

T.C.
DOKUZ EYLÜL ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
YÖNETİM BİLİŞİM SİSTEMLERİ ANABİLİM DALI
YÖNETİM BİLİŞİM SİSTEMLERİ PROGRAMI
YÜKSEK LİSANS TEZİ

YAPAY ZEKA TABANLI
PERSONEL SEÇİM SİSTEMİ
UYGULAMASI

Ömer Faruk EREKEN

Danışman
Doç.Dr.Çiğdem TARHAN

İZMİR – 2021

TEZ ONAY SAYFASI



YEMİN METNİ

Yüksek Lisans Tezi olarak sunduğum “Yapay Zeka Tabanlı Personel Seçim Sistemi Uygulaması” adlı çalışmanın, tarafımdan, akademik kurallara ve etik değerlere uygun olarak yazıldığını ve yararlandığım eserlerin kaynakçada gösterilenlerden oluştuğunu, bunlara atıf yapılarak yararlanılmış olduğunu belirtir ve bunu onurumla doğrularım.

01/02/2021

Ömer Faruk EREKEN

ÖZET
Yüksek Lisans Tezi
Yapay Zeka Tabanlı Personel Seçim Sistemi Uygulaması
Ömer Faruk EREKEN

Dokuz Eylül Üniversitesi
Sosyal Bilimler Enstitüsü
Yönetim Bilişim Sistemleri Anabilim Dalı
Yönetim Bilişim Sistemleri Programı

Bir kurum veya işletme için en önemli varlık sahip olduğu insan gücüdür. İnsan gücünün sağlam olabilmesi için en başta doğru personelin işe alınması gerekmektedir. Günümüzde kadro gereksinimleri, aday nitelikleri ve iş başvurusu sayıları artmıştır. Bu durum incelenmesi gereken veri miktarını çok yüksek boyutlara taşıyarak doğru adayı belirleme sürecini daha karmaşık ve zor hale getirmiştir. İşe alım kararı geri dönülmesi zor, uzun vadeli sonuçlar doğuran kritik bir karardır. Bu kritik kararın aynı anda birçok farklı işi takip etmek zorunda olan sınırlı sayıdaki personel tarafından verilmesi gerekmektedir.

Yapay zeka yüksek boyuttaki verileri inceleyerek anlamlı çıktılar sunabilme yeteneği sayesinde yeni personel seçimine yardımcı olabilecek başlıca teknolojidir. Bu tez kapsamında insan kaynağı temin süreçlerinde yaşanan sorunları azaltmak ve daha kısa sürede doğru adaya ulaşılmasını sağlamak için başvuru formlarını makine öğrenmesi algoritmaları ile değerlendirerek ön eleme yapabilen bir uygulama geliştirilmiştir. İşe alım ekipleri normalde günlerce, haftalarca sürebilecek değerlendirme süreçlerini uygulama ile kısa sürelerde gerçekleştirebilecektir. Geçmişte yapılmış değerlendirmeler ile eğitilen makine öğrenmesi algoritmalarının yeni başvurular üzerinde yüksek doğruluk oranında tahminler yapabildiği gözlemlenmiştir.

Anahtar Kelimeler: Yapay Zeka, Makine Öğrenmesi, İnsan Kaynakları, Personel Seçimi

ABSTRACT
Master's Thesis
An Application Of Artificial Intelligence Based Staff Selection System
Ömer Faruk EREKEN

Dokuz Eylül University
Graduate School of Social Sciences
Department of Management Information Systems
Management Information Systems Program

The most significant asset for an institution or business is its manpower. For the manpower to be strong, the right staff should be hired first. Today, staff requirements, candidate qualifications and the number of job applications have increased. This situation made the process of identifying the right candidate more complex and difficult by increasing the amount of data to be analyzed to very high dimensions. The recruitment decision is a critical decision which is hard to reverse and has long-term consequences. This critical decision has to be made by a limited number of staff who have to follow many different tasks at the same time.

Artificial intelligence is the primary technology that can help new staff selection, thanks to its ability to provide meaningful outputs by examining large-scale data. Within the scope of this thesis, in order to reduce the problems in recruitment process and to reach the right candidate in a shorter time, an application which can pre-eliminate by evaluating application forms with machine learning algorithms has been developed. With the application, recruitment teams will be able to perform the evaluation processes which can normally take days or weeks in a short time. It has been observed that machine learning algorithms trained with the evaluations made in the past can make predictions with high accuracy on new job applications.

Keywords: Artificial Intelligence, Machine Learning, Human Resources, Staff Selection

YAPAY ZEKA TABANLI PERSONEL SEÇİM SİSTEMİ UYGULAMASI

İÇİNDEKİLER

TEZ ONAY SAYFASI	ii
YEMİN METNİ	iii
ÖZET	iv
ABSTRACT	v
İÇİNDEKİLER	vi
KISALTMALAR	x
TABLolar LİSTESİ	xi
ŞEKİLLER LİSTESİ	xii
GİRİŞ	1

BİRİNCİ BÖLÜM LİTERATÜR ARAŞTIRMASI

1.1. İNSAN KAYNAĞI TEMİNİ SİSTEMLERİ	4
1.1.1 Bilgisayar ile Puanlandırma Sistemleri	4
1.1.2 Yapay Zeka Tabanlı Sistemler	4
1.1.2.1 Özgeçmiş Değerlendirmesi	5
1.1.2.2 Oyun ve Video Mülakat Değerlendirmesi	5
1.1.2.3 Sosyal Medya Analizi	6
1.1.2.4 Sohbet Robotlarının Kullanımı	6
1.1.3 Değerlendirme ve Problemler	7

İKİNCİ BÖLÜM YAPAY ZEKA VE MAKİNE ÖĞRENMESİ

2.1. YAPAY ZEKA VE MAKİNE ÖĞRENMESİ KAVRAMLARI	9
2.2. MAKİNE ÖĞRENMESİ TÜRLERİ	11

2.2.1. Denetimli Öğrenme	11
2.2.1.1. Sınıflandırma Problemleri	11
2.2.1.2. Regresyon Problemleri	12
2.2.2. Denetimsiz Öğrenme	12
2.2.3. Pekiştirmeli Öğrenme	12
2.3. DENETİMLİ ÖĞRENME DEĞERLENDİRME ÖLÇÜTLERİ	13
2.3.1. Sınıflandırma Problemleri Değerlendirme Ölçütleri	13
2.3.1.1. Doğruluk	14
2.3.1.2. Geri Çağırma	15
2.3.1.3. Hassasiyet	15
2.3.1.4. F1 Değeri	16
2.3.2. Regresyon Problemleri Değerlendirme Ölçütleri	16
2.3.2.1. Ortalama Mutlak Hata (OMH)	16
2.3.2.2. Ortalama Karesel Hata (OKH)	17
2.3.2.3. Kök Ortalama Karesel Hata (KOKH)	17
2.4. DENETİMLİ ÖĞRENME MODELLERİ	18
2.4.1. Sınıflandırma Problemleri Modelleri	18
2.4.1.1. Lojistik Regresyon Modeli	18
2.4.1.2. K En Yakın Komşular Modeli (KEYK)	19
2.4.1.3. Destek Vektör Makineleri (DVM) Modeli	20
2.4.1.4. Karar Ağaçları Modeli	21
2.4.1.5. Topluluk Öğrenim Metodları	22
2.4.1.5.1. Rastgele Orman Modeli	23
2.4.1.5.2. XGBoost Modeli	23
2.4.2. Regresyon Problemleri Modeli	24

ÜÇÜNCÜ BÖLÜM

VERİ, ARAŞTIRMA YÖNTEMİ VE KISITLAR

3.1. VERİ	25
3.1.1. Verinin Hazırlanması	25
3.1.1.1. İsim, Soy İsim ve Cinsiyet	26

3.1.1.2. Sütunların Oluşturulması	27
3.1.1.3. Yüzde Doldur Fonksiyonu	27
3.1.1.4. Parametreler	28
3.1.1.5. Doğum Tarihleri	29
3.1.1.6. Medeni Durum	30
3.1.1.7. Askerlik Durumu	30
3.1.1.8. İkamet İli ve İlçesi	31
3.1.1.9. Eğitim Seviyesi	31
3.1.1.10. Lisans, Yüksek Lisans ve Doktora Bölümleri	32
3.1.1.11. Program Bilgisi	34
3.1.1.12. Yabancı Dil Bilgisi	35
3.1.1.13. İş Tecrübesi	36
3.1.1.14. Maaş Beklentisi	37
3.1.1.15. Var-Yok Sütunları	38
3.1.1.16. Yapay Zeka Bilgisi	39
3.1.1.17. En Son Görev Yapılan Yerdeki Ücret	40
3.2. ARAŞTIRMA YÖNTEMİ	41
3.3. KISITLAR	43

DÖRDÜNCÜ BÖLÜM

UYGULAMA

4.1. VERİ SETİNİN DEĞERLENDİRİLMESİ	44
4.2. VERİNİN MAKİNE ÖĞRENMESİ MODELİNE HAZIRLANMASI	46
4.2.1. Modele Girmeyecek Sütunların Düşürülmesi	46
4.2.2. Kategorik Verilerin Sayısallaştırılması	47
4.3. MAKİNE ÖĞRENMESİNİN ÇALIŞTIRILMASI	48
4.3.1. Lojistik Regresyon Modeli	49
4.3.2. KEYK Modeli	51
4.3.3. DVM Modeli	52
4.3.4. Karar Ağaçları Modeli	53
4.3.5. Rastgele Orman Modeli	53

4.3.6. XGBoost Modeli	54
4.3.7. Modellerin Değerlendirilmesi	55
4.4. UYGULAMANIN GELİŞTİRİLMESİ	57
4.4.1. Başvuru Girişi	57
4.4.2. Yönetici Girişi	66
4.4.2.1. Model Puanlarını Görme	68
4.4.2.2. Başvuru Değerlendirme	70
4.4.3. İletişim Bölümü	71
SONUÇ	72
KAYNAKÇA	74

KISALTMALAR

CSV	Virgülle Ayrılmış Değer (Comma Seperated Value)
DN	Doğru Negatif
DP	Doğru Pozitif
DVM	Destek Vektör Makineleri (Support Vector Machines)
KEYK	K En Yakın Komşular
KOKH	Kök Ortalama Karesel Hata
NAN	Sayı Değil (Not A Number)
OKH	Ortalama Karesel Hata
OMH	Ortalama Mutlak Hata
SGYD	Sistem Geliştirme Yaşam Döngüsü
TÜİK	Türkiye İstatistik Kurumu
XGBOOST	Ekstrem Gradyan Artırım (Extreme Gradient Boosting)
YDS	Yabancı Dil Bilgisi Seviye Tespit Sınavı
YN	Yanlış Negatif
YP	Yanlış Pozitif

TABLÖLAR LİSTESİ

Tablo 1: Hata Matrisi	s. 13
Tablo 2: Kadro Nitelikleri	s. 45
Tablo 3: Lojistik Regresyon Modeli Değerlendirme Sonuçları	s. 50
Tablo 4: Standardize Model Değerlendirme Sonuçları	s. 51
Tablo 5: KEYK Modeli Değerlendirme Sonuçları	s. 52
Tablo 6: DVM Modeli Değerlendirme Sonuçları	s. 52
Tablo 7: Karar Ağaçları Modeli Değerlendirme Sonuçları	s. 53
Tablo 8: Rastgele Orman Modeli Değerlendirme Sonuçları	s. 54
Tablo 9: XGBoost Modeli Değerlendirme Sonuçları	s. 55
Tablo 10: Model Değerlendirmeleri	s. 55

ŞEKİLLER LİSTESİ

Şekil 1: Yapay Zeka Kümesi	s. 10
Şekil 2: Denetimli Öğrenmede Nitelikler ve Çıktı	s. 11
Şekil 3: Lojistik Regresyon	s. 18
Şekil 4: KEYK Gruplandırması	s. 19
Şekil 5: DVM Hiper Düzlem	s. 20
Şekil 6: Karar Ağaçları Diyagramı	s. 21
Şekil 7: Topluluk Öğrenimi Örneği	s. 22
Şekil 8: Doğrusal Regresyon Grafiği	s. 24
Şekil 9: Python Kütüphane Yüklemeleri	s. 26
Şekil 10: Aday Listesinin Oluşturulması	s. 27
Şekil 11: Yüzde Doldur Fonksiyonu	s. 28
Şekil 12: Parametreler Örneği	s. 28
Şekil 13: Doğum Tarihleri Üretimi	s. 29
Şekil 14: Yaş Dağılımları	s. 29
Şekil 15: Medeni Durum Dağılımı	s. 30
Şekil 16: Askerlik Durumu Dağılımı	s. 30
Şekil 17: Eğitim Seviyesi Dağılımı	s. 31
Şekil 18: Veri Bilimci İlk 5 Bölüm	s. 32
Şekil 19: Bölüm Dağılımları	s. 33
Şekil 20: Program Bilgisi Dağılımı	s. 34
Şekil 21: Yabancı Dil Bilgisi Oranları	s. 35
Şekil 22: İş Tecrübesi Dağılımı	s. 36
Şekil 23: Maaş Beklentisi Dağılımı	s. 37
Şekil 24: Var-Yok Sütunları	s. 38
Şekil 25: Yapay Zeka Bilgisi Dağılımı	s. 39
Şekil 26: Son Ücret Oranları	s. 40
Şekil 27: Sistem Geliştirme Yaşam Döngüsü	s. 41
Şekil 28: Karar Oranları	s. 46
Şekil 29: Kategorikten Sayısala Dönüşüm	s. 47
Şekil 30: Modele Hazır Veri Seti	s. 48

Şekil 31: Kütüphaneler	s. 49
Şekil 32: Verinin Bölünmesi	s. 49
Şekil 33: Lojistik Regresyon Modeli	s. 50
Şekil 34: Standardize Lojistik Regresyon Modeli	s. 50
Şekil 35: KEYK Modeli	s. 51
Şekil 36: DVM Modeli	s. 52
Şekil 37: Karar Ağaçları Modeli	s. 53
Şekil 38: Rastgele Orman Modeli	s. 54
Şekil 39: XGBoost Modeli	s. 54
Şekil 40: Öğrenim Türleri	s. 56
Şekil 41: Başvuru Sekmesi Kodları	s. 58
Şekil 42: Uygulama Ekranı 1	s. 58
Şekil 43: Cinsiyet, Medeni Durum ve Askerlik Durumu Kodları	s. 59
Şekil 44: Uygulama Ekranı 2	s. 59
Şekil 45: Eğitim Durumu Kod Örneği	s. 60
Şekil 46: Uygulama Ekranı 3	s. 61
Şekil 47: Programlama Dilleri, Yapay Zeka Kursu Kodları	s. 62
Şekil 48: Uygulama Ekranı 4	s. 62
Şekil 49: Yapay Zeka Bilgisi, Dil, Alan Tecrübesi Kodları	s. 63
Şekil 50: Uygulama Ekranı 5	s. 63
Şekil 51: Adli Sicil, Seyahat, Referans, Vardiya Kodları	s. 64
Şekil 52: Uygulama Ekranı 6	s. 65
Şekil 53: Veri Seti Sözlüğünün Bir Kısım	s. 65
Şekil 54: Başvuru Tamamlama Kodları	s. 66
Şekil 55: Yönetici Girişi Kodları	s. 66
Şekil 56: Yönetici Yan Sekme Ekranı	s. 67
Şekil 57: Veri Setini Yükleme Kodları	s. 68
Şekil 58: Model Değerlendirme Kodları	s. 69
Şekil 59: Uygulama Ekranı 7	s. 69
Şekil 60: Uygulama Ekranı 8	s. 70
Şekil 61: Model Değerlendirme Kodları	s. 70
Şekil 62: Başvuru Değerlendirme Ekranı	s. 71

Şekil 63: İletişim Bölümü Kodları

s. 71

Şekil 64: İletişim Uygulama Ekranı

s. 71



GİRİŞ

İnsanlık tarihinin başlangıcından itibaren insanlar yeteneklerine göre iş paylaşımı yaparak belirli rolleri sahiplenmişlerdir. Bununla birlikte iş paylaşımının nasıl yapılması gerektiği, bir iş için ehil olanın kim olduğu sorunsalı ortaya çıkmıştır. Doğru kişinin seçilmesi maksadıyla tarihsel süreçte pek çok farklı yöntem izlenmiştir.

