2 | MONK3 | SOYBEAN |
|-----------|----------------------|-------|-------|-------|---------|
| TDIDT     | Kural Sayısı         | 46    | 87    | 28    | 109     |
|           | Ortalama Şart Sayısı | 4.04  | 4.74  | 3.36  | 5.45    |
| PRISM     | Kural Sayısı         | 25    | 73    | 26    | 107     |
|           | Ortalama Şart Sayısı | 3.00  | 4.00  | 2.81  | 3.57    |
| REX-1     | Kural Sayısı         | 21    | 78    | 26    | 98      |
|           | Ortalama Şart Sayısı | 2.76  | 4.25  | 2.81  | 4.13    |
| REX-2     | Kural Sayısı         | 21    | 83    | 24    | 98      |
|           | Ortalama Şart Sayısı | 2.76  | 4.25  | 2.96  | 4.13    |
| REX-3     | Kural Sayısı         | 22    | 79    | 26    | 91      |
|           | Ortalama Şart Sayısı | 2.77  | 4.25  | 2.81  | 4.16    |

Not : TDIDT ve PRISM algoritmalarından elde edilen bilgiler INDUCED[130] programı kullanılarak elde edilmiştir.

Tablo 7.9. Test setlerini kullanarak doğruluk oranlarının karşılaştırılması

| Veri Seti | Test Setindeki Örnek Sayısı | Doğruluk Oranları (%) |       |              |              |              |
|-----------|-----------------------------|-----------------------|-------|--------------|--------------|--------------|
|           |                             | TDIDT                 | PRISM | REX-1        | REX-2        | REX-3        |
| Monk1     | 36                          | 75.00                 | 77.78 | <b>100</b>   | <b>100</b>   | <b>100</b>   |
| Monk2     | 52                          | 46.15                 | 53.85 | 73.1         | <b>78.8</b>  | <b>78.8</b>  |
| Monk3     | 36                          | <b>91.67</b>          | 83.33 | 83.33        | 83.33        | 83.33        |
| Soybean   | 204                         | 85.78                 | 84.80 | <b>97.06</b> | <b>97.06</b> | <b>97.37</b> |

Tablo 7.8’deki kural sayılarına bakıldığında TDIDT dışındaki diğer dört algoritmanın birbirlerine yakın sonuçlar ürettikleri görülmektedir. Ancak Tablo 7.9’da, test veri setleri kullanılarak elde edilen sonuçlar incelendiğinde yeni

geliştirilen algoritmaların doğruluk oranlarının çok yüksek olduğu görülmektedir. Bunun temel sebeplerinden biri karakteristik seçiminde entropi ve bilgi kazançları değerlerine göre bir seçimin yapılmasındandır. REX-3 algoritmasında ise bütün kombinasyonların denenmesinden ve en iyi kombinasyonun seçilmesinden kaynaklanmaktadır.

## 7.7. Geliştirilen Endüktif Öğrenme Programı (REX123)

REX-1, REX-2 ve REX-3 algoritmaları için bir program hazırlanmıştır. Program, Delphi-6 programlama dili kullanılarak görsel ortamda geliştirilmiştir. Program ile seçilen örnek setindeki verilerden istenen algoritma ile kural çıkarmak mümkündür. Elde edilen kurallar text ortamda bir dosyada saklanabilmekte ve varsa test veri seti kullanılarak doğruluk testi yapılabilmektedir.

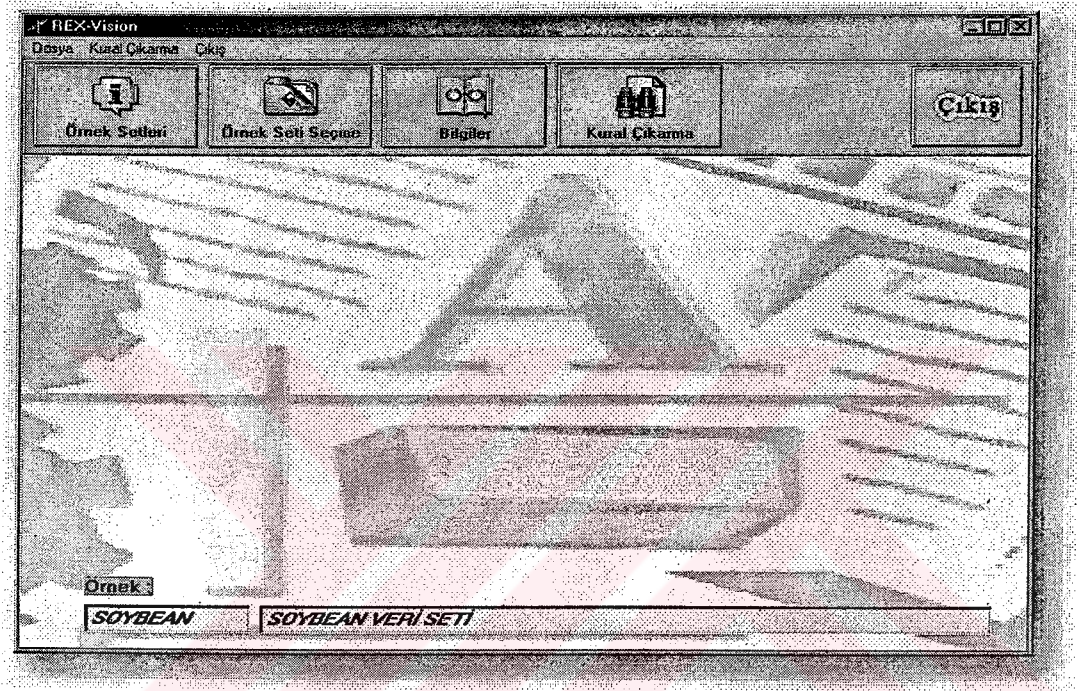
### 7.7.1. Programın özellikleri

- Program Delphi 6 programlama dili kullanılarak hazırlanmıştır.
- Veri setlerini tutmak için Paradox-7 veritabanı kullanılmıştır.
- Programa veriler, text olarak hazırlanmış bir dosyadan aktarılabilceği gibi kullanıcı tarafından da tek tek girilebilir. Kullanıcının bilgiler üzerinde değişiklik yapma imkanı vardır.
- Program aktarılabilecek veri setlerindeki karakteristik sayısı 2 – 250 arasında olabilir.
- Veri setindeki ilk alan birincil anahtar (primary index) olarak alınır.
- Veriler sayısal, alfabetik veya tarih formatında olabilir.
- Programa aktarılabilecek veri setlerinin sayısı için herhangi bir kısıtlama yoktur.
- Veri setinde bulunacak örnek sayısı istenen sayıda olabilir.
- Çıkarılan kural setleri text dosya olarak saklanabilir.
- Çıkarılan kural setlerini kullanarak, test veri setlerini hangi doğruluk oranlarında sınıflandırdığı bulunabilir.
- Sayısal karakteristikler için aralık tanımlamalarının yapılmasına izin verir.

- Seçilen örnek setinden REX-1, REX-2 ve REX-3 algoritmaları kullanılarak kural çıkarma işlemi aynı ekran üzerinden yapılabilir.

### 7.7.2. REX123 Programı

Programın ana menüsü Şekil 7.1’de gösterilmiştir.