Günümüz dünyasında kurum, işletme veya şirketler için en önemli değer sahip oldukları insan gücüdür. İşleri yöneten personel gücü sağlam olmadığı takdirde sahip olunan maddi olanaklar değerlerini yitirebilmektedir. Bu nedenle iş verenler, iş yerlerine en büyük faydayı sağlayacak kişileri seçmek için çaba göstermektedirler. Her ne kadar yapılacak işlerin niteliği değişse de doğru kişinin nasıl seçileceği sorunu tüm iş kolları için varlığını korumaktadır.

Büyük ölçekli işletmelerde insan kaynakları bölümü, küçük ölçekli işletmelerde ise görevlendirilen personel veya direkt olarak yönetici tarafından gerçekleştirilen işe alım süreci, uzun ve meşakkatli bir süreçtir. Bunun yanında alınan kararlar uzun vadeli sonuçlar doğuran, geri dönüşü çok zor kararlardır.

Tüm iş kollarının profesyonelleştiği, alanında eğitim almadan, diploma, sertifika sahibi olmadan işe girmenin çok zor olduğu günümüz dünyasında işe alımdan sorumlu kişiler, başvuru yapan binlerce kişinin bilgilerini, tek tek, detaylı bir şekilde incelemek zorunda kalmaktadır. İnsan gözüyle bakıldığında tasnif etmenin çok zor olduğu yüzlerce bilgiyi okuyarak ön eleme, görüşmeye çağırma ve derecelendirme süreçlerini yürütmektedirler.

Ancak neredeyse tüm alanlarda insanoğlunun yardımına koşmaya başlayan yapay zeka ve makine öğrenmesi teknolojisi işe alım süreçlerinde de kullanıldığı takdirde haftalarca, aylarca sürebilecek olan süreçler çok kısa sürelerde gerçekleştirebilir ve işe alım ekiplerinin enerjilerini doğru yerlere yönlendirebilir.

İnsan kaynağı temini bir iş yerinin faaliyetlerini devam ettirebilmek için ihtiyaç duyduğu personeli işe almasıdır. İşe alım faaliyeti belirli seçim aşamaları dahilinde takip edilmektedir. Bu tezin amacı uzun ve zorlu bir süreç olan personel seçim sürecini, yapay zeka yöntemlerini kullanarak daha hızlı ve doğru şekilde gerçekleştirmeyi sağlayacak örnek bir uygulama ortaya koymaktır.

Kurumlar için insan kaynağı temin süreci temel manada üç aşamada tarif edilmektedir. Birinci aşama adayları bulma, ikinci aşama adaylar arasından doğru kişiyi seçme ve son aşama da seçilen personeli işe alıştırmadır (Tonus, 2018: 63). Burada atlanmaması gereken nokta temin süreci öncesinde yapılması gereken hazırlıklardır. Her şeyden önce, doğru personel temini yapılabilmesi için alım yapılacak işin tanımı ve adaylarda bulunması beklenen nitelikler net ve belirgin olmalıdır.

Her faaliyette olduğu gibi insan kaynağı temin sürecinde de en kritik noktalardan birisi hazırlık sürecinin doğru yapılmasıdır. Öncelikle boşalan ya da ihtiyaç duyulan kadronun tanımının doğru şekilde yapılması, varsa geçmiş yıla ait kayıtların, çalışmaların temin edilerek incelenmesi ve şartlarda revizyon gerekiyorsa yapılması gerekmektedir. Geçmiş yıllara ait veriler insan kaynağı temininde hazırlık sürecinin önemli bir parçasını oluşturmaktadır. Bir iş yerinde kurumsal hafızanın korunması o iş yerinin başarısı ve işleyişinin devamı açısından kritiktir. İnsan kaynağı temininde çoğunlukla temin yapılacak kadrolar geçmiş yıllarda da personel alımı yapılmış olan kadrolardır. Kurumsal hafıza kapsamında geçmiş yılların tecrübelerinin yeni personel temininde kullanılması çok büyük avantajlar sağlayacaktır.

Önceki yıllarda yapılmış temin faaliyetlerinde başvuru yapan binlerce kişi arasından bazı adaylar koşulları sağlamadığı için ön elemeye tabi tutulmuş, bazı adaylar da temin sürecinde elenmiş, sona kalan birkaç kişi işe kabul edilmiştir. Aslında geçmiş yıllarda yapılan seçimler şirketin işe alım politikasında bir model ortaya koymuştur. Ancak tüm bu verilerin zaten yoğun olan temin sürecinde, birçok farklı işin arasında personel tarafından tekrar incelenmesi çok zor olacak ve genellikle ihmal edilecektir. Önceki temin faaliyetinde yer alan personel yeni teminde görevlendirilemez ve geçmişteki o veriler de incelenmez ise yeni temin faaliyeti kurumsal hafızadan uzak, aynı zorlukların tekrar yaşandığı bir süreç olacaktır.

Yapay zeka tabanlı insan kaynağı temini uygulamasının en büyük avantajlarından birisi de hazırlık süreci kapsamında değerlendirilebilecek olan geçmiş verilerin incelenmesi sürecini çok kısa sürede gerçekleştirmesi, gereken dersleri alması ve elde edilen bilgiler ışığında çok kısa sürede yeni adayları süzerek analiz edebilmesidir.

İnsan kaynağı bulma, temin yapılacak kadrolar için başvurmaya istekli kişilere ulaşma ve adayların bilgilerini istenen formatta ilgililere ulaştırmalarını sağlama sürecidir (Tonus, 2018: 66). Günümüzde iletişim teknolojilerinin sağladığı avantajlar sayesinde iş ilanları özellikle internet üzerinden yapılan reklam ve ilanlar sayesinde geniş kitlelere duyurulabilmektedir. Personel temini yapılacak kadroya başvuran tüm adayların oluşturduğu gruba literatürde “Aday Havuzu” denmektedir. Normal şartlarda aday havuzunun çok geniş olması veya tam tersi çok dar olması istenen bir durum değildir (Uyargil’den aktaran Tonus, 2018: 66).

Günümüzde iletişim yöntemleri sayesinde ulaşılan geniş kitleler ve nitelikli işsiz sayısının artmasıyla birlikte aday havuzlarının genişlemesi beklenen bir durum olacaktır. Ancak yapay zeka ve makine öğrenmesi yöntemleriyle birlikte çok geniş aday havuzları arasından seçim yapma süreci dahi ön eleme ve hızlı analiz imkanlarıyla birlikte hızlanacak ve kolaylaşacaktır.

İnsan kaynağı seçme, aday havuzundan personel alımı yapılacak kadroya en uygun adayı belirleme sürecidir. Burada dikkat edilmesi gereken nokta yapılacak seçim en iyi, en nitelikli kişiyi seçmeye yönelik değil iş için en uygun adayı seçmeye yöneliktir. Bir aday tüm nitelikleriyle birlikte değerlendirildiğinde aday havuzundaki en iyi kişi olabilir ancak bizim gerekliliklerimiz çerçevesinde değerlendirildiğinde en doğru seçim başka bir aday çıkabilecektir (Tonus, 2018: 70).

Klasik yöntemler kullanıldığında insan kaynağı temininde en uzun süreç seçim sürecidir. Öncelikle başvuru formları incelenerek ön eleme yapılması gerekmektedir. Sonrasında duruma göre ön görüşme ve sınav yapılabilir. Genellikle son süreç olarak mülakat yapılarak adaylar arasından kadro sayısı kadar aday seçilmektedir.

Bu tezdeki ilgi odağı insan kaynağı temin sürecinin tamamen insandan bağımsız olması değil, insan kaynakları veya işe alım ekiplerinin faaliyetlerine yardımcı olacak şekilde makine öğrenmesi algoritmalarının kullanılmasıdır. Süreç, uzmanlardan bağımsız olarak düşünülemez. Uzmanların yönlendirdiği bir makine öğrenmesi modeliyle özellikle ön eleme ve puanlandırma süreçlerinin kolaylaştırılması ve eldeki dağınık verilerden anlamlı bilgilere ulaşılması hedeflenmektedir.

BİRİNCİ BÖLÜM

LİTERATÜR ARAŞTIRMASI

1.1 İNSAN KAYNAĞI TEMİNİ SİSTEMLERİ

İnsan kaynağı temininde karar vericilere yardımcı olabilmek için geçmişte bilgisayar tabanlı birçok çalışma yapılmıştır. Yapay zeka teknolojisinin henüz gelişmediği zamanlarda bu sistemler daha çok bilgisayarın kayıt maksatlı kullanımı, görsel analiz araçlarından faydalanma veya bilgisayarı belirli koşullara programlamayla sonuç elde etme amaçlarına yönelikti. Günümüzde yapay zeka teknolojisinin gelişmesiyle birlikte çoğu alanda olduğu gibi insan kaynakları disiplini de entegrasyon çalışmaları hızlı bir şekilde devam etmektedir.

1.1.1 Bilgisayar ile Puanlandırma Sistemleri

Aday değerlendirmeleri için kullanılan ilk sistemler daha çok adayları özelliklerine göre puanlandırma üzerine kuruluydu. Yazılımcı tarafından olası veri girişleri belirli puan ağırlıklarıyla eşleştirilmekte ve sonuçta puanlar toplanarak bir skor elde edilmekteydi. 2005 yılında Konya Büyükşehir Belediyesinde personel seçimi gerçekleştirmek üzere hazırlanan “Expert-Personel Uzman Sistemi” puanlandırma ile kurulan personel seçim sistemlerine bir örnek teşkil etmektedir (Kaya ve Gözen, 2005: 355-376).

Puanlandırma sistemlerinin dezavantajı sistemin tamamen yazılımcının yönlendirmeleri doğrultusunda, belirli bir çerçeve dahilinde hareket etmesidir. Yani eğer-ise yapılarıyla kurulan temel yazılım mantığı kullanılmaktadır. Günümüz yapay zeka teknolojilerinde ise sistemin bilinç yeteneğine erişerek, bir karar verici gibi hareket etmesi hedeflenmektedir.

1.1.2 Yapay Zeka Tabanlı Sistemler

Yapay zekanın insan kaynağı temininde kullanımı için uzun yıllardır çalışmalar yapılmaktadır. Ancak özellikle 2018 yılında yapay zeka teknolojisi işe alım uzmanları

arasında popüler hale gelmiştir (Upadhyay ve Khandelwal, 2018: 255). Literatür araştırmasında da 2018 ve yakın zamanlı yıllarda bu alandaki çalışmaların, haberlerin sayısında gözle görülür bir artış olduğu gözlemlenmektedir.

Büyük çaplı pek çok uluslararası şirket zaman ve kaynak tasarrufu sağlayabilmek için yapay zekanın personel seçim sürecinde kullanımına yatırım yapmakta ve işe alımlarda teknolojiye faydalanmaktadır.

Personel seçim sürecinde yapay zekanın kullanımı belirli bir çerçevede sınırlandırılmamalıdır. Özgeçmiş göre değerlendirme, sanal oyunlar ile değerlendirme, online mülakat, sohbet robotuyla iletişim, sosyal medya değerlendirmesi gibi birçok farklı yöntem aday değerlendirmelerinde kullanılabilmektedir.

1.1.2.1 Özgeçmiş Değerlendirmesi

Yapay zekanın personel seçiminde kullanım alanlarından birisi adaylara ait bilgilerin yer aldığı özgeçmiş formlarının incelenmesiyle bilgi edinme yöntemidir (Strohmeier ve Piazza, 2015:164-165). Örneğin Fareportal isimli şirket iş ilanlarındaki kadro gerekliliklerini analiz ederek adayların özgeçmiş formuyla kıyaslayan “Resume Scorer” isimli bir uygulama geliştirmiştir. Böylece insan kaynakları uzmanları tek tek özgeçmiş formlarını incelemeye ayıracakları vakti en başarılı adayları incelemeye ayırmaktadır (Leong, 2018: 50-52).

1.1.2.2 Oyun ve Video Mülakat Değerlendirmesi

Yıllık 1.8 milyon iş başvurusu alan ve bu adaylar arasından 30 binden fazla kişiyi işe alan Unilever şirketi, Pymetrics isimli işe alım uzmanlık hizmeti sağlayan şirket ile anlaşarak adayları kendi ev ortamlarında bilgisayar veya cep telefonu aracılığıyla uzaktan değerlendirebilecek bir yapay zeka sistemi kurmuştur. Öncelikle adayların bazı oyunları oynamaları beklenmektedir. İkinci aşamada adaylardan belirli soruları cevaplandırdıkları 30 dakikalık bir video mülakatı göndermeleri istenmektedir. Aşamalar makine öğrenmesi algoritmaları ile değerlendirilerek sonuçlar elde edilmektedir. Şirketin insan kaynakları bölümü yöneticisi Leena Nair

yapay zeka sistemi sayesinde 70 bin personel ile saatlerce süren mülakatlar ve değerlendirme aşamalarından kurtulduklarını belirtmiştir. Örneğin şirket yapay zeka değerlendirmeleri ile milyonlarca aday arasından 3500 kişiyi değerlendirme merkezine davet etmekte ve bu adaylar arasından da 800 kişiyi işe kabul etmektedir. Sistem aynı zamanda başvuruda bulunan herkese geri dönüş yaparak şirket memnuniyetini üst seviyelerde tutmaktadır (Marr, 2018).

1.1.2.3 Sosyal Medya Analizi

İnsanların kariyer durumlarını sergilediği, iş arayan ve işverenlerin bulunduğu, iş dünyasının sosyal medya platformu denilince ilk akla gelen şirket LinkedIn'dir. LinkedIn sadece sosyal medya mecrası sağlamakla kalmamış aynı zamanda kendi yapay zeka destekli işe alım sistemi "LinkedIn Recruiter"u da geliştirmiştir.

LinkedIn Recruiter işverenlere aradıkları özellikler kapsamında yapay zeka teknolojisini kullanarak sosyal medya platformu üzerinden potansiyel adayları sunmaktadır. Sonrasında işveren ve iş arayanlar arasında sosyal medya üzerinden iletişim imkanı sağlayarak iki tarafı buluşturmaktadır (LinkedIn, 2020).

Sosyal medya analizini kullanan farklı çalışmalar da bulunmaktadır. Örneğin 2014 yılında makine öğrenmesi algoritmaları kullanarak yapılmış bir çalışmada sistem adayların LinkedIn profillerindeki belirli nitelikleri iş gereksinimleri ile karşılaştırarak çıkarımlar elde etmektedir. Aynı zamanda adayların blog gönderilerinde yazdıkları içerikler dil analizi ile değerlendirilmekte ve sonuçta adayın kişisel karakteristikleri belirlenmektedir (Faliagka ve diğerleri, 2014: 515-528).

1.1.2.4 Sohbet Robotlarının Kullanımı

İnsan kaynağı temin faaliyetlerinde kullanılan yapay zeka teknolojilerinden birisi sohbet robotlarıdır. Sohbet robotları NLP (Doğal Dil İşleme) teknolojisi ile geliştirilirler ve dil kullanma yeteneğine erişmişlerdir. Sahip oldukları bu yetenek sayesinde insan kaynağı temin faaliyetlerinde mülakat veya adaylarla diğer iletişim ihtiyaçları maksadıyla kullanılabilirler.

Sohbet robotları sayesinde aday başvuru formu tamamlanabilir, ayrıntılı mülakat yapılabilir, adayın şirket hakkındaki soruları cevaplanabilir veya mülakat tarihi belirlenebilir. Fakat sohbet robotlarının zaman ve kaynak tasarrufu, adaylarda stres seviyesini azaltması gibi faydalı yanlarıyla birlikte zayıf oldukları yönler de bulunmaktadır. Bunlar; empati ve mizah yönü eksikliği, eğitildiği alan (veri topluluğu) dışında bir durumla karşılaştığında yetersiz kalma ve net olmayan cevapları değerlendirememesi gibi sıkıntılardır (Joshi, 2019).