Şekil 7.1. REX123 programının ana menüsü

Şekil 7.1’de gösterilen ana menü kullanılarak tüm işlemler yapılabilir. Buradan yapılabilecek işlemler aşağıda sıralanmıştır:

|                         |                                                                                                                                                        |
|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Örnek Setleri</b>    | Yeni bir örnek için veritabanının oluşturulmasını sağlar.                                                                                              |
| <b>Örnek Seti Seçme</b> | Mevcut örnek setlerinden birinin seçilmesini sağlar. Aktif olan örnek seti üzerinde işlemler yapılır.                                                  |
| <b>Bilgiler</b>         | Örnek setindeki bilgilerin görüntülenmesini/değiştirilmesini sağlar. Örnek setine bilgi ekleme ve silme işlemleri bu bölümden yapılabilir (Şekil 7.2). |
| <b>Kural Çıkarma</b>    | Kural çıkarma ekranının görüntülenmesini sağlar (Şekil 7.3).                                                                                           |

Örnek Seti

| Num | HAVA     | SICAKLIK | NEM    | RUZGAR   | KARAR  |
|-----|----------|----------|--------|----------|--------|
| 1   | GÜNEŞLİ  | SICAK    | YÜKSEK | HAFİF    | OYNAMA |
| 2   | GÜNEŞLİ  | SICAK    | YÜKSEK | KUVVETLİ | OYNAMA |
| 3   | BULUTLU  | SICAK    | YÜKSEK | HAFİF    | OYNA   |
| 4   | YAĞMURLU | ORTA     | YÜKSEK | HAFİF    | OYNA   |
| 5   | YAĞMURLU | SOĞUK    | NORMAL | HAFİF    | OYNA   |
| 6   | YAĞMURLU | SOĞUK    | NORMAL | KUVVETLİ | OYNAMA |
| 7   | BULUTLU  | SOĞUK    | NORMAL | KUVVETLİ | OYNA   |
| 8   | GÜNEŞLİ  | ORTA     | YÜKSEK | HAFİF    | OYNAMA |
| 9   | GÜNEŞLİ  | SOĞUK    | NORMAL | HAFİF    | OYNA   |
| 10  | YAĞMURLU | ORTA     | NORMAL | HAFİF    | OYNA   |
| 11  | GÜNEŞLİ  | ORTA     | NORMAL | KUVVETLİ | OYNA   |
| 12  | BULUTLU  | ORTA     | YÜKSEK | KUVVETLİ | OYNA   |
| 13  | BULUTLU  | SICAK    | NORMAL | HAFİF    | OYNA   |
| 14  | YAĞMURLU | ORTA     | YÜKSEK | KUVVETLİ | OYNAMA |

Text Olarak Kaydet Kapat

Şekil 7.2. Bilgiler ekranı

KURAL ÇIKARMA EKRANI [GOLF OYNAMA]

| Num | HAVA     | SICAKLIK | NEM    | RUZGAR   | KARAR  |
|-----|----------|----------|--------|----------|--------|
| 1   | GÜNEŞLİ  | SICAK    | YÜKSEK | HAFİF    | OYNAMA |
| 2   | GÜNEŞLİ  | SICAK    | YÜKSEK | KUVVETLİ | OYNAMA |
| 3   | BULUTLU  | SICAK    | YÜKSEK | HAFİF    | OYNA   |
| 4   | YAĞMURLU | ORTA     | YÜKSEK | HAFİF    | OYNA   |
| 5   | YAĞMURLU | SOĞUK    | NORMAL | HAFİF    | OYNA   |
| 6   | YAĞMURLU | SOĞUK    | NORMAL | KUVVETLİ | OYNAMA |
| 7   | BULUTLU  | SOĞUK    | NORMAL | KUVVETLİ | OYNA   |
| 8   | GÜNEŞLİ  | ORTA     | YÜKSEK | HAFİF    | OYNAMA |
| 9   | GÜNEŞLİ  | SOĞUK    | NORMAL | HAFİF    | OYNA   |
| 10  | YAĞMURLU | ORTA     | NORMAL | HAFİF    | OYNA   |
| 11  | GÜNEŞLİ  | ORTA     | NORMAL | KUVVETLİ | OYNA   |

Algoritma

REX-1 Algoritması

REX-2 Algoritması

REX-3 Algoritması

Seğir

Test

Kaydet

Kapat

ÜRETİLEN KURALLAR

Kural 1: EĞER NEM-YÜKSEK VE HAVA-GÜNEŞLİ İSE KARAR-OYNAMA [3]  
 Kural 2: EĞER HAVA-BULUTLU İSE KARAR-OYNA [4]  
 Kural 3: EĞER RUZGAR-HAFİF VE HAVA-YAĞMURLU İSE KARAR-OYNA [3]  
 Kural 4: EĞER RUZGAR-KUVVETLİ VE HAVA-YAĞMURLU İSE KARAR-OYNAMA [2]  
 Kural 5: EĞER NEM-NORMAL VE HAVA-GÜNEŞLİ İSE KARAR-OYNA [2]  
 ...  
 Toplam Kural Sayısı :5  
 Toplam Şart Sayısı :9  
 Kural Başına Şart Ortalaması :1.8  
 ...  
 Sınıflandırılmayan Örnek Sayısı:0

| Karakteristik Değer | Entropi  | D.Entropi |
|---------------------|----------|-----------|
| NEM                 | YÜKSEK   | 0.985     |
| NEM                 | NORMAL   | 0.592     |
| NEM                 | > > >    | 0.788     |
| RUZGAR              | HAFİF    | 0.811     |
| RUZGAR              | KUVVETLİ | 1.000     |
| RUZGAR              | > > >    | 0.892     |
| HAVA                | GÜNEŞLİ  | 0.971     |
| HAVA                | BULUTLU  | 0.000     |
| HAVA                | YAĞMURLU | 0.971     |

Örnek Sayısı :14

| Karakteristik | Değer    | Sayı |
|---------------|----------|------|
| NEM           | YÜKSEK   | 7    |
| NEM           | NORMAL   | 7    |
| RUZGAR        | HAFİF    | 8    |
| RUZGAR        | KUVVETLİ | 6    |

Şekil 7.3. Kural çıkarma ekranı

Kural çıkarma ekranından, istenen algoritmaya göre kural çıkarma işlemi gerçekleştirilir. Seçilen örnek seti ile ilgili bilgiler (karakteristikler, değerler ve değer sayıları) görülebilir. Ayrıca, örnek setindeki her karakteristik ve değere ait entropi, düzeltilmiş entropi ve bilgi kazanç oranları görüntülenir. Kural çıkarma ekranında, üretilen kurallar text dosya olarak saklanabilir. 'Test' düğmesi ile seçilen örnek setine ait test veri seti kullanılarak test işlemi yapılabilir. Test işlemleri için test veri setinin 'Örnek Adı'+ '\_TEST'.TXT ismi ile kaydedilmiş olması gerekir.



## BÖLÜM 8. TARTIŞMA ve ÖNERİLER

Bilgisayarla öğrenme, kendi kendini geliştiren süreçler üzerine odaklanmış olan yapay zekanın bir alanıdır. Bilgisayarla öğrenmenin temel amacı bilgi kazanımını otomatik hale getirmektir. Bilgi tabanlı sistemleri hazırlamak hem çok pahalı hem de uzman personelin çalışmasını gerektirir. Bilgi transfer sürecinin bir makine tarafından yapılması, yeni uzman sistemlerin maliyetini düşürmeyi ve bilginin elde edilmesini hızlandırır.

Bu tez çalışmasında, Bilgisayarla Öğrenme'nin bir alt dalı olan Endüktif Öğrenmede otomatik bilgi kazanımı için üç yeni algoritma geliştirilmiştir.