1.1.3 Değerlendirme ve Problemler

Yapay zeka tabanlı işe alım sistemlerinin insan kaynağı temininde büyük faydaları bulunmaktadır. Büyük aday havuzları arasından doğru kişilerin belirlenebilmesi, yüksek orandaki verilerin incelenebilmesi, daha yüksek sayılarda adaya erişebilme, insan kaynakları uzmanlarının enerjisinin doğru alanlara yönlendirilebilmesi, uzun soluklu ve insan önyargılarının yer alabildiği mülakat süreçlerinin daha doğru değerlendirilebilmesi, güçlü sosyal medya analizlerinin mümkün olması, tüm adaylara geri besleme imkanı ve adaylar evden ayrılmadan değerlendirme yapılabilmesi gibi pek çok fayda sayılabilir.

Ancak insan kaynağı temini direkt olarak insan olgusuyla ilgilendiği için hassas bir alandır. Bu sektörde gerçekleşecek olası yapay zeka hataları bir evin veya arabanın fiyat tahmininde gerçekleşen hatalardan daha ciddi sonuçlar doğurabilecektir. Günümüzde nihai kararlar her ne kadar sorumlu yöneticilere bırakılsa da sistem tarafından yapılan yanlış yönlendirmeler de hem şirket hem de bu alandaki çalışmalar için olumsuz etki oluşturabilirler.

2018 yılında basında yer alan bir habere göre büyük bir online ticaret şirketi işe alım faaliyetleri için yapay zeka teknolojisi denemeleri yapıyordu. Ancak şirket bir süre sonra yapay zekanın kadın adaylara karşı negatif bir önyargı geliştirdiğini fark etti ve teknolojinin kullanımını durdurdu. Bunun sebebi şirketin geçmişteki 10 yıllık iş başvurusu verilerini hesaba katması ve o verilerin de çoğunda erkek adayların bulunmasıydı. Makine öğrenmesi bu verilerden yanlış bir öğrenim gerçekleştirmiş ve sonuç olarak önyargılı bir yapay zeka sistemi ortaya çıkmıştı (Dastin, 2018).

Öte yandan şirketlerin yapay zeka tabanlı sistemlere geçiş yapabilmesi için gerekli alt yapı ve personel eğitimlerini tamamlamaları gerekmektedir. Hazırlık yapılmadan geçilmiş bir bilişim sistemi şirketler için yatırım ve zaman kayıplarına sebep olabilir.

2018 yılında LinkedIn tarafından yayınlanan rapora göre Kuzey Amerika'da bulunan şirketlerin sadece %22'si insan kaynakları veri analitiğini kurabilmiştir (LinkedIn, 2018: 8). Deloitte'nin 2017 Küresel İnsan Kapitali Trend Raporu'na göre şirketlerin %71'i insan analitiğini yüksek öncelikte görmekte ancak sadece %9'u kendi organizasyonları için hangi yeteneğin performans getireceğini iyi anladığını ve %8'i kullanışlı veriye sahip olduğunu ifade etmektedir (Collins ve diğerleri, 2017: 97). Yapay zeka ile insan kaynağı temininde faydalı veri çok önemli bir yer tutmaktadır. Bahsi geçen raporlar ise bu alanda daha fazla yatırım ve çalışma yapılması gerektiğini kanıtlamaktadır.

İKİNCİ BÖLÜM

YAPAY ZEKA VE MAKİNE ÖĞRENMESİ

2.1. YAPAY ZEKA VE MAKİNE ÖĞRENMESİ KAVRAMLARI

İnsan dünyada gördüğümüz en karmaşık ve zeki varlıktır. Ancak insanın tek başına, kendi varlığıyla ulaşabileceği noktalar sınırlıdır. Bundan dolayı tarihsel süreç içerisinde insanoğlu karşılaştığı problemlere çözüm üretmek, sınırlarını aşabilmek için belirli alanlarda kendisinden daha başarılı olabilen pek çok ürün ortaya koymuştur. Yapay zeka da insan tarafından geliştirilen, insan zekasının bir ürünü olan, makinelerin öğrenmesine ve onlara zeka kazandırılmasına dayalı bir kavramdır.

Yapay zekanın temelleri 1950 yılında Matematikçi Alan M. Turing'in yayınladığı "Hesaplayan Makineler ve Zeka" makalesine dayanmaktadır. "Makineler Düşünebilir mi?" sorusuyla makalesine giriş yapan Turing makinelerin düşünme ve zeka kabiliyetlerinin test edildiği ve kendi adıyla anılacak olan "Turing Testi"ni ortaya koymuştur (Turing, 1950: 433-460).

Makine öğrenmesi kavramı 1950'li yıllarda Arthur Samuel tarafından ortaya atılmıştır. Samuel o yıllarda hatalarından öğrenerek kendisini sürekli geliştiren dama oynayan bir bilgisayar programlamıştır (Fogg, 2017; Samuel, 1959: 211). Derin öğrenmenin tarihi ise 1943 yılında W.Pitts ve W.McCulloch tarafından insan beyninin sinir ağlarına benzer yapıda oluşturulan bilgisayar modeline dayandırılmaktadır. Ancak ilk derin öğrenme algoritmasının A.G.Ivakhnenko ve V.G.Lapa tarafından 1965 yılında ortaya konduğu söylenebilir. Yapay sinir ağları kapsamında derin öğrenme kavramının ilk defa kullanımı ise I.Aizenberg ve arkadaşları tarafından 2000 yılında olmuştur (Foote, 2017; Şeker ve diğerleri, 2017: 48-49).

Makine öğrenmesi yapay zeka çatısının, derin öğrenme de makine öğrenmesi çatısının altında bulunan alt disiplinlerdir (Khanna, 2019: 457). Yapay zeka kümesinin temsili Şekil 1'de gösterilmiştir. Bu çalışmada yapay zeka kümesi altında bulunan makine öğrenmesi yöntemleri kullanılacaktır.

Şekil 1: Yapay Zeka Kümesi



Kaynak: Yazar tarafından derlenmiştir.

Makine öğrenmesi bilgisayar tarafından istatistiksel yöntemler kullanılarak belli sonuçlar elde edilmesi hedeflenen bir daldır. Bilgisayarın verilerden belirli kurallar ortaya çıkarması beklenmektedir. Burada dikkat edilmesi gereken husus bilgisayara belirli kurallar programlanarak verilerden çıktılar oluşturulması makine öğrenmesi değil kurallar dahilinde bilgisayarın programlanmasıdır. Makine öğrenmesinde bilgisayar, verileri matematiksel modeller kullanarak karşılaştırır ve onlardan öğrenir. Daha sonrasında sadece veri girişi yapıldığında, çıktıları tahmin eder ya da verileri tasnif eder.

Günümüzde hayatımızda çok önemli bir yer tutan internet ve burada oluşturulan veri toplulukları ile bunlardan anlamlı sonuçlar çıkarma arayışı hız kazanmıştır. Yapay zeka artık hayatımızın her alanına girmektedir. Fiyat tahmin sistemleri, görsel algılama ve analiz, ticari tahmin ve analizler, öneri motorları ve daha pek çok alanda yapay zeka ve makine öğrenmesi kullanılmaktadır. Muhtemeldir ki önümüzdeki yıllar yapay zeka teknolojisini doğru kullanan, hayata, sektöre entegre edebilenlerin başarılı olabildiği bir dönem olacaktır.

2.2. MAKİNE ÖĞRENMESİ TÜRLERİ

Makine öğrenmesi 3 farklı türe ayrılmaktadır. Bunlar denetimli öğrenme, denetimsiz öğrenme ve pekiştirmeli öğrenmedir (Dasgupta ve Nath, 2016: 6). Bu tezde denetimli öğrenme türünün sınıflandırma problemi modelleri kullanılacaktır.

2.2.1. Denetimli Öğrenme

Denetimli öğrenme, girdi ve çıktıların makine öğrenmesi modeline birlikte girdiği öğrenme türüdür. Burada sistem matematiksel algoritmalar yardımıyla nitelikler (girdiler) ve çıktı (etiket) arasında bir sebep-sonuç ilişkisi kurarak bir tahmin modeli üretmekte, yani öğrenme işlemini gerçekleştirmektedir. Üretilen tahmin modeli sayesinde daha önceden görülmemiş yeni girdilerin çıktıları tahmin edilebilmektedir. Denetimli öğrenme farklı kaynaklarda farklı alt kümeler ile ifade ediliyor olsa da temel anlamda sınıfsal ve regresyon (sayısal çıktı tahmini) tipi iki probleme çözüm üretmektedir (Hastie ve diğerleri, 2009: 1-10). Şekil 2'deki örnekte görüldüğü üzere burada sondaki hariç sütunlar nitelikleri ve sondaki sütun ise hedefi ya da diğer bir ifade şekliyle bu verilerden ulaşılması beklenen anlamlı bilgiyi temsil etmektedir.

Şekil 2: Denetimli Öğrenmede Nitelikler ve Çıktı

	yaş	medenidurum	egitim	programbilgisi	yabancidilbilgisi	alantecrübesi	maaşbeklentisi	seyahatengeli	referans	ensonücret	vardıyalımkı	karar
0	24	bekar	lisans	java, julia	ingilizce	5.0	4000.0	yoktur	yoktur	3000.0	çalışabilirim	Ön Görüşmeye Çağrılacak
1	32	evli	lisans	java, julia	ingilizce	6.0	5000.0	yoktur	vardır	2000.0	çalışabilirim	Ön Görüşmeye Çağrılacak
2	27	bekar	ön lisans	python, scala	bilmiyor	3.0	4000.0	yoktur	yoktur	3000.0	çalışabilirim	Elendi
3	27	evli	ön lisans	python, ofis programları	bilmiyor	5.0	4000.0	yoktur	yoktur	3000.0	çalışabilirim	Elendi
4	31	evli	lisans	python, ofis programları	ingilizce, fransızca	8.0	8000.0	yoktur	yoktur	2000.0	çalışabilirim	Ön Görüşmeye Çağrılacak

Kaynak: Yazar tarafından derlenmiştir.

2.2.1.1. Sınıflandırma Problemleri

Sınıflandırma insan yaşamında en sık karşılaşılan karar verme problemlerinden birisini oluşturmaktadır. Burada verilerden ulaşılacak istenen anlamlı bilgi sınıfsal bir

nitelik arz etmektedir (Sathya ve Abraham, 2013: 36). Yani makine öğrenmesinden sayısal, sonsuzluk eğrisinde bir tahmin üretmesi değil kategorik bir tahmin yapması beklenmektedir. Örneğin verilere göre yapılan sahte-gerçek, geçti-kaldı, kanatlı-sürüngen gibi tahminler sınıfsal problemlere ait çıktılardır.

2.2.1.2. Regresyon Problemleri

Denetimli öğrenmede ikinci problem tipi devamlı, sonsuz nitelik arz eden çıktıların tahmin edilmesi problemidir. Regresyon problemlerinde sınıfsal problemlerden farklı olarak tahmin edilmek istenen değer sayısal niteliktedir. Sistemin gözlemlerden elde ettiği model ile sonsuz skalada bir değer tahmini yapması beklenmektedir. Ev kira veya satış fiyatı, araba fiyatı tahmini gibi sayısal çıktılar regresyon problemlerinin ürünüdür.

2.2.2. Denetimsiz Öğrenme

Denetimsiz öğrenmede denetimli öğrenmeden farklı olarak çıktılar, yani anlamlı bilgiler yoktur. Etiket (çıktı) bilgisi olmadan makine öğrenmesi modeline giren verilerden sonuç alınmaya çalışılmaktadır. En yaygın türü olan kümeleme analizinde sistem matematiksel algoritmalar ile incelediği verileri gruplandırmaktadır. (Polat, 2017: 446). Ancak sistem hangi grubun ne olduğunu bilemez sadece niteliklerden kümeler oluşturmuş olur. Sonrasında alan uzmanı kişiler kümeleri inceleyerek anlamlı bilgiye ulaşırlar.

2.2.3. Pekiştirmeli Öğrenme

Pekiştirmeli öğrenme yöntemi dünyadaki canlı varlıkların öğrenme yöntemine benzemektedir. Pekiştirmeli öğrenme; en yüksek ödül bildirimini alabilmek için ne yapmak gerektiğini öğrenmektir. Eğitilen, hangi davranışları izlemesi gerektiği konusunda yönlendirilmemektedir. Onun yerine eğitilen, deneme yöntemiyle hangi davranışın en yüksek ödülü kazandıracakını keşfetmektedir. Pekiştirmeli öğrenme etiketsiz verideki saklı yapıyı bulmaya çalışan denetimsiz öğrenme ile karıştırılmamalıdır (Sutton ve Barto, 2018: 1-2).

2.3. DENETİMLİ ÖĞRENME DEĞERLENDİRME ÖLÇÜTLERİ

Makine öğrenmesinde model eğitildikten sonra belirli girdiler yapılarak tahminler alınır. Fakat model her zaman doğru tahminlerde bulunamayabilir veya regresyon problemlerinde tahmin ettiği değerler doğru değere belirli oranlarda yaklaşır ancak tam değeri tahmin edemeyebilir.

Eğitilen modellerin geliştirilmesi veya sonuçlarının istatistiki olarak ifade edilebilmesi için modellerin analiz şekline göre değerlendirme ölçütleri bulunmaktadır. Bu tezde değerlendirme ölçütlerinin yalnızca bir kısmına değinilmiştir.

2.3.1. Sınıflandırma Problemleri Değerlendirme Ölçütleri

Sınıflandırma analizi değerlendirme ölçütlerine geçmeden önce ölçütlerde kullanılan hata matrisini incelememiz gerekmektedir. Hata matrisi sınıflandırma modellerinin tahminlerinin değerlendirilmesine olanak sağlamaktadır (Kılınç ve diğerleri, 2015: 117, Sokolova ve Lapalme, 2009: 429).

Hata matrisinde tahmin değerleri ile gerçek değerler kıyaslanmakta ve isimlendirilmektedir. Tablo 1’de görüldüğü üzere üst kısımda tahmin değerleri, yan tarafta ise gerçek değerler bulunmaktadır. Burada pozitif veya negatif değerler yapılan sınıflandırmanın niteliğini arz etmektedir.

Tablo 1: Hata Matrisi

		TAHMİN DEĞERİ	
		-	+
GERÇEK DEĞER	-	DN DOĞRU NEGATİF	YP YANLIŞ POZİTİF <u>I.TİP HATA</u>
	+	YN YANLIŞ NEGATİF <u>II.TİP HATA</u>	DP DOĞRU POZİTİF

Kaynak: Portilla, 2020

Hata matrisini bir örnek üzerinden anlatmak daha doğru anlaşılmasını sağlayacaktır. Örneğin kişilerin seyahat vizesi alıp alamayacaklarının değerlendirildiği bir veri seti olsun. Burada gözlemdaki kişi vize alabilecek iken model doğru şekilde vize alabilir değerlendirmesinde bulunursa tabloda sağ altta yer alan doğru pozitif

değerlendirmesi yapılmış olur. Eğer kişi vize alabilecek iken vize alamaz tahmininde bulunulursa sol altta yer alan yanlış negatif analizi yapılmıştır.

Yapılan hatalı tahminler I. ve II. Tip Hata olarak ikiye ayrılmakta ve değerlendirilen veri setinin durumuna göre bir tanesi daha kritik hata sayılabilmektedir. Kanserli hasta tahminin yapıldığı bir veri setinde kanser hastasına kanser değildir tahmini yapmak II. Tip, sağlıklı bir kişiye kanser tahmini yapmak ise I. Tip hataya girmektedir. Fakat burada II. Tip hatayı yapmış olmak kanserli bir hastayı sağlıklı olarak tahmin ettiğinden kritik bir hata sayılmaktadır ve analizcilerin bu hatadan daha çok kaçınması gerekecektir (Portilla, 2020).

2.3.1.1. Doğruluk

Doğruluk değerlendirmesi en temel ve basit değerlendirme ölçütüdür. Doğru tahmin edilen değerlerin sayısının tüm tahminlerin sayısına bölünmesiyle doğruluk değerine ulaşılır (Nizam ve Akın, 2014: 132-133; Sokolova ve Lapalme, 2009: 430). Örneğin 100 tane gözlem verisinden 70 tane doğru sınıfsal tahmin yapıldıysa bu modelin doğruluk oranı %70'tir.

$$\text{DOĞRULUK} = \frac{\text{DOĞRU TAHMİN SAYISI (DP + DN)}}{\text{GENEL TAHMİN SAYISI}}$$

Doğruluk modeli sınıf sayıları arasında eşit dağılım olduğu takdirde faydalıdır. Sınıfların oranı arasında büyük eşitsizlikler bulunduğu takdirde doğruluk değerlendirme ölçütünü kullanmak mantıklı değildir. Örneğin 100 tane gözlem verisinin olması gereken çıktıların 90 tanesi kanser değildir, 10 tanesi kanser ise ve model 90 tane doğru tahmin yaptıysa bu modele başarılı demek doğruluk değerlendirmesiyle pek mümkün değildir. Çünkü model içerisine sokulan tüm değerleri kanser değildir olarak değerlendiriyor olsa ve 100 tahminin hepsine de kanser değildir çıktısı verseydi yine %90 doğruluk oranına erişmiş olacaktı (Portilla, 2020).

2.3.1.2. Geri Çağırma

Geri çağırma değerlendirme ölçütü, doğru pozitif sayısının toplam gerçek pozitif sayısına oranıdır. Toplam gerçek pozitif sayısı, doğru pozitiflerin yanlış negatiflerle toplanması ile elde edilir (Davis ve Goadrich, 2006: 235; Nizam ve Akın, 2014: 133; Sokolova ve Lapalme, 2009: 430). Pozitif değerlerin yüzde kaçının doğru tahmin edildiğini göstermektedir.

Örneğin şeker hastalarının tahmin edilmeye çalışıldığı bir veri setinde geri çağırma değeri, şeker hastası olan kişilerin yüzde kaçının doğru tahmin edildiğini göstermektedir.

$$\text{GERİ ÇAĞIRMA} = \frac{\text{DOĞRU POZİTİF SAYISI (DP)}}{\text{TOPLAM GERÇEK POZİTİF SAYISI (DP + YN)}}$$

2.3.1.3. Hassasiyet

Hassasiyet değerlendirme ölçütü, doğru tahmin edilen pozitif sayısının toplam pozitif tahmin sayısına oranıdır (Davis ve Goadrich, 2006: 233-235; Nizam ve Akın, 2014: 133; Sokolova ve Lapalme, 2009: 430). Hassasiyet, sınıflandırma modelinin gerçekten pozitif olan pozitif tahminlerinin toplam pozitif tahminine oranını vermektedir. Yani pozitif tahminlerin yüzde kaçının doğru çıktığını göstermektedir. Yine şeker hastası tahminini düşünüldüğünde burada hassasiyet şeker hastası olduğu tahmin edilen kişilerin yüzde kaçının gerçekten hasta olduğunu göstermektedir.

$$\text{HASSASİYET} = \frac{\text{DOĞRU POZİTİF SAYISI (DP)}}{\text{TOPLAM POZİTİF TAHMİN SAYISI (DP + YP)}}$$

2.3.1.4. F1 Değeri

F1 değeri, hassasiyet ve geri çağırma ölçütlerinin harmonik ortalaması alınarak elde edilen bir sınıflandırma modeli değerlendirme ölçütüdür. Bu ölçüt bir sınıfın az sayıda olduğu, oran olarak dengesiz dağılımlarda yaygın olarak kullanılmaktadır (Lipton ve diğerleri, 2014: 225-227).

$$\text{HARMONİK ORTALAMA} = \frac{n (\text{ELEMEN SAYISI})}{\sum_{i=1}^n \frac{1}{x_i}}$$

$$F1 = \frac{2}{\frac{1}{\text{HASSASIYET}} + \frac{1}{\text{GERİ ÇAĞIRMA}}} = 2 \times \frac{\text{HASSASIYET} \times \text{GERİ ÇAĞIRMA}}{\text{HASSASIYET} + \text{GERİ ÇAĞIRMA}}$$

2.3.2. Regresyon Problemleri Değerlendirme Ölçütleri

Regresyon modelleri değerlendirme ölçütleri sayısal tahminlerde bulunan modellerin çıktılarının değerlendirilmesi amacıyla kullanılan ölçütlerdir. Bu kısımda yalnızca temel anlamda kullanılan 3 ölçüte yer verilmiştir. Formüllerde yazılı olan h_i değeri, tahmin edilen değer ile gerçek değer arasındaki farkı yani hatayı temsil etmektedir.

2.3.2.1. Ortalama Mutlak Hata (OMH)

OMH, model tarafından tahmin edilen değerlerin gerçek değerlerle aralarındaki farkların mutlak değerlerinin aritmetik ortalamasını vermektedir. OMH formülü aşağıda belirtilmiştir (Willmot ve Matsuura, 2005: 79-80; Chai ve Draxler, 2014: 1248).

$$n^{-1} \sum_{i=1}^n |h_i|$$

2.3.2.2. Ortalama Karesel Hata (OKH)

OKH, model tarafından tahmin edilen değerlerin gerçek değerlerle arasındaki farkların karelerinin aritmetik ortalamasını vermektedir. OKH'da hataların karesi alındığından dolayı OMH'e göre yanlış tahmin değerleri daha yüksek oranda ölçüte yansıtılmaktadır. OKH formülü aşağıda belirtildiği gibidir (Taşar ve diğerleri, 2018: 546).

$$n^{-1} \sum_{i=1}^n h_i^2$$

2.3.2.3. Kök Ortalama Karesel Hata (KOKH)

KOKH, OKH'nin kökünün alınmasıyla elde edilmektedir. KOKH kök işlemi öncesinde farklı ölçekte olduğundan anlamlandırması zor olan değerleri veri seti ölçeğine indirerek daha anlaşılır hale getirmektedir. Formülü aşağıda belirtildiği gibidir (Willmot ve Matsuura, 2005: 79-80; Chai ve Draxler, 2014: 1248).

$$\sqrt{n^{-1} \sum_{i=1}^n h_i^2}$$

2.4. DENETİMLİ ÖĞRENME MODELLERİ

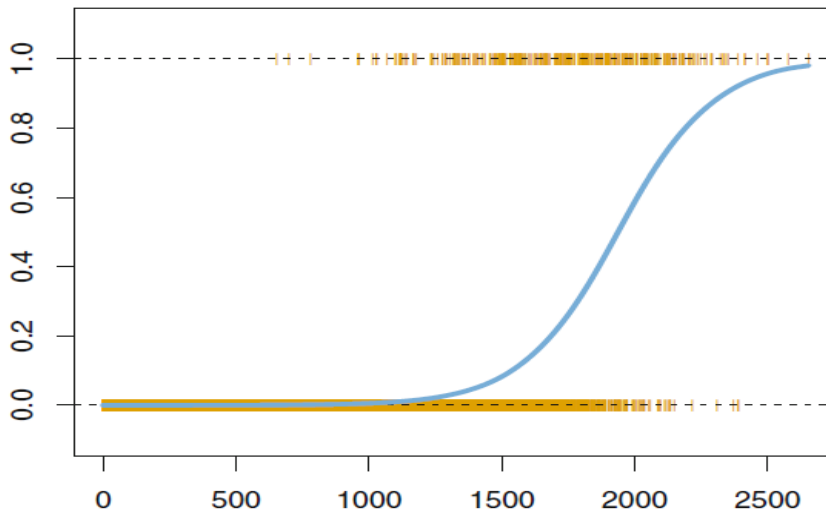
2.4.1. Sınıflandırma Problemleri Modelleri

Sınıflandırma problemlerinin çözümünde kullanılan birçok model bulunmakla birlikte bu kısımda yalnızca tezde kullanılan modellere yer verilmiştir.

2.4.1.1. Lojistik Regresyon Modeli

Sayısal çıktıların tahminlerinde doğrusal regresyon kullanılabilmektedir. Ancak sınıfsal çıktılarda doğrusal regresyon kullanımının bazı sakıncaları bulunmaktadır. Bundan dolayı sınıfsal tahminler için lojistik regresyon geliştirilmiştir. Lojistik regresyonda çıktı değeri direkt olarak modellenmektense, çıktının bir sınıfa ait olma ihtimali modellenmektedir. Verilen niteliklere göre bir sınıfa ait olma ihtimali lojistik fonksiyonu kullanılarak 0-1 arasında bir değer almaktadır. Sonrasında ihtimal eşiğine karar verilmektedir. Örneğin 0.5 seçildiyse, A sınıfına ait olma ihtimali 0.5 üzerinde olanlar A sınıfında, altında olanlar B sınıfındadır şeklinde sınıfsal analiz yapılmış olmaktadır. Doğrusal regresyondaki doğrunun yerini Şekil 3'te gösterilen S şekilli eğri almaktadır (James ve diğerleri, 2013: 128-132).

Şekil 3: Lojistik Regresyon



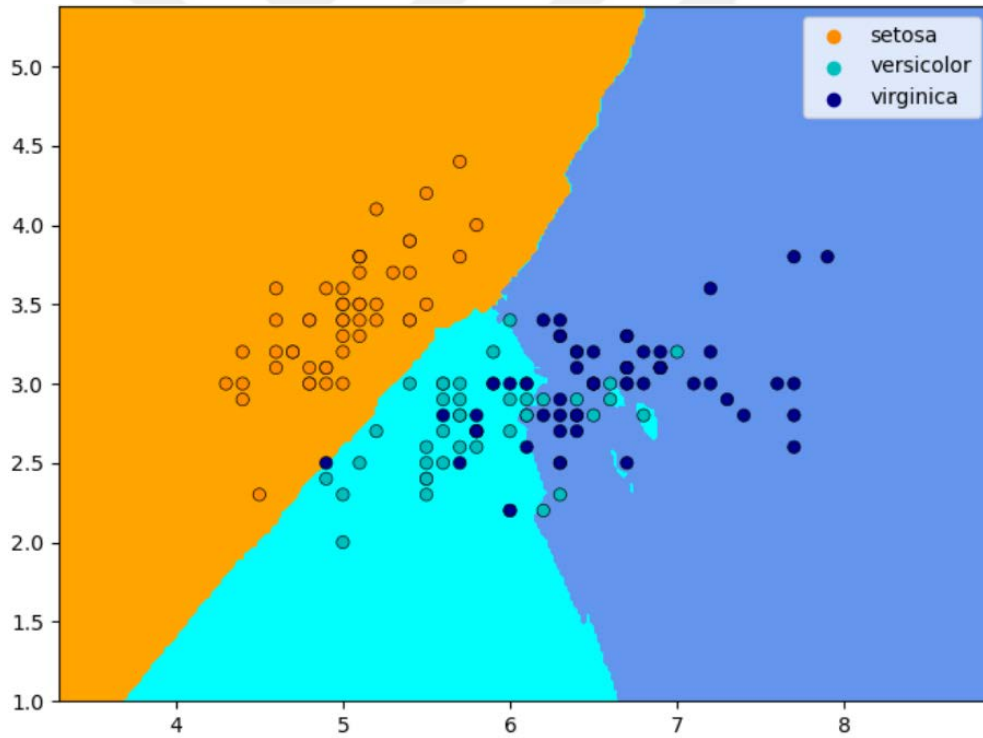
Kaynak: James ve diğerleri, 2013: 131

2.4.1.2. K En Yakın Komşular (KEYK) Modeli

KEYK Modeli, 1966 yılında T.M.COVER ve P.E.HART tarafından bulunan “En Yakın Komşu Karar Kuralı” kullanılarak oluşturulmuştur. Bu kural sınıflandırılmamış bir örneklemin daha önceden sınıflandırılmış bulunan en yakın veri kümesiyle aynı sınıfta sayılması önerisinde bulunmaktadır (Cover ve Hart, 1967: 21).

KEYK modeli makine öğrenmesinde, tahmin için girilen yeni değerleri eğitim setindeki en yakın komşularını bularak bir sınıflandırmaya tabi tutmaktadır. Birbirine yakın olan veri noktaları birbirleriyle aynı sınıfta yer alma temayülündedir. Şekil 4’te KEYK kullanılarak tasnif edilmiş 3 farklı bitki türünün makine öğrenmesi tarafından gruplandırılması ve gerçek türleri gösterilmektedir.

Şekil 4: KEYK Gruplandırması



Kaynak: Scikit-learn, 2020a

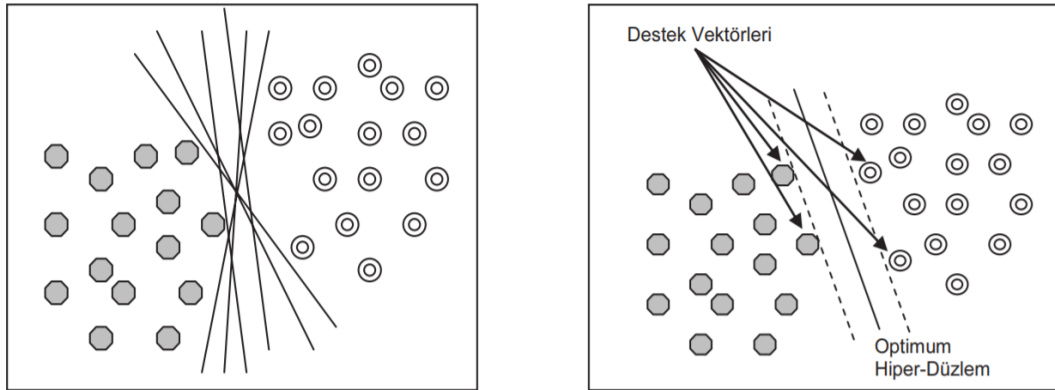
2.4.1.3. Destek Vektör Makineleri (DVM) Modeli

DVM'nin farklı öncüleri bulunmakla birlikte 1995 ve 1998 yıllarında Vladimir Naumovich Vapnik'in DVM'ne özel önem veren kitapları yayımlandıktan sonra metod ayrı bir ivme kazanmıştır (Steinwart ve Christmann, 2008: VII). DVM hem sınıflandırma hem de regresyon problemlerinde kullanılabilmektedir.

DVM literatürde sınıflandırma problemlerinde kullanılan modellerinin çoğuna göre daha başarılı ve etkin sonuçlar veren bir model olarak yer almaktadır. DVM sınıflandırma yapabilmek için sınıfları birbirinden ayıran optimum hiper düzlemi belirlemektedir (Ayhan ve Erdoğan, 2014: 175-178).

Daha basit ifade etmek gerekirse sınıfları birbirinden ayıran en uygun karar düzlemi bulunmaya çalışılmaktadır. Böylece yeni gelen veri noktaları düzleme olan konumlarına bakılarak sınıflandırılabilir. Şekil 5'in sol tarafında sınıfları birbirinden ayıran birçok farklı hiper düzlem çizilebileceği, optimum hiper düzlemin ise sınıflara ait en yakın noktaların kendisine maksimum uzaklıkta yer aldığı sağ taraftaki düzlem olduğu gösterilmektedir. Optimum hiper düzlem ve veri noktaları arasındaki sınırı çizen vektörlere ise destek vektörleri denmektedir (Kavzaoğlu ve Çölkesen, 2010: 75-76).