Algoritmaların başarısını test etmek için en son geliştirilen ve en çok kullanılan algoritmalar seçilmiştir. Bunlar arasında ID3, C4.5, C4.5Rules, C4.5Prune, GDT-RS, GDT-NR, ILA, ILA-2, Rules-3, Rules-3 Plus, Rules-4, OC1, CN2, AQ ve J-Pruned algoritmaları sayılabilir.

Performans karşılaştırmaları için gerçek hayattan alınan ve bu alanda en çok kullanılan örnek veri setleri seçilmiştir. Bunlar arasında IRIS, Vote, Lenses, Tic-Tac-Toe, Zoo, Parity5+5, Splice, Promoter, Monk1, Monk2, Monk3 ve Balance veri setleri sayılabilir. Bu veri setleri farklı boyutlarda ve farklı özelliklere sahiptirler. Örneğin, Tic-Tac-Toe veri seti 638 adet örnek ve 9 karakteristikten, Splice veri seti ise 700 adet örnek ve 60 adet karakteristikten oluşmaktadır.

Geliştirilen her üç algoritma için Delphi programlama dili kullanılarak programlar hazırlanmıştır. Delphi programlama dilinin seçilmesinin en önemli sebepleri olarak; hızlı işlem yapması ve görsel bir dil olmasındandır. Bütün örnek veri setleri için sonuçlar, ilgili programlar kullanılarak elde edilmiştir.

Elde edilen sonuçlara ve yapılan performans karşılaştırmalarına göre geliştirilen her üç algoritmanın sonuçları çok iyi derecededir. Sonuçların bu derece iyi olmasının sebepleri olarak, karakteristiklere ait entropi ve bilgi kazançlarının hesaplanarak bilgi değeri daha yüksek olan karakteristiklere öncelik verilmesi (REX-1 ve REX-2) ve en çok örnek sınıflandıran (REX-3) kuralların seçilmesi sayılabilir.

Bundan sonra Endüktif Öğrenme alanında değişik çalışmalar yapılabilir:

- Geliştirilecek algoritmalar sayısal verileri işleyebilmelidir. Geçmiş yıllarda yapılan çalışmalara bakıldığında çoğu algoritma nominal olarak kabul edilen alfabetik bilgilere göre kural çıkarmaktadır. Ancak mühendislik verilerinin büyük bir kısmı sayısal değerler içermektedir. Geliştirilecek algoritmalar sayısal veriler için aralık tanımlamalarına izin vermelidir. Bu aralık tanımlamalarının sınıflandırma tekniklerine göre yapılması algoritmanın gücünü artıracaktır. Aralık tanımlamaları hem otomatik hem de isteğe göre yapılabilirdir.
- Karakteristik seçimi için bilgi kazancına (entropi) alternatif olarak Genetik Algoritmalar kullanılabilir.
- Daha genel kuralları seçmek için özel kuralları eleyecek bir algoritma geliştirilebilir. Özel kurallar çok az sayıda (örnek setinin büyüklüğüne bağlı olarak) örneği sınıflandıran kurallardır. Bu kuralların elenmesi daha az sayıda kural elde edilmesini sağlayacaktır.
- Fuzzy, endüktif öğrenme metodunu kullanarak Bulanık Mantık tarafından kullanılabilecek kurallar üretebilen algoritmalar geliştirilebilir.
- Endüktif Öğrenme algoritmalarının karmaşıklıkları ile ilgili bir çalışma yapılabilir.

## KAYNAKLAR

- [1] FEIGENBAUM, E. A., "Expert System in the 1980s", in Infotech State of the Art Report on Machine Intelligence, Ed:A. Bond, Maidenhead, Pergamon, Infotch, 1981.
- [2] MICHIE, D., "Current Development in Expert System", in Applications of Expert Systems, Ed:J.R. Quinlan, Addison-Wesley, 1987.
- [3] WANG, X., "Inductive Learning Algorithms", Ph.D. Thesis, University of Wales Cardiff, 1997.
- [4] MICHALSKI, R.S., KODRATOFF, Y., "Research in machine learning: recent progress, classification of methods, and future directions", Machine Learning Vol.3, Morgan Kaufmann, San Mateo, CA, 3-30, 1990.
- [5] FORSYTH, R., "Machine Learning - Principle and Technology", Chapman and Hall, London, 1989.
- [6] ARCISZEWSKI, T., "Learning engineering: The key to automatic knowledge acquisition, Artificial Intelligence for Engineering Design, Analysis and Manufacturing", Cambridge University Press, 1996.
- [7] ATTAR, M.E., HAMERY, X., "Industrial expert system acquired by machine learning", Applied Artificial Intelligence, Taylor&France, 1994.

[8] MICHALSKI, R.S., "Synthesis of optimal and quasi-optimal variable-valued logic formulas", Proceeding of the 1975 Int. Symposium on Multiple-Valued Logic, Bloomington, Indiana, 76-87, 1975.

[9] MICHALSKI, R.S., "A theory and methodology of inductive learning", Machine Learning - An Artificial Intelligence Approach, Michalski; R.S., Carbonell, J.G. and Mitchell, T.M. (Eds), Morgan Kaufmann, Los Altos, CA, 83-134, 1983.

[10] CLARK P., NIBLETT T., "The CN2 inductive algorithm", Machine Learning, Vol 3, 261-283, 1989.

[11] CLARK, P., BOSWELL, R., "Rule induction with CN2: Some recent improvement", Machine Learning -Proceedings of the Fifth European Conference (EWSL-91), Kodratoff, Y (Ed), Berlin, Springer-Verlag, 151-163, 1991.

[12] MICHALSKI, R.S., MOZETIC, I., HONG, J., LAVRAC, N., "The multi-purpose incremental learning system AQ15 and its testing application in three medical domains", Proceedings of the Fifth National Conference on Artificial Intelligence, Philadelphia, PA, 1041-1047, 1986.

[13] QUINLAN, J.R., "Learning efficient classification procedures and their application to chess end games", Machine Learning - An Artificial Intelligence Approach, Eds: Michalski; R.S., Carbonell, J.G. and Mitchell, T.M. (Eds), Tioga Publishing Co, Palo Alto. CA, 463-482, 1983.

[14] CHENG, J., FAYYAD, U.M., IRANI, K.B., QIAN, Z., "Improved decision trees: A generalized version of ID3", Proceedings of the Fifth international conference on Machine Learning, Ann Arbor, Michigan, 100-106, 1988.

[15] UTGOFF, P.E., "ID5:An incremental ID3", Proceedings of the Fifth International Conference on Machine Learning, The University of Michigan, 107-20, 1988.

- [16] CENDROWSKA, J., "PRISM: An algorithm for inducing modular rules", Knowledge-Based Systems Vol 1: Knowledge Acquisition for Knowledge-Based Systems, Gaines, B.R. and Boose, J.H. (Eds), Academic Press, London, 255-276, 1988.
- [17] QUINLAN, J.R., "C4.5: Programs for Machine Learning", Morgan Kaufmann, San Mateo, CA, 1993.
- [18] BREIMAN, L., FRIEDMAN, J.H., OLSHEN, R. A., STONE, C. J., "Classification and Regression Trees", Wadsworth International Group, Belmont, California, 1984.
- [19] PAWLAK, Z., "Rough sets", International Journal of Computer and Information Sciences, 11(5), 341-356, 1982.
- [20] BERAN, R.J., "Minimum Hellinger distance for parametric models", Statistics, Vol. 5, 445-463, 1977.
- [21] CHEESEMAN, P., "AutoClass: A Bayesian classification system", Proceedings of the Fifth International Workshop on Machine Learning, Morgan Kaufmann, San Mateo, CA, 54-64, 1988.
- [22] KING, R.D., HENERY, R., FENG, C., SUTHERLAND, A., "A comparative study of classification algorithms: statistical, machine learning and neural network", Machine Learning 13: Machine Intelligence and Inductive Learning, Furukawa, K., Michie, D. and Muggleton, S. (Eds), Clarendon Press, Oxford, 311-359, 1994.
- [23] FUJIHARA, H., SIMMONS, D.B., ELLIS, N.C., SHANNON, R.E., "Knowledge conceptualization tool", IEEE Transactions on Knowledge and Data Engineering, Vol 9, No 2, 1997.
- [24] KODRATOFF, Y., "Recent advances in machine learning", International Journal of Pattern Recognition and Artificial Intelligence, 469-511, 1992.

- [25] DILMAN, I., "Induction and Deduction", Basil Blackwell and Mott, Oxford, 1973.
- [26] KODRATOFF, Y., "Induction and the organization of knowledge", Machine Learning - A Multistrategy Approach, Vol.4, Michalski, R., Tecuci, G. (Eds), Morgan Kaufmann, San Mateo, CA, 85-106, 1994.
- [27] LANGLEY, P., "Elements of machine learning", Morgan Kaufmann, CA, 1996.
- [28] MICHALSKI, R.S., STEPP, R.E., "Learning from observation: conceptual clustering", Machine Learning-An Artificial Intelligence Approach, MICHALSKI, R.S. CARBONELL, J.G., MITCHELL, T.M. (Eds), Morgan Kaufmann, Los Altos, CA, 331-363, 1983.
- [29] KODRATOFF Y., GANASCIA, J.G., "Improving the generalization step in learning", Machine Learning - An Artificial Intelligence Approach, Vol.2, Michalski, R.S., Carbonell, J.G. and Mitchell, J.M. (Eds), Morgan Kaufmann, Los Altos, CA, 240-244, 1986.
- [30] KODRATOFF Y., "Machine learning", Knowledge Engineering Vol.1: Fundamentals, Adeli, H. (Ed), McGraw-Hill, New York, 1990.
- [31] WITTEN I.H., MACDONALD, B.A., "Using concept learning for knowledge acquisition", Machine Learning and Uncertain Reasoning--Knowledge Based Systems Vol.3, Gaines, B. and Boose, J. (Eds), Academic Press, London, 139-164, 1990
- [32] VAPNIK V.N., CHERVONENKIS A.Y., "On the uniform convergence of relative frequencies of events to their probabilities", Theory of Probability and Its Applications, 16(2), 264-280, 1971.

[33] VALIANT, L.G., "A theory of the learnable", *Communications of the ACM*, Vol.27, 1134-1142; or, *Readings in Machine Learning*, (1990) Shavlik, J.W. and Dietterich, T.G. (Eds), Morgan Kaufmann, San Mateo, CA, 192-200, 1984.

[34] LINIAL, N., MANSOUR, Y., RIVEST, R.L., "Result on learnability and the Vapnik-Chervonenkis dimension", *29th Symposium on the Foundations of Computer Science*, White Plains, NY, 120-128, 1988.

[35] HAUSSLER, D. "Quantifying inductive bias: AI learning algorithms and Valiant's learning framework", *Artificial Intelligence*, 36, 177-221; reproduced in: *Readings in Machine learning*, (1990) Shavlik, J.W. and Dietterich, T.G., Morgan Kaufmann, San Mateo, CA, 205-227, 1988.

[36] SCHAPIRE, R.E., "The strength of weak learnability", *30th Annual Symposium on Foundations of Computer Science*, Research Triangle Park, NC, IEEE Computer Soc. Press, Los Alamitos, CA, 28-33, 1989.

[37] BLUMER, A., EHRENFEUCHT, A., HAUSSLER, D., WARMUTH, M.K., "Learnability and the Vapnik-Chervonenkis dimension", *Journal of Association for Computing Machinery*, 36(4), 929-965; reproduced in: *Machine Learning*, 8(1), 1992, 45-73, 1989.

[38] SHALIN, V. L., WISNIEWSKI, E.J., LEVI, K.R., SCOTT, P.D., "A formal analysis of learning system for knowledge acquisition", *Machine Learning and Uncertain Reasoning-Knowledge Based Systems Vol.3*, Gaines.B. and Boose. J. (Eds), Academic Press, London, 75-92, 1990.

[39] ALBERT, M.K., AHA, D.W., "Analyses of instance-based learning algorithms", *AAAI-91 Proceedings of the 9th National Conference on Artificial Intelligence Vol.2*, AAAI Press, Menlo Park, CA, 553-558, 1991.

[40] OBLOW, E.M., "Implementing Valiant's learnability theory using random sets", *Machine Learning*, 8(1), Kluwer Academic Publishers, Boston, 45-73, 1992.

- [41] KRZYSZTOF J. C., "An Algorithm Which Learns Multiple Covers Via Integer Linear Programming Part I: the CLILP2 algorithm", *Kybernetes*, Vol. 24 No. 2, pp. 29-50, MCB University Press, 0368-492X, 1995.
- [42] MITCHELL, T.M., "The need for biases in learning generalizations", *Readings in Machine Learning*, Shavlik, J.W. and Dietterich, T.G. (Eds), Morgan Kaufmann, San Mateo, CA, 184-191, 1990.
- [43] DIETTERICH, T.G., "Limitations on inductive learning", *Proceedings of the 6th International Workshop on Machine Learning (89 ML)*, Ithaca, NY. and Segre, A.M. (Ed), Morgan Kaufmann, San Mateo, CA, 124-128, 1989.
- [44] BOLC, L., CYTOWSKI, J., "Search Methods for Artificial Intelligence", Academic Press, London, 1992.
- [45] THORNTON, C.J., BOULAY, B., *Artificial Intelligence through Search*, Intellect, Oxford, 267-290, 1992.
- [46] BLUMER, A., EHRENFEUCHT, A., HAUSSLER, D., WARMUTH, M.K., "Occam's razor", *Information Processing Letters* 24, 377-380; reproduced in: *Readings in Machine Learning*, (1987) Shavlik, J.W. and Dietterich, T.G. (Eds), Morgan Kaufmann, San Mateo, CA, 201-204, 1990.
- [47] COHEN, P.R., Feigenbaum, E.A., "The Handbook of Artificial Intelligence", Vol. 3, Pitman, London, 385-400, 1982.
- [48] AKSOY, M. S., ÇAĞIL, G., TÜRKER, A. K., "Number-plate recognition using inductive learning", *Robotics and Autonomous Systems*, 33, 149-153, 2000.
- [49] QUINLAN, J.R., "Induction of decision trees", *Machine Learning Vol.1*, Kluwer Academic Publishers, Boston, 81-106; reproduced in: *Readings in Machine*

learning, (1990) Shavlik, J.W. and Dietterich, T.G. (Eds), Morgan Kaufmann, San Mateo, CA, 57-66, 1986.

[50] SCHLIMMER, J.C., FISHER, D.H., "A case study of incremental concept induction", AAAI 86 - Proceedings of the 5th National Conference on Artificial Intelligence, Philadelphia, PA, 496-501, 1986.

[51] UTGOFF, P.E., "Incremental induction of decision trees", Machine Learning Vol.4, 161-186, 1989.

[52] CRAWFORD, S.L., "Extensions to the CART algorithm", Machine Learning and Uncertain Reasoning - Knowledge-Based Systems, Vol. 3, Gaines, B and Boose, J. (Eds), Academic Press, London, 15-35, 1990.

[53] MURTHY, S.K., KASIF, S., SALZBERG, S., "A system for induction of oblique decision trees", Journal of Artificial Intelligence Research, 2, AI Access Foundation and Morgan Kaufmann, San Mateo, CA, 1-32, 1994.