Şekil 5: DVM Hiper Düzlemi

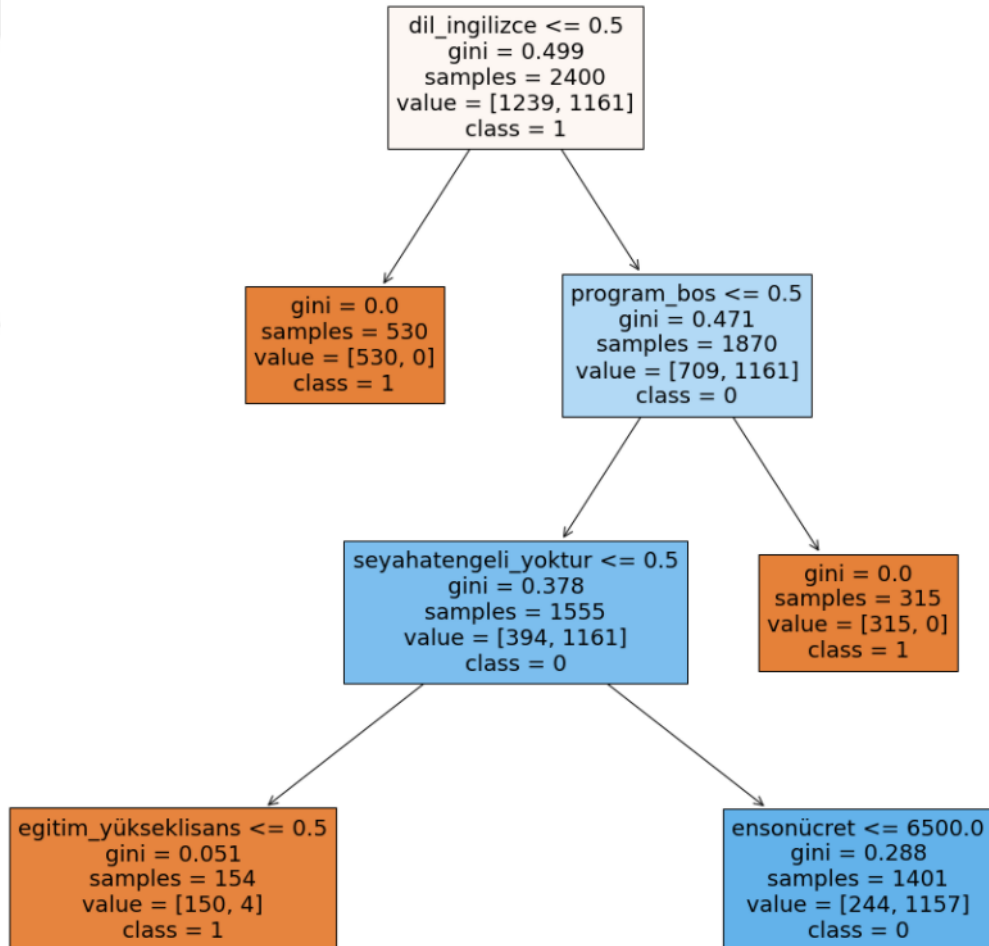


Kaynak: Kavzaoğlu ve Çölkesen, 2010:75-76

2.4.1.4. Karar Ağaçları Modeli

Karar Ağaçları yüksek doğruluk oranı ve basit çalışma mantığıyla denetimli öğrenme problemlerinde yaygın olarak kullanılan başarılı bir tekniktir. Karar Ağaçları soru sorma ve cevaba göre dallanma şekilde çalışmaktadır. Modelimizin diyagramı ters duran bir ağaç şeklindedir. Şekil 6'da görüleceği üzere en üstte kök soru aşağıya doğru da ağaç diyagramı oluşturan dallanan sorular bulunmaktadır. Amaç iki cevaplı sorularla ulaşılmak istenen çıktı sınıflarına erişmektir (Elouedi ve diğerleri, 2000: 80-81).

Şekil 6: Karar Ağaçları Diyagramı

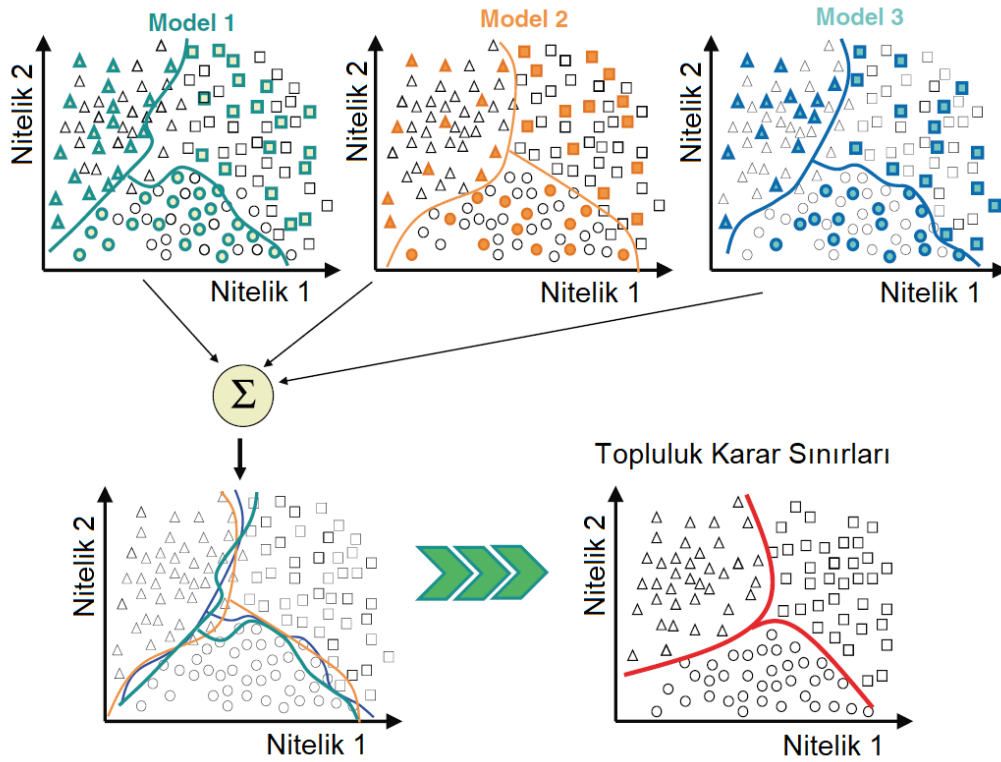


Kaynak: Yazar Tarafından Derlenmiştir.

2.4.1.5. Topluluk Öğrenim Metotları

Standart makine öğrenmesi modelleri bilinmeyeni tanımlamak için olası fonksiyonlar (hipotez) veya karar ağaçları uzayından en iyi tahmini gerçekleştirecek fonksiyonu, ağacı bulmaya çalışmaktadır. Topluluk öğrenme algoritmaları ise veriyi anlamlandırmak için bir hipotez bulmaktansa, birçok hipotezin yer aldığı bir küme (topluluk) inşa etmekte ve sonrasında yeni bir veri için her bir hipotezin belirli ağırlıklarda tahmin yapmasıyla sonuca ulaşmaktadır (Dietterich, 2002: 405-406).

Şekil 7: Topluluk Öğrenimi Örneği



Kaynak: Polikar, 2012:3

Topluluk öğrenim metotlarında iki temel algoritmaya değinilecektir. Bunlar torbalama ve artırım algoritmalarıdır. Leo Breiman tarafından 1994 yılında geliştirilen torbalama algoritması topluluk öğrenim metotlarının en temel ve basit algoritmalarından birisi olmakla birlikte gayet etkilidir. Torbalama bir eğitim veri setinde birbirinden bağımsız sınıflandırıcıları değişimli olarak, eğitim setinden alının

örneklerle eğitir. Her bir eğitim kümesi genel veri setinden rastgele seçimle belirlenmektedir. Daha sonra eğitilmiş sınıflandırıcılar basit çoğunluk oylamasıyla birleştirilir. Örneğin bir tahmin yapılacak iken tüm sınıflandırıcıların tahminleri bir oy kabul edilerek sonuca ulaşılır (Polikar, 2012:12).

Artırım, düşük tahmin başarısına sahip zayıf sınıflandırıcılar topluluğundan düşük tahmin hatasına sahip güçlü bir sınıflandırıcı elde eden bir yaklaşımdır. Torbalama gibi basit çoğunluk oylamasıyla çalışan artırım önemli bir farklılığa sahiptir. Torbalamada her eğitim kümesi genel veri setinden rastgele seçilmekteydi. Artırımda ise seçimlerde önceki sınıflandırıcılarda yanlış tahmin edilmiş veri setlerine odaklanılmaktadır (Polikar, 2012:13). Ekstrem Gradyan Artırım (XGBoost) modeli artırım topluluk öğrenimi metodunu kullanmaktadır.

2.4.1.5.1. Rastgele Orman Modeli

Rastgele Orman modeli adından da anlaşılacağı üzere aslında Karar Ağaçları modelinden türetilmiştir. Karar Ağaçlarında tek bir ağaç yapısı ile bir model ortaya konmaktaydı. Rastgele Orman modelinde ise topluluk öğrenimi metodu olan torbalama yöntemiyle karar ağaçlarından bir orman oluşturulmakta ve tahminler ormanda bulunan ağaçların oylarıyla belirlenmektedir (Breiman, 2001: 5-6).

2.4.1.5.2. XGBoost Modeli

Makine öğrenmesinde ağaç artırımı tahmin başarısını yüksek oranlarda artırabilmektedir. Bu nedenle yaygın bir makine öğrenmesi metodudur. XGBoost da ölçeklendirilebilir uçtan uca ağaç artırımı gerçekleştiren bir algoritmadır. XGBoost, Kaggle üzerinde düzenlenen makine öğrenmesi yarışmalarında yüksek başarı ve kullanım oranıyla kendisini kanıtlamış bir sistemdir (Chen ve Guestrin, 2016: 785). Modelde her bir ağacın hatalarından öğrenerek yeni bir ağaç yapısı kurulmakta ve sonuçta en gelişmiş ağaç yapısıyla tahmin gerçekleştirilmektedir.

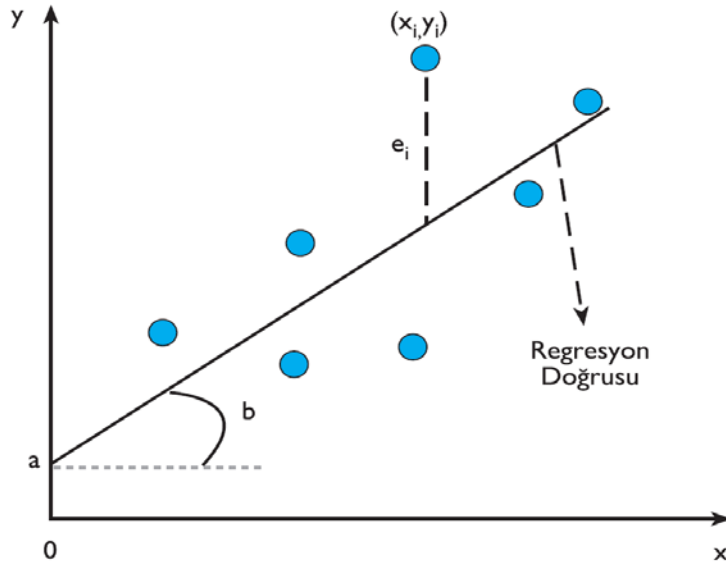
2.4.2. Regresyon Problemleri Modeli

Regresyon problemlerinin çözümünde kullanılan birçok farklı model bulunmaktadır. Burada yalnızca tezin ilerleyen çalışmalarında kullanılması planlı olan doğrusal regresyon modeli açıklanmıştır.

İstatistikte iki değişken arasındaki ilişkiye korelasyon, bir ya da birden fazla değişkenin başka bir değişken üzerinde oluşturduğu etkiyi tarif etmeye regresyon analizi denmektedir. Değişkenler arasındaki ilişki eğer doğrusal nitelikteyse buna doğrusal regresyon analizi denmektedir. Aşağıda belirtilen doğrusal regresyon formülünde i kaçınıcı gözlem olduğunu belirtmek üzere, y_i bağımlı değişken y 'nin değerini, x_i bağımsız değişken x 'in değerini, ε_i rassal hatayı, α $x=0$ olduğunda y 'nin değerini, β x 'teki bir birim değişimin y 'de oluşturacağı oransal değişimi temsil etmektedir (Sönmez, 2018:227-234). Doğrusal regresyon modeli nitelikler ve çıktı değeri arasında burada açıklandığı şekilde formül geliştirilerek yeni gelen veriler üzerinde tahminlerini gerçekleştirmektedir.

$$y_i = \alpha + \beta x_i + \varepsilon_i$$

Şekil 8: Doğrusal Regresyon Grafiği



Kaynak: Sönmez, 2018:234.

ÜÇÜNCÜ BÖLÜM

VERİ, ARAŞTIRMA YÖNTEMİ VE KISITLAR

3.1. VERİ

İş başvuruları esnasında beyan edilen veriler çoğunluk itibarıyla Kişisel Verilerin Korunması Kanunu kapsamında koruma altına alınan kişisel verilerdir. Bahse konu verilerin işlenebilmesi için özel şartlar haricinde kişilerin açık rızası olması zorunludur (Türkiye Büyük Millet Meclisi, 2016).

Yapay zeka sisteminin eğitilebilmesi için ise zengin içerikli iş başvurusu verilerine ihtiyaç duyulmaktadır. Gerçek verilerin temininin kanunlar ve şirket kuralları kapsamında gerçekleşmemesi nedeniyle tezde kullanılan veriler makine öğrenmesi sisteminin eğitilebilmesi amacıyla, Python programlama dili yardımıyla yazar tarafından oluşturulmuştur. Burada önemli olan husus sistemin verilerle eğitildikten sonra beklenen seviyede öğrenmeyi gerçekleştirip doğru çıktıları verebilmesidir.

3.1.1. Verinin Hazırlanması

Makine öğrenmesinde model oluşturabilmek ve bilgisayarı eğitebilmek için yüksek sayıda veriye ihtiyaç duyulmaktadır. Bu kapsamda Yapay Zeka Tabanlı Personel Seçim Sistemi Uygulamasının geliştirilebilmesi için 3000 kişilik yapay bir veri seti oluşturulmuştur.

Veriler hazırlanırken gerçek hayata paralel, normalde de olması muhtemel veriler oluşturulabilmesi için üstün gayret gösterilmiştir. Temel bir örnekle açıklamak gerekirse; 1997 doğum tarihli bir kişinin doktora mezunu olması hayatın genel akışına aykırı bir veri olacaktır. Bunun gibi, veriler doldurulurken uyumsuz, olağan dışı veri girişi olmayacak şekilde ayarlama yapılmıştır.

Veri setimiz 3000 Satır ve 25 sütundan oluşmaktadır. 24 sütunluk bölüm nitelikleri, sondaki “Karar” sütunu ise çıktıyı temsil etmektedir. Verilerin hepsi Python programlama diliyle kodlanarak üretilmiştir. Yazılım dili ara yüzü olarak Jupyter Notebook kullanılmıştır (Jupyter, 2020).

Veri setinin oluşturulması aşamasında temel olarak Pandas kütüphanesi kullanılmıştır. Pandas, Python programlama dilinde yüksek performanslı, kullanımı kolay veri analiz araçları sağlayan açık kaynak bir kütüphanedir (McKinney ve diğerleri, 2020:1). Pandas ve diğer kütüphaneler kullanılabilmeleri için öncelikle Şekil 9’de görüleceği üzere kütüphane yüklemeleri yapılmıştır.

Şekil 9: Python Kütüphane Yüklemeleri

```
import pandas as pd
import numpy as np
import seaborn as sns
import random
import matplotlib.pyplot as plt
import datetime
```

Kaynak: Yazar tarafından derlenmiştir.

Bu bölümde görülen veri görsel analizleri için “Tableau Desktop” uygulaması kullanılmıştır. “Tableau Desktop” görsel analiz imkanlarının kolaylıkla gerçekleştirilebilmeleri amacıyla kullanılan bir iş zekası yazılımıdır (Tableau, 2020).

3.1.1.1. İsim, Soy İsim ve Cinsiyet

İsim ve soy isim verileri için örnek kadın, erkek isimleri (karşılarında cinsiyet bilgileri yer almaktadır) ve soy isimlerin bulunduğu CSV uzantılı iki adet dosya oluşturuldu. İsim ve soy isimler CSV dosyasından içe aktarılarak bir listenin içerisine saklanmıştır. Sonrasında rastgele seçimle 3000 kişilik aday listesi oluşturulmuştur. Yapılan işlemler Şekil 10’da görüldüğü gibidir.

Şekil 10: Aday Listesinin Oluşturulması

```
veri = pd.DataFrame()

isimler = pd.read_csv("isimler.csv")

soyisimler = pd.read_csv("soyisimler.csv")

sec = isimler.sample(3000, replace=True, random_state=seed)

veri = pd.DataFrame(sec[["İSİM", "CİNSİYET"]].values, columns=['isim', 'cinsiyet'])

veri["soyisim"] = soyisimler.sample(3000, replace=True, random_state=seed).values

veri = veri[["isim", "soyisim", "cinsiyet"]]

veri.head()
```

	isim	soyisim	cinsiyet
0	MEHMET		E
1	BUKET		K
2	ARDA		E
3	TURGAY		E
4	DENİZ		E

Kaynak: Yazar tarafından derlenmiştir.

3.1.1.2. Sütunların Oluşturulması

Veri tablomuza bilgi girişi yapılmadan önce 'doğumtarihi', 'yas', 'medenidurum', 'askerlikdurumu', 'ikametili', 'ikametilcesi', 'egitim', 'lisansbölümü', 'yükseklisansbölümü', 'doktorabölümü', 'programbilgisi', 'yabancıdildbilgisi', 'alantecrübesi', 'maaşbeklentisi', 'adlisicilkaydı', 'seyahatengeli', 'referans', 'kurs', 'ensonücret', 'vardiyaimkanı', 'yzbilgi', 'karar' isimlerinde içleri NAN (Sayı Değil) olarak doldurulmuş sütunlar eklenmiştir.

3.1.1.3. Yüzde Doldur Fonksiyonu

Python programlama dilinde fonksiyonların kullanımı çok önemlidir çünkü yapılan bir işlemin benzer durumlarda da kullanılabilmesini ve kodun düzenli olmasını sağlamaktadır. Bu nedenle veri üretiminde kullanılmak üzere, verilen parametreleri

sütuna istenen oranda dağıtabilen, Şekil 11'deki "yüzdedoldur" isimli fonksiyon yazılmıştır.

Şekil 11: Yüzde Doldur Fonksiyonu

```
def yüzdedoldur(df, sart, column, a, orana, b, oranb, c, oranc, d, orand, e, orane):
    df2 = df[sart]
    liste = []
    for i in range(round(len(df2) * orana)):
        liste.append(a)
    for i in range(round(len(df2) * oranb)):
        liste.append(b)
    for i in range(round(len(df2) * oranc)):
        liste.append(c)
    for i in range(round(len(df2) * orand)):
        liste.append(d)
    for i in range(round(len(df2) * orane)):
        liste.append(e)
    random.shuffle(liste)

    x = len(df2) - len(liste)
    if x > 0:
        for i in range(x):
            liste.append(a)
    elif x < 0:
        for i in range(abs(x)):
            liste.remove(a)

    df2[column] = pd.DataFrame(np.array(liste)).values
    for i in df2[column].index:
        df[column][i] = df2[column][i]
```

Kaynak: Yazar tarafından derlenmiştir.

3.1.1.4. Parametreler

Bir uygulama geliştirirken ufak dokunuşlarla değişiklikler yapabilmek gerekmektedir. Kodlamanın başına değişken parametreleri belirlenerek kodlama bittikten sonra detaylara inmeden değişiklik yapılabilmesi sağlanmaktadır. Bu nedenle veriler oluşturulurken kodun başına nitelikler için parametreler eklenmiştir. Örneğin Şekil 12'de lisans bölümleri sütununa ait parametreler görülmektedir.

Şekil 12: Parametreler Örneği

```
lisans_bölüm_1 = "Bilgisayar Mühendisliği"
lisans_bölüm_1_oran = 0.40
lisans_bölüm_2 = "Endüstri Mühendisliği"
lisans_bölüm_2_oran = 0.20
lisans_bölüm_3 = "İstatistik"
lisans_bölüm_3_oran = 0.20
lisans_bölüm_4 = "Matematik"
lisans_bölüm_4_oran = 0.10
lisans_bölüm_5 = "Matematik Mühendisliği"
lisans_bölüm_5_oran = 0.10
```

Kaynak: Yazar tarafından derlenmiştir.

3.1.1.5. Doğum Tarihleri

Adayların doğum tarihlerini üretebilmek için bir tarih üretme fonksiyonu yazılması gerekiyordu. Şekil 13'te belirtilen fonksiyon ile belirli aralıklarda rastgele günler üretildikten sonra bir liste içerisinde adaylara dağıtılarak doğum tarihleri oluşturulmuş oldu.

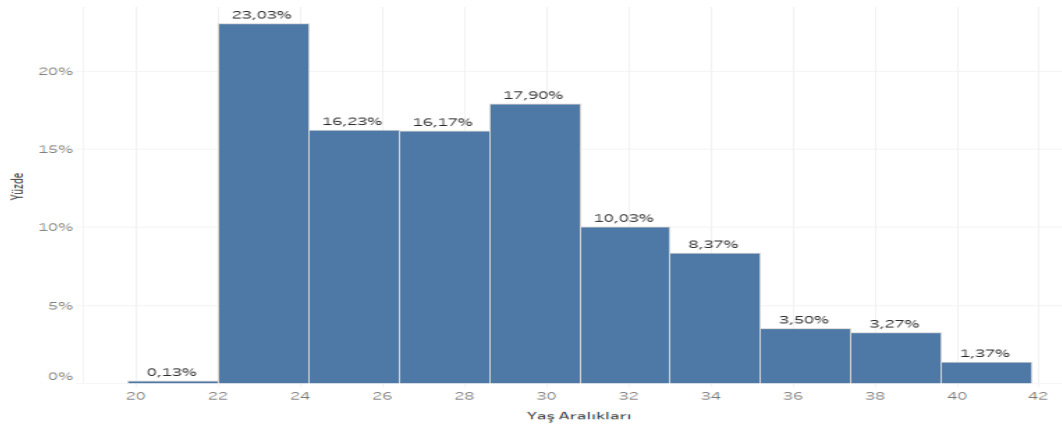
Doğum tarihleri sayısal nitelik arz etmediği için makine öğrenmesi modelinde kullanılamamaktadır. Bu nedenle doğum tarihine göre yaşı belirleyen bir fonksiyon daha yazılarak adayların yaşlarının bulunduğu bir sütun eklenmiştir.

Şekil 13: Doğum Tarihleri Üretimi

```
def tarihuret(bas, son, adet):  
    tarihliste = []  
    for i in range(adet):  
        tarihbaslangic = datetime.date(bas, 1, 1)  
        tarihson = datetime.date(son, 12, 30)  
        aralik = tarihson - tarihbaslangic  
        aralikkun = aralik.days  
        randomkun = random.randrange(aralikkun)  
        randomtarih = tarihbaslangic + datetime.timedelta(days=randomkun)  
        tarihliste.append(randomtarih)  
    return pd.array(tarihliste)  
  
dogum1 = pd.DataFrame(tarihuret(1990, 1998, 2100))  
dogum2 = pd.DataFrame(tarihuret(1985, 1990, 600))  
dogum3 = pd.DataFrame(tarihuret(1980, 1985, 300))  
  
dogum = dogum1.append(dogum2, ignore_index=True).append(dogum3,  
ignore_index=True)  
  
veri["dogumtarihi"] = dogum.sample(3000, random_state=seed)[0].values
```

Kaynak: Yazar tarafından derlenmiştir.

Şekil 14: Yaş Dağılımları

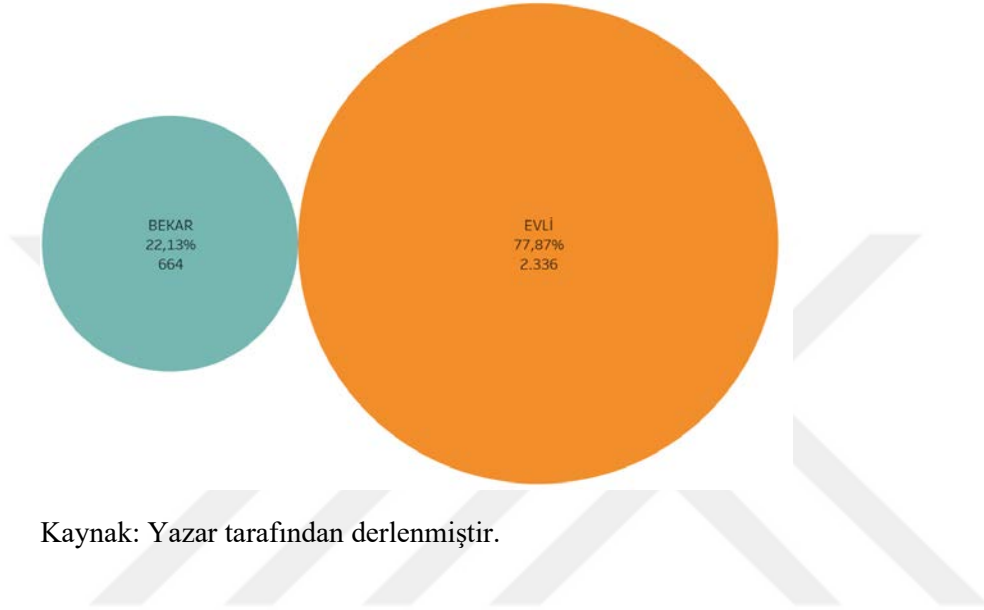


Kaynak: Yazar tarafından derlenmiştir.

3.1.1.6. Medeni Durum

Medeni durum sütunu evli/bekar olarak oluşturulmuştur. Veri seti dahilindeki medeni durum dağılımı toplam sayı ve yüzde cinsinden Şekil-15'te görüldüğü gibidir.

Şekil 15: Medeni Durum Dağılımı

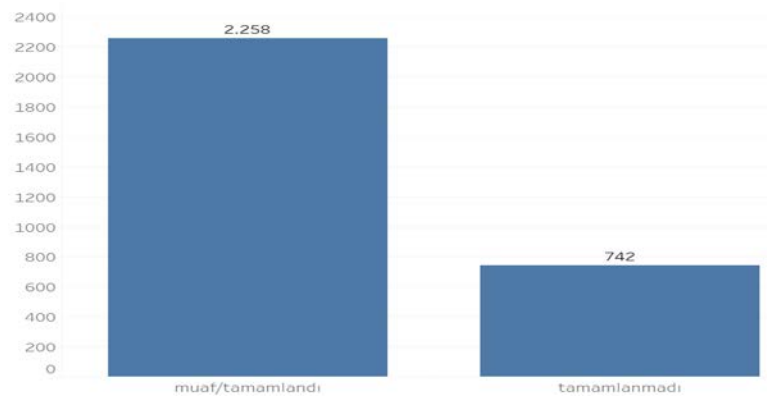


Kaynak: Yazar tarafından derlenmiştir.

3.1.1.7. Askerlik Durumu

Askerlik Durumu sütununda cinsiyeti kız olan veya 1985 öncesi doğumluların hepsi, geri kalan erkeklerin ise belirlenen parametreler dahilindeki oranı muaf/tamamlandı sayılmıştır.

Şekil 16: Askerlik Durumu Dağılımı



Kaynak: Yazar tarafından derlenmiştir.

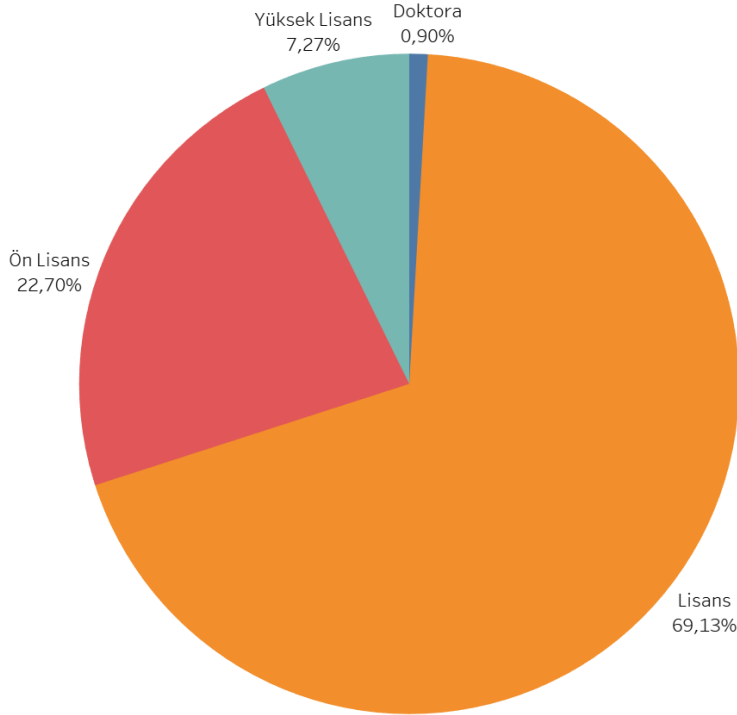
3.1.1.8. İkamet İli ve İlçesi

İkamet ili tüm adaylar için “İstanbul” olarak belirlenmiş, ikamet ilçesi ise TÜİK tarafından Adrese Dayalı Nüfus Kayıt Sistemine göre oluşturulan 2019 İl ve İlçelere göre nüfus verilerine göre oran dahilinde yüzededoldur fonksiyonu ile dağıtılmıştır (Türkiye İstatistik Kurumu, 2020a).

3.1.1.9. Eğitim Seviyesi

Eğitim seviyeleri yaş ve “Bitirilen Eğitim Durumuna ve Cinsiyete Göre Nüfus 2008-2019” TÜİK verilerine göre oluşturulmuştur (Türkiye İstatistik Kurumu, 2020b). Nüfustaki doktora, yüksek lisans ve lisans verilerine bağlı alınarak oluşturulan eğitim verilerinin dağılımı Şekil 17’de görüldüğü gibidir.

Şekil 17: Eğitim Seviyesi Dağılımı

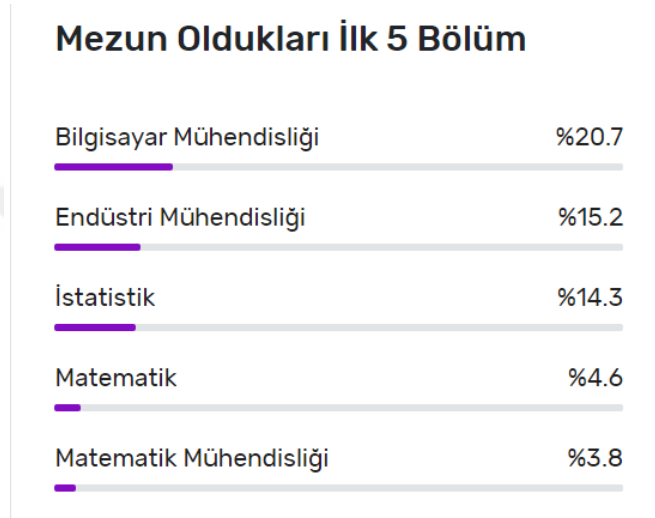


Kaynak: Yazar tarafından derlenmiştir.

3.1.1.10. Lisans, Yüksek Lisans ve Doktora Bölümleri

Eğitim bölümlerinin belirlenmesinde lisans düzeyi için, iş bulma platformu “kariyer.net” sitesinde yer alan Şekil 18’deki veri bilimci pozisyonunda çalışanların mezun oldukları ilk 5 bölüm seçilmiştir.