[54] BOO, S. K., DEOK, H. C., SANG, C. P., "Intelligent process control in manufacturing industry with sequential processes", Int. J. Production Economics, 60-61, p: 583-590, 1999.

[55] SABRINO, M. D. C., BRAVO, L. J. B., "Knowledge acquisition from batch semiconductor manufacturing data", Intelligent Data Analysis, vol:3, Issue 5, Pages 399-408, November 1999.

[56] BOO, S. K., SANG, C. P., "Integrated machine learning approaches for complementing statistical process control procedures", Decision Support Systems - 29, 59-72, 2000.

[57] JEREZ-ARAGONE, J. M., GOMEZ-RUIZ, J. A., RAMOS-JIMENEZ, G., MUNOZ-PEREZ, J. ALBA-CONEJO, E., "A combined neural network and decision

trees model for prognosis of breast cancer relapse”, *Artificial Intelligence in Medicine* 27, 45–63, 2003.

[58] LASHKIA, G.V., ANTONY L., “An inductive learning method for medical diagnosis”, *Pattern Recognition Letters* - 24, 273 –282, 2003.

[59] AKSOY, M.S., “New Algorithms for Machine Learning”, Thesis of PhD, University of Walles, Cardiff, United Kingdom, 1993.

[60] LUZEAUX, D., “Process control and machine learning: rule-based incremental control”, *IEEE Transiblic Autom. Control* 39 (1994) 1166–1171.

[61] KOSTOV, A., ANDREWS, B., STEIN, R.B., POPOVIC, D., W.W. ARMSTRONG, “Machine learning in control of functional electrical stimulation systems for locomotion”, *IEEE Trans. Biomed. Eng.* 44 (1995) 541–551.

[62] PHAM, D.T., AKSOY, M.S., “A new algorithm for inductive learning”, *Journal of System Engineering* 5, 115–122, 1995.

[63] SKIBNIEWSKI, M., ARCISZEWSKI, T., LUEPRASERT, K., “Constructability analysis: machine learning approach”, *ASCE J. Comput. Civil Eng.* 12 (1997) 8–16.

[64] MROZEK, A., “Rough set and dependency analysis among attributes in computer implementations of expert’s inference models”, *International Journal of Man-machines Studies*, 30, 457-473, 1989.

[65] PAWLAK, Z., “Rough classification of patients after highly selective vagotomy for duodenal ulcer”, *International Journal Man-Machine Studies*, 24, 413-433, 1986.

[66] PHAM, D. T., PHAM, P. T. N., “Artificial intelligence in engineering”, *International Journal of Machine Tools & Manufacture* 39, 937–949, 1999.

- [67] RUSSELL, S.J., NORVIG, P., "Artificial Intelligence: A Modern Approach", Prentice-Hall, Englewood Cliffs, NJ, 1995.
- [68] BRATKO, I., "Knowledge-based problem-solving in AL3", Machine Intelligence 10, Michie, D. and Pao, Y-H. (Eds), Ellis Horwood, 63-85, 1982.
- [69] PEREZ, R.A., HALL, L.O., ROMANIUK, S., LILKENDEY, J.T., "Inductive learning for expert systems in manufacturing", Proceedings of the 25th International Conference on System Science, Milutinovic, V., Shriver, B.D. Nunamaker, J.F., Jr., and Sprague, R.H., Jr., (Eds), IEEE Computer Soc. Press, Los Alamitos, CA, 14-25, 1992.
- [70] PIASTA, Z., "Diagnostic classification for Brittle Matrix composites assisted by pattern recognition and rough sets analysis", Proceedings of the Third Intl. Symposium on Brittle Matrix Composites (BMC 3) Bradt, A.M. and Marshall, I.H. (Eds), Staszic Palace, Warsaw, Poland, 258-268, 1991.
- [71] WHITEHALL, B.L., LU, Y. STEPP, R. E., "CAQ: A machine learning tool for engineering", Artificial Intelligence in Engineering, 5(4), 189-198, 1990.
- [72] WARREN, LIAO, T. , ZHAN, Z. H., MOUNT, C. R., "An integrated database and expert system for failure mechanism identification:Part I-automated knowledge acquisition", Engineering Failure Analysis, Volume 6, Issue 6, Pages 387-406, December 1999.
- [73] DOLSAK, B., MUGGLETON, S., "The application of inductive logic programming to finite element mesh design", Inductive Logic Programming, Muggleton, S. (Ed), Academic Press, London, 453-472, 1992.
- [74] PACHOWICZ, P.W., "Low-level numerical characteristics and inductive learning methodology in texture recognition", (1st) IEEE Int. Workshop on tools for Artificial Intelligence (TAI 89), Fairfax, VA, 91-98, 1989.

- [75] BATUR, C., SRINIVASAN, A., CHAN, C.C., "Automated rule-based model generation for uncertain complex dynamic systems", *Engineering Applications of Artificial Intelligence* 4 (5), Pergamon Press, Oxford, 359-366, 1991.
- [76] SRINIVASAN, A., BATUR, C., CHAN C. C., "Using inductive learning to determine fuzzy rules for dynamic systems", *Engineering Applications of Artificial Intelligence*, 6(3), 257-264, 1993.
- [77] GIORDANA, A., SAITTA, L., BERGADANO, F., BRANCADORI, F., MARCHI, D.D., "ENIGMA: A system that learns diagnostic knowledge", *IEEE Transactions on Knowledge and Data Engineering*, 5(1), 15-28, 1993.
- [78] WARREN, LIAO T. , ZHAN, Z. H., MOUNT, C. R., "An integrated database and expert system for failure mechanism identification: Part II-the system and performance testing", *Engineering Failure Analysis*, Volume 6, Issue 6, 407-421, December 1999.
- [79] SMYTH, P., MELLSTROM, J., "Detecting novel classes with applications to fault diagnosis", *Machine Learning: Proceedings of the Ninth Intl. Workshop (ML92)*, Sleeman, D. and Edwards, P. (Eds), M. Kaufmann, CA, Morgan Kaufmann, San Mateo, CA, 416-425, 1992.
- [80] HASHEMI-NASSAB, R., JELOVSEK, F.R., "Inductive learning from examples: a rough sets approach", *1991 Symposium on Applied Computing*, IEEE Comput. Soc. Press, CA, 346-349, 1991.
- [81] QUINLAN, J.R., "Learning efficient classification procedures and their applications to chess end games", in R.S. Michalski, J.G. Carbonell, T.M. Mitchell (Eds.), *Machine Learning, An Artificial Intelligence Approach*, Morgan Kaufmann, Palo Alto, CA, 1983. pp. 463-482, 1995.

[82] MICHALSKI, R.S., "A theory and methodology of inductive learning", in J.W. Shavlik, T.G. Dietterich (Eds.), Kaufmann, Readings in Machine Learning, San Mateo, CA, 70-95, 1990.

[83] MUGGLETON, S., "Inductive Logic Programming". Academic Press, London, 565 p, 1992.

[84] WANG, L.X., MENDEL, M., "Generating fuzzy rules by learning from examples", IEEE Trans. Syst., Man Cybernet. 22, 1414-1427, 1992.

[85] TÜRKYILMAZ, A., ŞEVKLİ, M, AKSOY, M.S., "Inspection of ceramic tiles using inductive learning", 2nd International Conference on Responsive Manufacturing in Gaziantep, Turkey, 508-513, June 2002.

[86] HAYKIN, S., "Neural networks a comprehensive foundation", Prentice Hall, Upper Saddle River, New Jersey, 446-452, 1994.