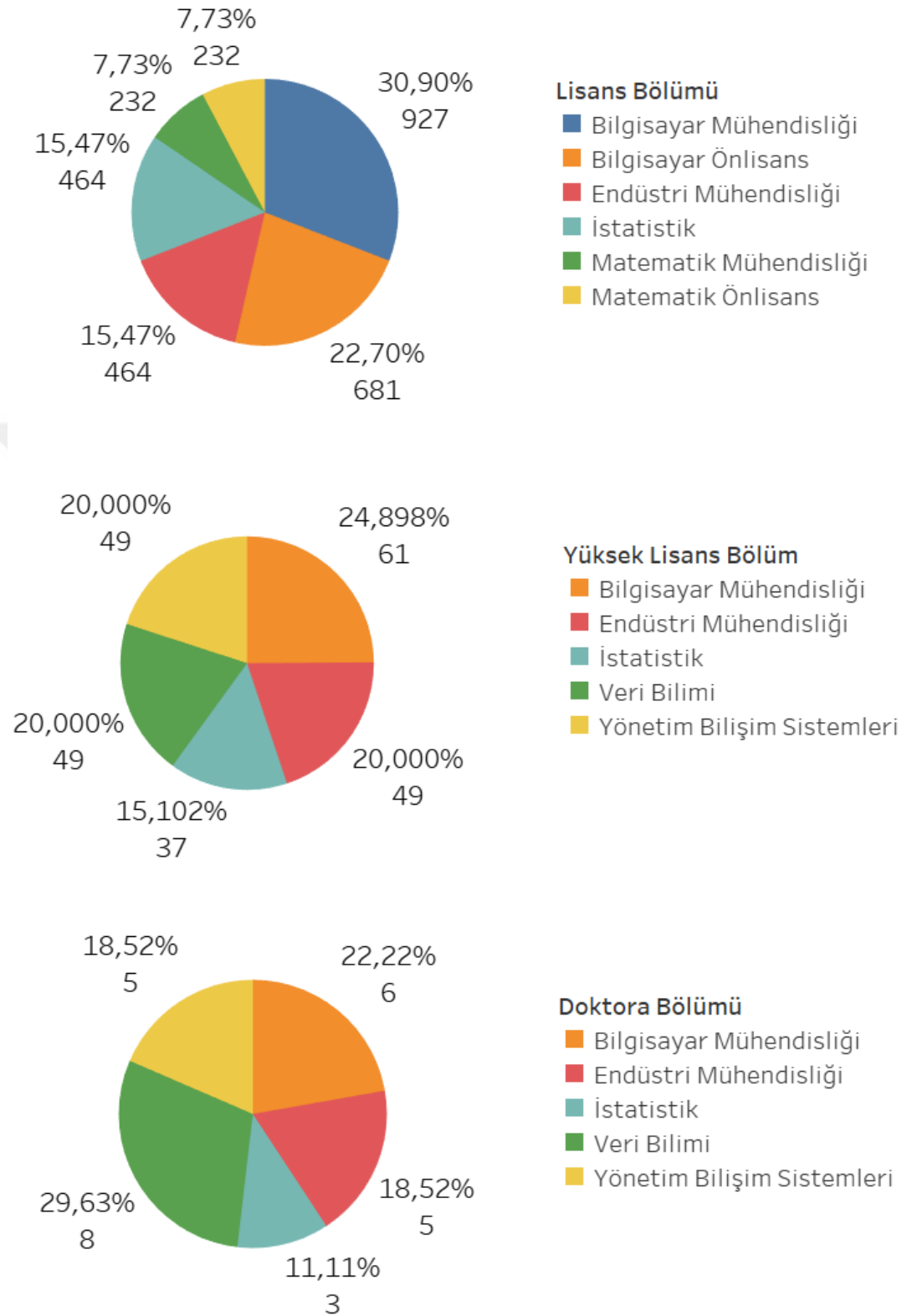
Şekil 18: Veri Bilimci İlk 5 Bölüm



Kaynak: (Kariyer.net, 2020)

Ön lisans mezunları lisans mezunu grubunda yer almış ve bilgisayar alanından mezun olanların hepsinin bölümü “Bilgisayar” olarak isimlendirilmiştir. Bölümlerin dağılımı Şekil 19’da gösterilmiştir.

Şekil 19: Bölüm Dağılımları

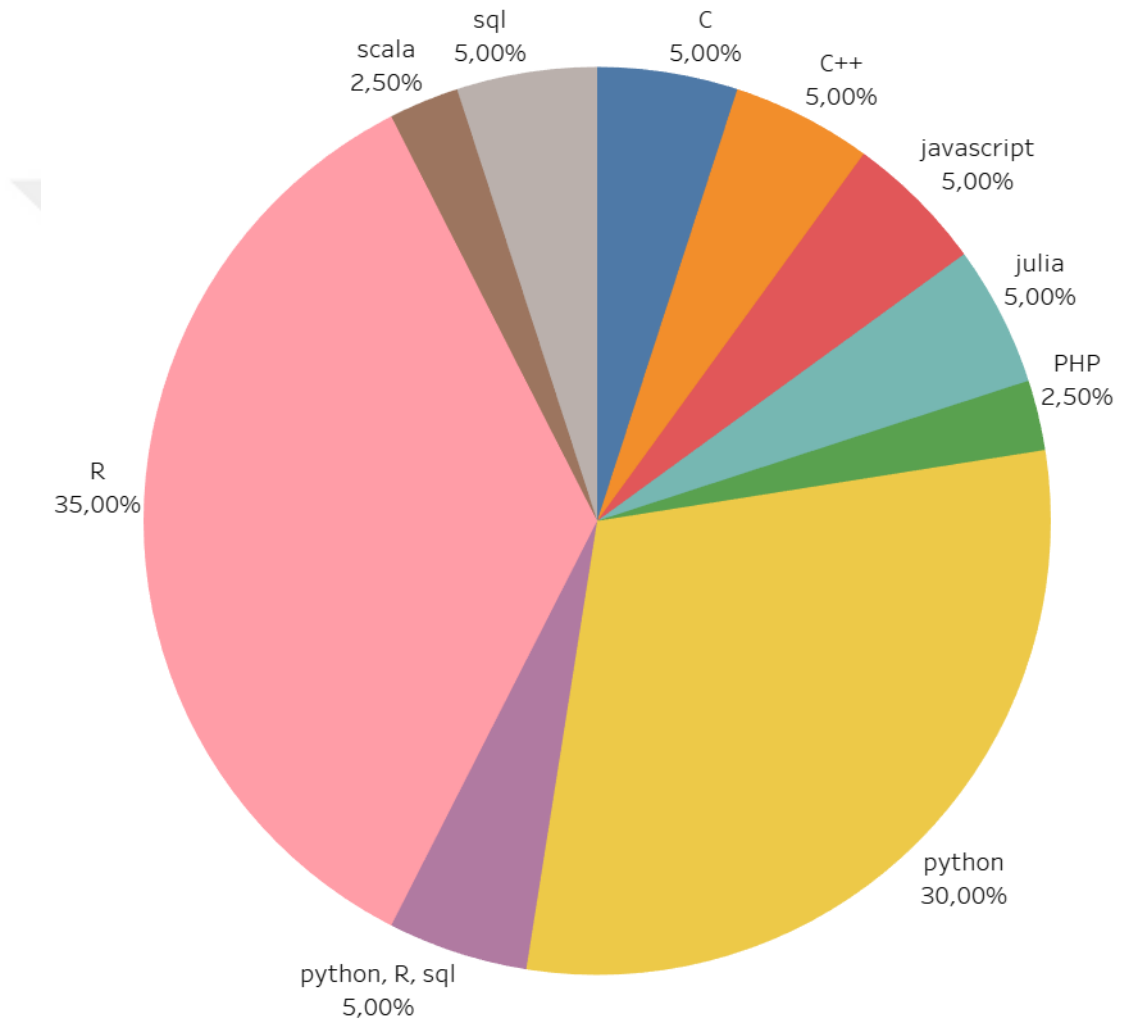


Kaynak: Yazar tarafından derlenmiştir.

3.1.1.11. Program Bilgisi

Veri bilimi uzmanları için açılan iş ilanlarının ortak özelliği Python veya R programlama dillerinden en az birisini bilme şartının aranmasıdır. Bu nedenle program bilgisi sütununda bahse konu programlama dilleri yüksek oranda dağıtılmıştır.

Şekil 20: Program Bilgisi Dağılımı

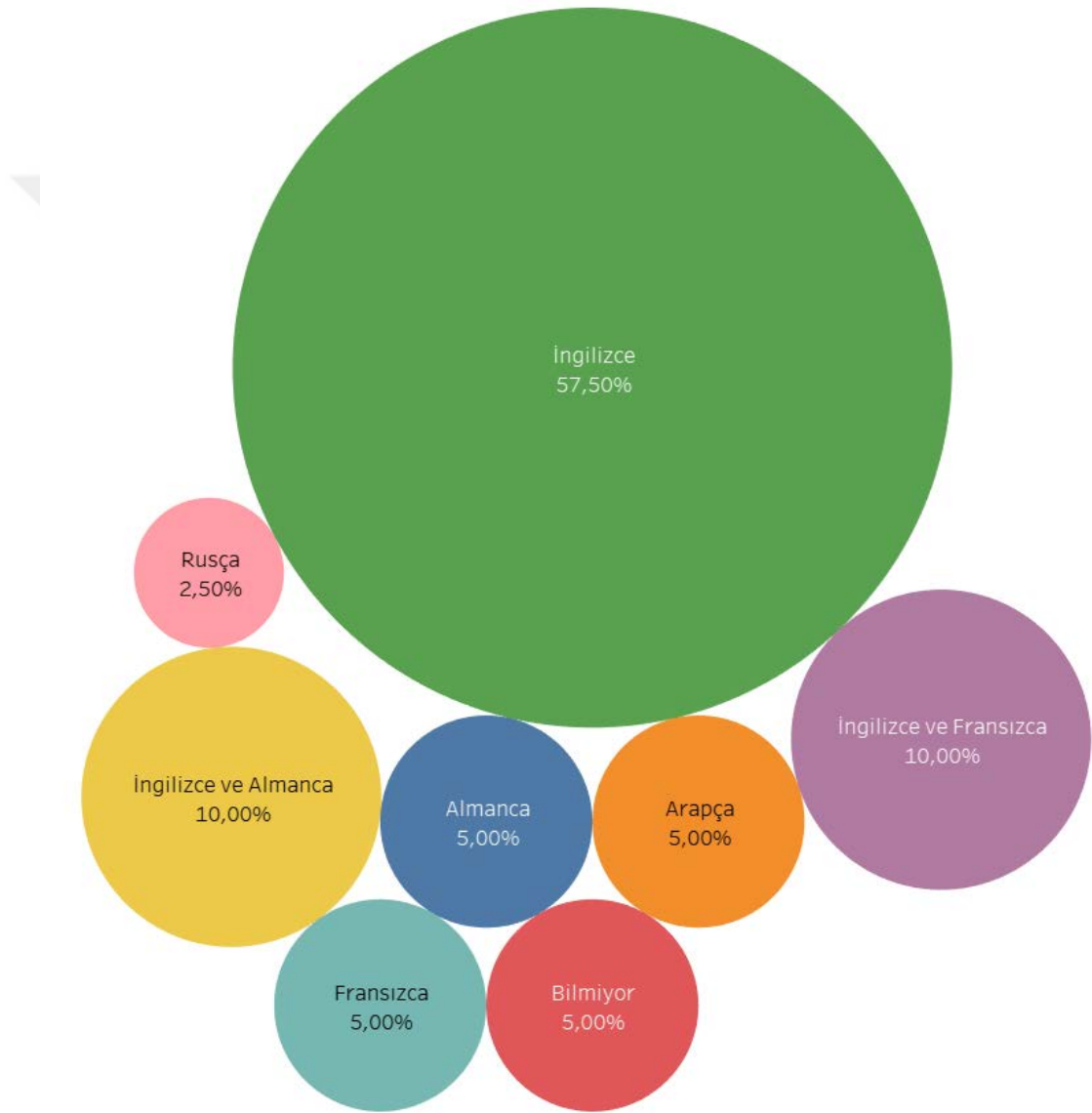


Kaynak: Yazar tarafından derlenmiştir.

3.1.1.12. Yabancı Dil Bilgisi

Bir veri bilimi uzmanı için olmazsa olmazlardan birisi İngilizce bilmesidir. Bundan dolayı adaylarda İngilizce bilen oranı yüksek tutulmuştur. Veri seti dahilinde yabancı dil bilgisi oranları Şekil 21’de görüldüğü gibidir.

Şekil 21: Yabancı Dil Bilgisi Oranları

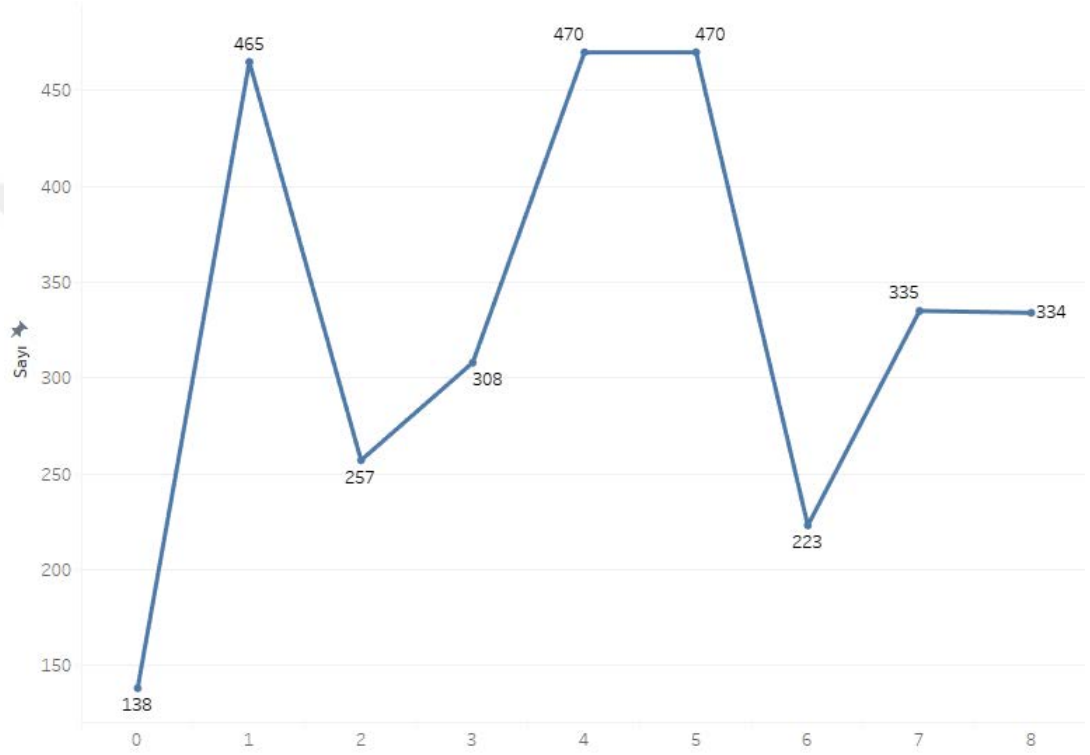


Kaynak: Yazar tarafından derlenmiştir.

3.1.1.13. İş Tecrübesi

Bilişim teknolojileri alanındaki iş tecrübesi 0-8 yıl arasında adayların yaşları hesaba katılarak oluşturulmuştur. Şekil 22’de iş tecrübesinin dağılımı sayılarla gösterilmektedir.