[87] ESTABROOKS, A., "A combination scheme for inductive learning from imbalanced data sets", Master tezi, Dalhousie University – Daltech, Halifax, Nova Scotia, 2000.

[88] NAKAKUKI, Y., KOSEKI, Y., TANAKA, M., "Inductive learning in probabilistic domain", in proc. Eighth National Conf. On AI, Boston, 809-814, July 29-August 3, 1990.

[89] FORSYTH, R., "Machine learning principles and techniques", Ed:R.Forsyth, Chapman and Hall, London, 1989.

[90] HANCOX, P.J., MILLS, W.J. REID, B.J., "Artificial Intelligence / Expert Systems", Ergosyst Associates, Lawrence, Kansas, 1990.

[91] AL-ATTAR, A., "Rule induction from mythology to methodoloji", Research and developments in expert systems VIII, London, 85-103, September-1991.

- [92] QUINLAN, J.R., "Generating production rules from decision trees", in proc. Tenth IJCAI-87, Milan, Italy, 304-307, 1987.
- [93] QUINLAN, J.R., "Learning logical definitions from relations", Machine learning, 5, Kluwer Publishers, Boston, 239-266, 1990.
- [94] HUNT, E. B., MARIN, J., STONE, P. J., "Experiments in induction", Academic Press, New York, 1966.
- [95] PHAM, D. T., AKSOY, M.S., "An algorithm for automatic rule induction", Artificial Intel. Eng., 8, 277-282, 1993.
- [96] PHAM D. T., DIMOV S. S., "An efficient algorithm for automatic knowledge acquisition", Pattern Recognition Vol: 30, No:7 1137-1143, 1997.
- [97] PHAM, D. T; DIMOV, S.S., "An Algorithm for Incremental Inductive Learning". Proc. Instn. Mech. Engrs, vol. 211, part B, pp. 239 – 249, 1997.
- [98] PHAM, D. T., DIMOV, S. S., "The RULES-4 incremental inductive learning algorithm", Applications of Artificial Intelligence in Engineering XII, R.A. Adey G. Rzevski and R. Teti (Eds) Computational Mechanics Publications Southampton Boston , 163-166, 1997.
- [99] KAUFMAN, K.A., MICHALSKI, R.S., "An Adjustable Rule Learner for Pattern Discovery Using the AQ Methodology", Journal of Intelligent Information Systems, 14, pp. 199-216, 2000.
- [100] WNEK, J., KAUFMAN, K., BLOEDORN, E., MICHALSKI, R.S., Selective Inductive Learning System AQ15c: The Method And User's Guide. Center for Machine Learning and Inference, George Mason University, Fairfax, VA, 1995.

- [101] PACHOWICZ, P.W., BALA, J., "Improving recognition effectiveness of noisy texture concept through optimization of their description", In proc. 8th Int. Workshop on Machine Learning, Evanston, p:625-629, 1991.
- [102] MICHALSKI, R.S., "A theory and methodology of inductive learning", Machine Learning - An Artificial Intelligence Approach, Michalski; R.S., Carbonell, J.G. and Mitchell, T.M. (Eds), Morgan Kaufmann, Los Altos, CA, 83-134, 1983.
- [103] WNEK, J., MICHALSKI, R.S., "Hypothesis-Driven Constructive Induction in AQ17: A Method and Experiments", Reports of the Machine Learning and Inference Laboratory, MLI 91-4, School of Information Technology and Engineering, George Mason University, Fairfax, VA, May 1991.
- [104] BLOEDORN, E., MICHALSKI, R.S., "Data Driven Constructive Induction in AQ17-PRE: A Method and Experiments," Proceedings of the Third International Conference on Tools for AI, San Jose, CA, November 9-14, 1991.
- [105] KAUFMAN, K.A., MICHALSKI, R.S., "Learning in an Inconsistent World: Rule Selection in AQ18", Reports of the Machine Learning and Inference Laboratory, MLI 99-2, George Mason University, Fairfax, VA, 1999.
- [106] KAUFMAN, K.A., MICHALSKI, R.S., "The AQ18 System for Machine Learning: User's Guide," Reports of the Machine Learning and Inference Laboratory, MLI 00-3, George Mason University, Fairfax, VA, 2000.
- [107] CERVONE, G., PANAIT, L. A., MICHALSKI, R.S., "The Development of the AQ20 Learning System and Initial Experiments", Proceedings of the International Conference on Intelligent Systems (IIS 2000), Poland, July 2001.
- [108] QUINLAN, J.R., "The Minimum Description Length Principle and Categorical Theories", Proceedings of the 11th International Conference on Machine Learning, pp. 233-241, 1994.

- [109] BRAMER, M.A., "Using J-pruned to reduce overfitting in classification trees", Knowledge-Based Systems, Vol:15, 301-308, 2002.
- [110] NAZAR, K., BRAMER, M.A., "Concept dispersion, feature interaction and their effect on particular sources of bias in machine learning", Research and Development in Expert Systems XIV, SGES Publications, 1997.
- [111] NAZAR, K., BRAMER, M.A., "Estimating concept difficulty with cross-entropy", Knowledge Discovery and Data Mining, 1999.
- [112] FISHER, R.A., "The use of multiple measurements in taxonomic problems", Ann of Eugenics, 7, 466-475, 1936.
- [113] PAWLAK, Z., "Rough Sets", Theoretical Aspects of Reasoning about Data, Kluwer Academic Publishers, Dordrecht, 1991.
- [114] ZHONG, N., DONG, J., OHSUGA, S., "Rule discovery by soft induction techniques", Neurocomputing 36, P: 171-204, 2001.
- [115] KLINKENBERG, R., "Rule Set Quality Measures For Inductive Learning Algorithms", Master Thesis, University Of Missouri – Rolla, 1996.
- [116] SHAO, X., ZHANG, G., LI, P., CHEN, Y., "Application of ID3 algorithm in knowledge acquisition for tolerans design", Journal of Material Processing Technology, vol:117, p:66-74, 2001.
- [117] OPREA, M. M., "Inductive learning applied to knowledge acquisition for an expert systems", 35th Years of Petroleum-Gas University Activity, 27-29 November, 2002.
- [118] PHAM, D. T., DIMOV S. S., Salem Z., "Technique for Selecting Examples in Inductive Learning", ESIT 2000, Aachen, Germany, 14-15 September 2000.

- [119] BRAMER, M.A., "Automatic Induction of Classification Rules from Examples Using N-Prism", Research and Development in Intelligent Systems XVI. Springer-Verlag, pp. 99-121, 2000.
- [120] CENDROWSKA, J., "Knowledge Acquisition for Expert Systems: Inducing Modular Rules from Examples", PhD Thesis, The Open University, 1990.
- [121] WU, X., "Rule Induction with Extension Matrices", Journal of the American Society for Information Science, Volume 49, Number 5, pp 435-454, 1998.
- [122] HAMILTON, H. J., SHAN, N., CERCONI, N., "RIAC: A Rule Induction Algorithm Based on Approximate Classification Technical Report", Department of Computer Science University of Regina Regina, Saskatchewan, CANADA, S4S 0A2, CS-96-06, ISSN 0828-3494 ISBN 0-7731-0321-X, May 1996.
- [123] TOLUN, M. R., ABU-SOUD S.M., "ILA: An Inductive Learning Algorithm For Rule Extraction", Expert Systems With Applications, Vol: 14, p:361-370, 1998.
- [124] TOLUN, M. R., SEVER, H., ULUDAĞ, M., "Improved Rule Discovery Performance on Uncertainty", The Second Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD-98), Melbourne, Australia, 15-17 April 1998.
- [125] AN, A., "Learning Classification Rules From Data", Computers and Mathematics with Applications 0, 2003.
- [126] THRUN, S.B., BALA, J., BLOEDORN, E., BRATKO, I., CESTNIK, B., CHENG, J., DE JONG, K., DZEROSKI, S., FAHLMAN, S.E., FISHER, D., HAMANN, R., KAUFMAN, K., KELLER, S., KONONENKO, I., KREUZIGER, J., MICHALSKI, R.S., MITCHELL, T., PACHOWICZ, P., REICH, Y., VAFAIE, H., WELDE, W. V., WENZEL, W., WNEK, J., ZHANG, J., "The MONK's Problems A Performance Comparison of Different Learning Algorithms", Carnegie Mellon University, CMU-CS-91-197, December 1991.

[127] BLAKE, C.L., MERZ, C.J., "UCI Repository of Machine Learning Databases", [<http://ftp.ics.uci.edu/pub/ml-repos/machine-learning-databases/>]. Irvine, CA: University of California, Department of Information and Computer Science, 1998.

[128] <http://www.sgi.com/tech/mlc/db/index.html>

[129] <http://www.rensselaer.edu/dept/cs/locker0/yesterday/C4.5/Data/>

[130] BRAMER, M. A., "Inducer: A Rule Induction Workbench for Data Mining", In: IFIP World Computer Congress Conference on Intelligent Information Processing, 2000, Beijing. Proceedings. Beijing: Publishing House of Electronics Industry, p. 499-506, 2000.



## EK-A. DÜZELTİLMİŞ ENTROPİ KULLANIMININ ÖNEMİ

Düzeltilmiş entropi hesaplamasının önemi Tablo A.1’de verilen örnek seti yardımıyla açıklanacaktır.

Tablo A.1. Golf Problemi örnek seti

| Örnek No | Hava     | Sıcaklık | Nem | Rüzgar   | Karar  |
|----------|----------|----------|-----|----------|--------|
| 1        | Yağmurlu | 21       | 96  | Kuvvetli | OYNAMA |
| 2        | Yağmurlu | 17       | 70  | Kuvvetli | OYNAMA |
| 3        | Bulutlu  | 21       | 90  | Kuvvetli | OYNA   |
| 4        | Bulutlu  | 29       | 78  | Hafif    | OYNA   |
| 5        | Yağmurlu | 23       | 80  | Hafif    | OYNA   |
| 6        | Bulutlu  | 16       | 66  | Kuvvetli | OYNA   |
| 7        | Güneşli  | 23       | 70  | Kuvvetli | OYNA   |
| 8        | Güneşli  | 27       | 90  | Kuvvetli | OYNAMA |
| 9        | Güneşli  | 30       | 85  | Hafif    | OYNAMA |
| 10       | Bulutlu  | 27       | 75  | Hafif    | OYNA   |
| 11       | Yağmurlu | 18       | 80  | Hafif    | OYNA   |
| 12       | Yağmurlu | 20       | 96  | Hafif    | OYNA   |
| 13       | Güneşli  | 21       | 95  | Hafif    | OYNAMA |
| 14       | Güneşli  | 19       | 70  | Hafif    | OYNA   |

İlk önce örnek setinde *Sıcaklık* ve *Nem* karakteristikleri sayısal değerli olduklarından dolayı bu karakteristikler için aralık tanımı yapılacaktır.

*Sıcaklık* karakteristiği için 7 ve *Nem* karakteristiği için de 2 aralık tanımı yapılabilir. *Sıcaklık* karakteristiğinin en büyük değeri 30 ve en düşük değeri 16 olduğundan, aralık değeri  $(30-16) / 7 = 2$  olarak hesaplanır. Bu değerler kullanıcı tarafından da belirlenebilir. *Nem* karakteristiği 75’ten küçük-eşit değerleri ve 75’ten büyük değerleri olmak üzere iki aralığa bölünebilir. Bu bilgilere göre *Sıcaklık* ve *Nem* karakteristiklerinin yeni değerleri Tablo A.2’deki gibi tanımlanabilir:

Tablo A.2’deki aralık tanımlamalarına göre örnek seti, Tablo A.3’teki gibi yeniden düzenlenir.

Tablo A.2. Sıcaklık ve Nem karakteristikleri için aralık tanımı

| Aralık No | Sıcaklık (S)        | Nem (N)     |
|-----------|---------------------|-------------|
| 1         | $16 \leq S < 18$    | $N \leq 75$ |
| 2         | $18 \leq S < 20$    | $N > 75$    |
| 3         | $20 \leq S < 22$    |             |
| 4         | $22 \leq S < 24$    |             |
| 5         | $24 \leq S < 26$    |             |
| 6         | $26 \leq S < 28$    |             |
| 7         | $28 \leq S \leq 30$ |             |

Tablo A.3. Aralık tanımlamalarına göre düzenlenmiş örnek seti

| Örnek No | Hava     | Sıcaklık | Nem | Rüzgar   | Karar  |
|----------|----------|----------|-----|----------|--------|
| 1        | Yağmurlu | 3        | 2   | Kuvvetli | OYNAMA |
| 2        | Yağmurlu | 1        | 1   | Kuvvetli | OYNAMA |
| 3        | Bulutlu  | 3        | 2   | Kuvvetli | OYNA   |
| 4        | Bulutlu  | 7        | 2   | Hafif    | OYNA   |
| 5        | Yağmurlu | 4        | 2   | Hafif    | OYNA   |
| 6        | Bulutlu  | 1        | 1   | Kuvvetli | OYNA   |
| 7        | Güneşli  | 4        | 1   | Kuvvetli | OYNA   |
| 8        | Güneşli  | 6        | 2   | Kuvvetli | OYNAMA |
| 9        | Güneşli  | 7        | 2   | Hafif    | OYNAMA |
| 10       | Bulutlu  | 6        | 1   | Hafif    | OYNA   |
| 11       | Yağmurlu | 2        | 2   | Hafif    | OYNA   |
| 12       | Yağmurlu | 3        | 2   | Hafif    | OYNA   |
| 13       | Güneşli  | 3        | 2   | Hafif    | OYNAMA |
| 14       | Güneşli  | 2        | 1   | Hafif    | OYNA   |

Aralık tanımlamalarından sonra her karakteristik ve değerler için entropi değerleri hesaplanır. *Hava* karakteristiği için entropi değeri aşağıdaki gibi hesaplanır:

*Hava* karakteristiği için *Güneşli* değerinin geçtiği örnek sayısı 5'tir. Bu örneklerin üç tanesi *Oynama* ve iki tane ise *Oyna* sınıflarına aittir. Buna göre  $\{Hava, Güneşli\}$  için entropi aşağıdaki gibi hesaplanır:

$$E_{Hava, Güneşli} = -\frac{3}{5} \log_2 \frac{3}{5} - \frac{2}{5} \log_2 \frac{2}{5}$$

$$E_{Hava, Güneşli} = 0.971 \text{ bit}$$

*Hava* karakteristiğinin *Yağmurlu* değerinin geçtiği örnek sayısı 5'tir. Bu örneklerin üç tanesi *Oyna* ve iki tanesi ise *Oynama* sınıfında geçmektedir. Buna göre {*Hava*, *Yağmurlu*} için entropi değeri;

$$E_{Hava,Yağmurlu} = -\frac{2}{5} \log_2 \frac{2}{5} - \frac{3}{5} \log_2 \frac{3}{5}$$

$$E_{Hava,Yağmurlu} = 0.971 \text{ bit} \quad \text{olur.}$$

*Hava* karakteristiğinin *Bulutlu* değeri 4 örnekte geçmektedir. Bu örneklerin hepsi *Oyna* sınıfına aittir. {*Hava*, *Bulutlu*} için entropi değeri sıfır'dır. *Hava* karakteristiğinin entropisi aşağıdaki gibi hesaplanır:

$$E_{Hava} = \frac{5}{14} x E_{Hava,Güneşli} + \frac{5}{14} x E_{Hava,Yağmurlu} + \frac{4}{14} x E_{Hava,Bulutlu}$$

$$E_{Hava} = \frac{5}{14} x (0.971) + \frac{5}{14} x (0.971) + \frac{4}{14} x (0)$$

$$E_{Hava} = 0.694 \text{ bit}$$

*Sıcaklık*, *Nem* ve *Rüzgar* karakteristikleri için de entropi değerleri hesaplandığında Tablo A.4'teki sonuçlar elde edilir.

Tablo A.4. Golf Problemi için hesaplanan entropi değerleri

| Karakteristik | Entropi (bit) | Değer                     | Entropi (bit) |
|---------------|---------------|---------------------------|---------------|
| Hava          | 0.694         | Bulutlu                   | 0             |
|               |               | Güneşli                   | 0.971         |
|               |               | Yağmurlu                  | 0.971         |
| Sıcaklık      | 0.714         | 1 ( $16 \leq S < 18$ )    | 1.000         |
|               |               | 2 ( $18 \leq S < 20$ )    | 0             |
|               |               | 3 ( $20 \leq S < 22$ )    | 1.000         |
|               |               | 4 ( $22 \leq S < 24$ )    | 0             |
|               |               | 6 ( $26 \leq S < 28$ )    | 1.000         |
|               |               | 7 ( $28 \leq S \leq 30$ ) | 1.000         |
| Nem           | 0.789         | 1 ( $N \leq 75$ )         | 0.515         |
|               |               | 2 ( $N > 75$ )            | 0.942         |
| Rüzgar        | 0.892         | Kuvvetli                  | 1.000         |
|               |               | Hafif                     | 0.811         |

Entropi değerlerine göre örnek seti, entropisi en küçük olandan en büyük olana doğru aşağıdaki gibi sıralanır:

$$\text{Entropi(Hava)} < \text{Entropi(Sıcaklık)} < \text{Entropi(Nem)} < \text{Entropi(Rüzgar)}$$

Ancak bu sıralamaya göre kural çıkarılması durumunda *Sıcaklık* karakteristiğinin içinde bulunduğu çok sayıda kuralın üretilmesi kaçınılmazdır. Bu duruma göre örnek setine REX-1 algoritmasının uygulanması sonucu Tablo A.5'teki kurallar elde edilir. Bunun için, REX-1 algoritmasının sadece entropi değerlerini kullanması sağlanmıştır.

Tablo A.5. Entropi değerlerine göre üretilen kurallar

| Kural | Kural Tanımı                                                         |
|-------|----------------------------------------------------------------------|
| 1     | EĞER HAVA=Yağmurlu VE RUZGAR=Kuvvetli İSE KARAR=Oynama               |
| 2     | EĞER HAVA=Bulutlu İSE KARAR=Oyna                                     |
| 3     | EĞER $22 \leq \text{SICAKLIK} < 24$ İSE KARAR=Oyna                   |
| 4     | EĞER HAVA=Güneşli VE $26 \leq \text{SICAKLIK} < 28$ İSE KARAR=Oynama |
| 5     | EĞER HAVA=Güneşli VE $28 \leq \text{SICAKLIK} < 31$ İSE KARAR=Oynama |
| 6     | EĞER $18 \leq \text{SICAKLIK} < 20$ İSE KARAR=Oyna                   |
| 7     | EĞER HAVA=Yağmurlu VE RUZGAR=Hafif İSE KARAR=Oyna                    |
| 8     | EĞER HAVA=Güneşli VE $20 \leq \text{SICAKLIK} < 22$ İSE KARAR=Oynama |

Bir karakteristik için *düzeltilmiş entropi* değeri, entropinin değer sayısı ile çarpımına eşittir. *Düzeltilmiş entropi* değerleri aşağıdaki gibi hesaplanır:

$$D\text{Entropi(Hava)} = 0.694 \times 3 = 2.082 \text{ bit}$$

$$D\text{Entropi(Sıcaklık)} = 0.714 \times 6 = 4.284 \text{ bit}$$

$$D\text{Entropi(Nem)} = 0.789 \times 2 = 1.578 \text{ bit}$$

$$D\text{Entropi(Rüzgar)} = 0.892 \times 2 = 1.784 \text{ bit}$$

*Düzeltilmiş entropi* değerleri kullanılarak örnek setinin yeniden sıralanması durumunda REX-1 algoritması ile Tablo A.6'daki kurallar elde edilir.

Sadece *entorpi* kullanılması durumunda toplam sekiz kural ve bu kurallarda *Sıcaklık* karakteristiğinin bulunduğu beş şart üretilmiş ve *Nem* karakteristiği hiç

kullanılmamıştır. *Düzeltilmiş entropi* kullanılması durumunda ise toplam beş kural ve bu kurallarda *Sıcaklık* karakteristiğinin geçtiği sadece iki şart üretilmiştir. Bu şartlar ise entropilerinin sıfır olmasından kaynaklanmaktadır.

Tablo A.6. Düzeltilmiş Entropi değerlerine göre üretilen kurallar

| Kural | Kural Tanımı                                           |
|-------|--------------------------------------------------------|
| 1     | EĞER RUZGAR=Kuvvetli VE HAVA=Yağmurlu İSE KARAR=Oynama |
| 2     | EĞER HAVA=Bulutlu İSE KARAR=Oyna                       |
| 3     | EĞER $22 \leq \text{SICAKLIK} < 24$ İSE KARAR=Oyna     |
| 4     | EĞER NEM $> 75$ VE HAVA=Güneşli İSE KARAR=Oynama       |
| 5     | EĞER $18 \leq \text{SICAKLIK} < 20$ İSE KARAR=Oyna     |
| 6     | EĞER RUZGAR=Hafif VE HAVA=Yağmurlu İSE KARAR=Oyna      |

## ÖZGEÇMİŞ

1970 yılında Urfa'da doğdu. İlk, orta ve lise öğrenimini Gaziantep'te tamamladı. 1987 yılında girdiği Ankara Üniversitesi K.M.Y.O Bilgisayar Programcılığı bölümünden 1989 yılında mezun oldu. 1990 yılında girdiği İstanbul Teknik Üniversitesi, Sakarya Mühendislik Fakültesi, Endüstri Mühendisliği bölümünden 1994 yılında mezun oldu. Aynı yıl Harran Üniversitesi Bilgi İşlem Daire Başkanlığında Uzman olarak göreve başladı. 1996 yılında aynı üniversitenin Mühendislik Fakültesi Endüstri Mühendisliği bölümüne Öğretim Görevlisi ve 2001 yılında aynı bölümde Araştırma Görevlisi kadrosuna atandı. Harran Üniversitesinde, Enformatik Bölüm Başkanı olarak 2 yıl ve Bilgi İşlem Araştırma ve Uygulama Merkezi Müdürü olarak 5 yıl görev yaptı. Yüksek Lisans öğrenimini 1995-1997 yılları arasında Sakarya Üniversitesi Fen Bilimleri Enstitüsü, Endüstri Mühendisliği anabilim dalında tamamladı.

Bilgisayar programlama alanında yayınlanmış beş kitabı bulunan Ömer AKGÖBEK, evli ve üç çocuk babasıdır.