Şekil 22: İş Tecrübesi Dağılımı

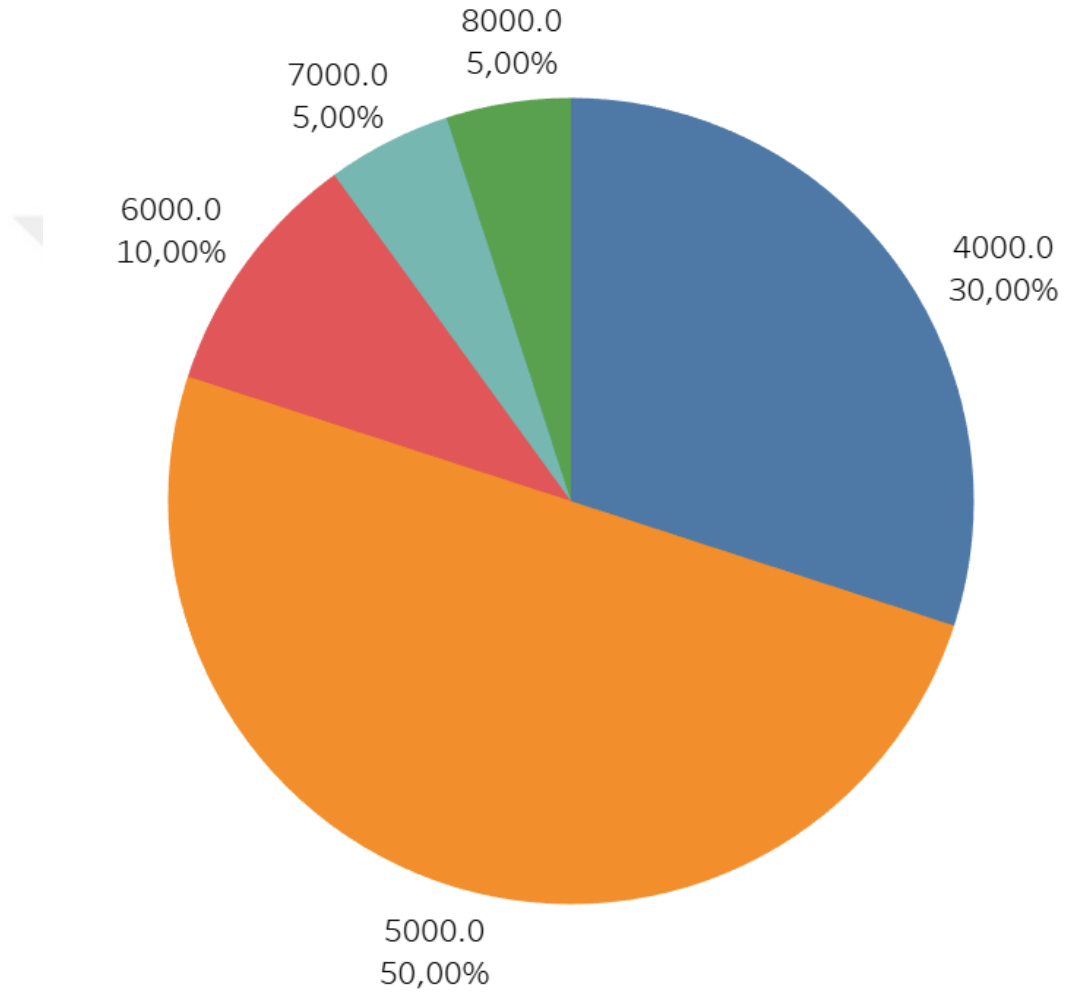


Kaynak: Yazar tarafından derlenmiştir.

3.1.1.14. Maaş Beklentisi

Maaş beklentisi işe alımda önemli bir kriter oluşturmaktadır. 4000-8000 TL aralığında Şekil 23'teki oranlarda dağıtımı yapılmıştır.

Şekil 23: Maaş Beklentisi Dağılımı



Kaynak: Yazar tarafından derlenmiştir.

3.1.1.15. Var-Yok Sütunları

Adli sicil kaydı, seyahate engel durum, kurum içi referans, kurs durumu ve vardiyalı çalışma durumu verilerinin dağılımı Şekil 24’te görüldüğü gibidir.

Şekil 24: Var-Yok Sütunları

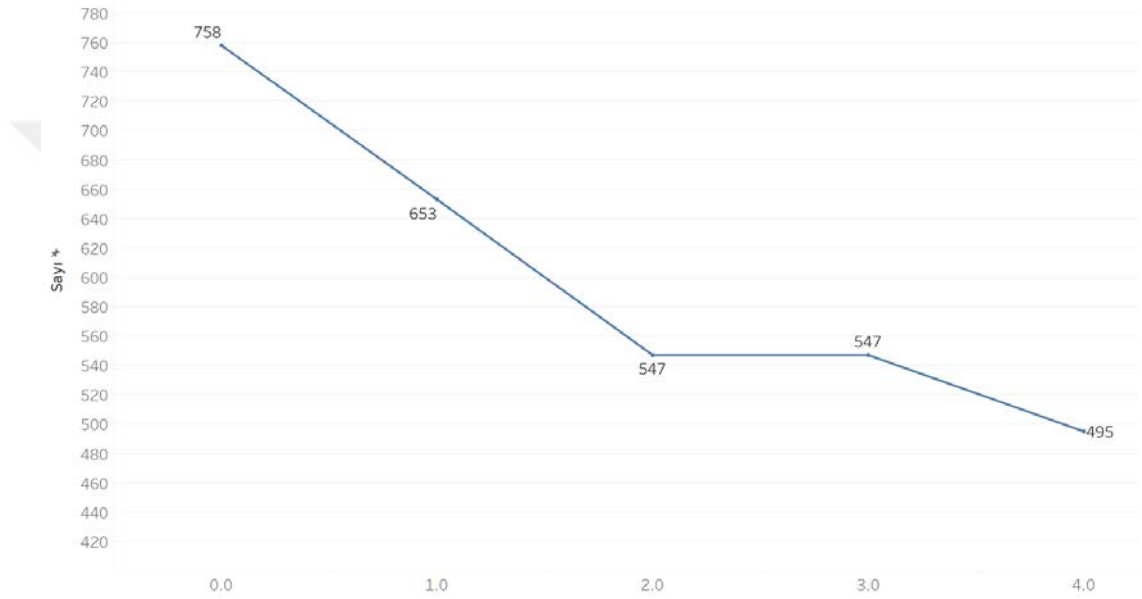


Kaynak: Yazar tarafından derlenmiştir.

3.1.1.16. Yapay Zeka Bilgisi

Yapay zeka veri bilimi uzmanının çalışma alanının önemli bir bölümünü oluşturmaktadır. Bu tezde bilginin gereklilikleri için detaylı açıklama verilmemiştir ancak puanlandırma için alt detaylar yayınlanacak iş ilanında bulunacaktır. Yapay zeka bilgisi 0-4 arasındaki değerlerde Şekil 25'te gösterilen sayılarda dağıtılmıştır.

Şekil 25: Yapay Zeka Bilgisi Dağılımı

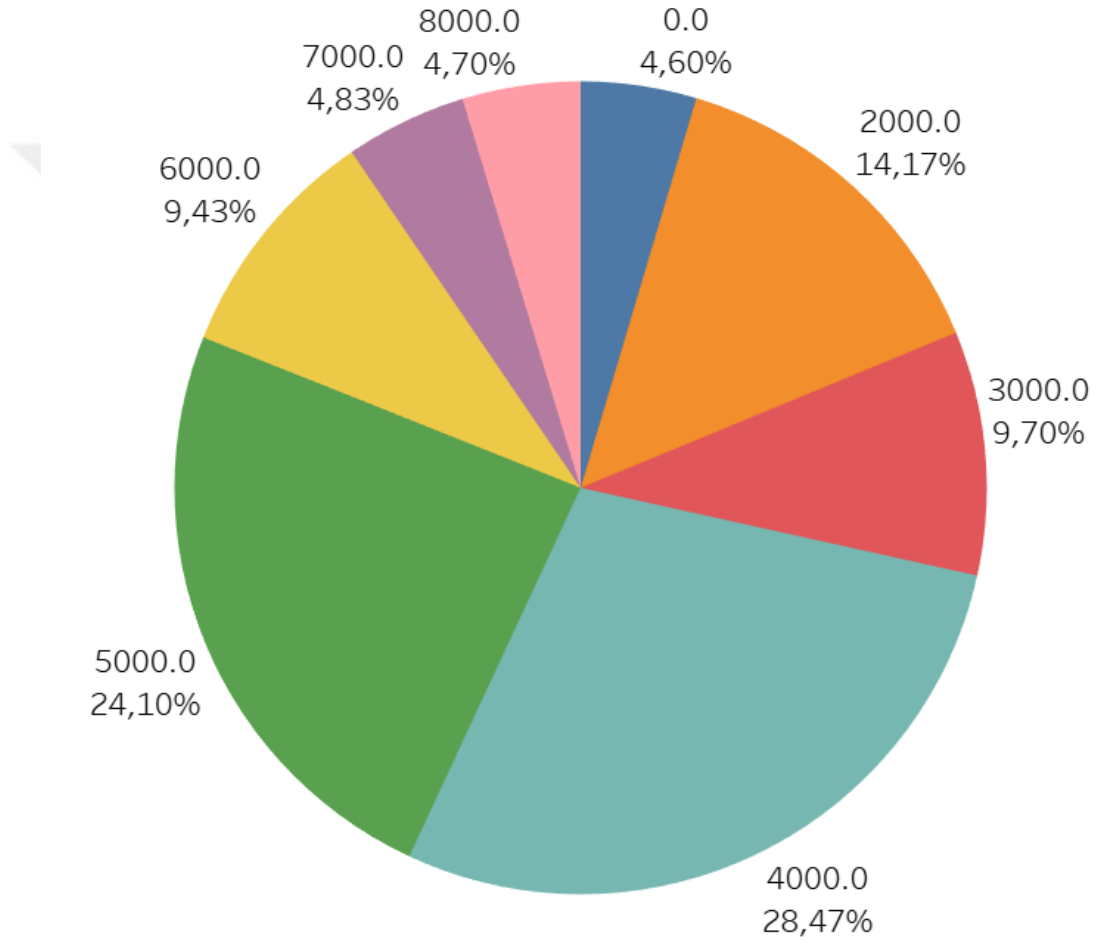


Kaynak: Yazar tarafından derlenmiştir.

3.1.1.17. En Son Görev Yapılan Yerdeki Ücret

En son çalışılan yerde alınan ücret iş başvuru formlarında yer alabilmekte ve uygulamada değerlendirme kriterleri arasında sayılmaktadır. Bu kapsamda Şekil 26’da görüldüğü gibi 4000-8000 TL aralığında belirtilen oranlarda dağıtılmıştır.

Şekil 26: Son Ücret Oranları



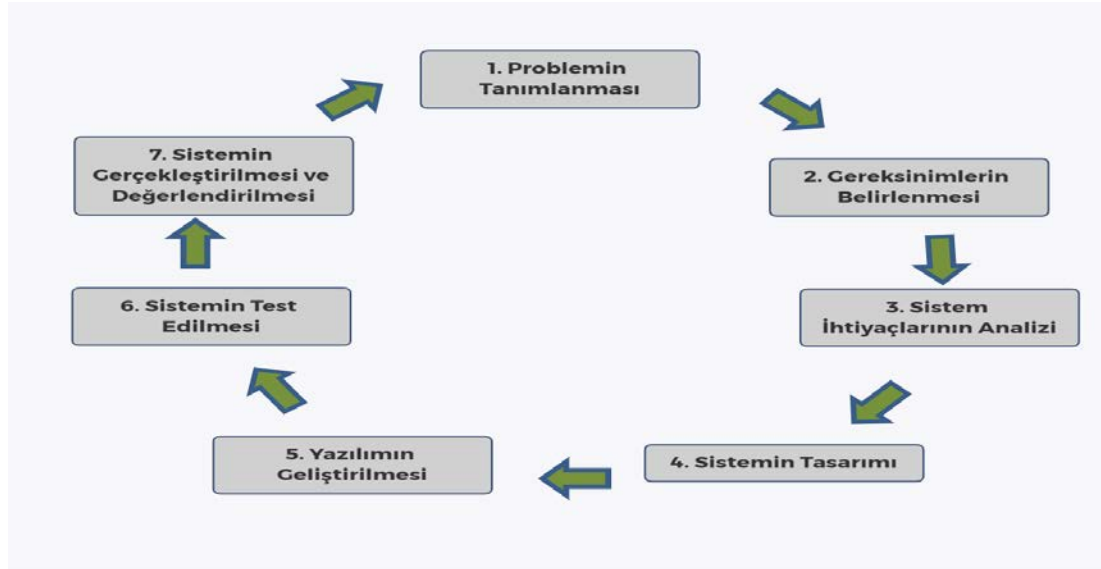
Kaynak: Yazar tarafından derlenmiştir.

3.2. ARAŞTIRMA YÖNTEMİ

Bu tezde araştırma yöntemi olarak Sistem Geliştirme Yaşam Döngüsü (SGYD) kullanılmıştır. Kurumlar bilişim sistemleri için büyük emek ve kaynaklar sarf etmektedirler. Bilişim sistemlerine artan bağımlılık güvenilir, emniyetli bilişim sistemi teknolojilerine ihtiyaç duyulmasını gerektirmektedir. SGYD bilişim sistemleri kurulurken doğru adımların takip edilebilmesi maksadıyla izlenecek sistematik bir yol haritası ihtiyacını karşılamaktadır (Department Of Justice, 2003).

SGYD standart kalıplar dahilinde sabit bir döngü olarak değil bir süreç olarak düşünülmelidir. Pek çok farklı tanımı ve adımları bulunan SGYD için genel bir tanım yapmak gerekirse; sistem geliştirme sürecinde en iyi ürünü ortaya koyabilmek maksadıyla bir döngü dahilinde takip edilen analiz ve dizayn maksatlı bir yaklaşım denilebilir. SGYD problem ve amaçların tanımlanması, bilgi gereksinimlerinin belirlenmesi, sistem ihtiyaçlarının analizi, önerilen sistemin tasarımı, yazılımın geliştirilmesi, sistemin test edilerek sürdürülmesi ve sistemin gerçekleştirilerek değerlendirilmesi safhalarından oluşmaktadır (Tecim, 2012).

Şekil 27: Sistem Geliştirme Yaşam Döngüsü



Kaynak: Tecim, 2012

SGYD'nin ilk adımı problemin ve amaçların tanımlanmasıdır. Bu safha sistemin temellerinin doğru atılması, amaçlar doğrultusunda bir yol izlenmesi için çok

mühimdir. Yapay Zeka Tabanlı Personel Seçim Sistemi Uygulamasında öncelikle problem ve amaç açık bir şekilde belirlenmiştir. Buradaki problem insan kaynağı temin süreçlerinde büyük aday havuzları, zengin nitelikler ve onları tasnif ihtiyacı dolayısıyla büyük zaman ve kaynak kayıpları yaşanabilmesidir. Bu net probleme karşılık yapay zeka tabanlı, zaman ve kaynak tasarrufu gerçekleştirebilecek bir ürün ortaya konması amaçlanmaktadır.

İkinci adım bilgi gereksinimlerinin belirlenmesidir. Burada sistemi kurmak için gerekli olan bilgi ve veri gereksinimleri ortaya konmalıdır. Personel Seçim Sistemi Uygulamasında gereksinimlerini karşılayabilmek maksadıyla yazar tarafından yapay zeka ve makine öğrenmesi alanında bilgi eksiklikleri giderilmeye çalışılmış ve mantıksal çerçevede yapay veri üretimi yoluna gidilmiştir.

Üçüncü adım sistem ihtiyaçlarının belirlenmesidir. Burada sistemin çalıştırılmasında kullanılacak yöntemler, araçlar belirlenecektir. Tezdeki uygulamada verinin analizi ve anlamlı çıktılar elde edebilmek için karar mekanizmasında rol alacak makine öğrenmesi modellerine, uygulamanın web ara yüzünün yapılması için ihtiyaç duyulan uygulama ve kodlama dillerine karar verilerek sistem ihtiyaçları belirlenmiştir.

Bir sonraki adım sistemin tasarımıdır. Uygulamanın ara yüzü, başvuru formu, veri girişleri, verilerin kaydedilmesi ve onların analizinin nasıl gerçekleştirileceği tasarlanmıştır.

Beşinci adım yazılımın geliştirilmesidir. Yazılım geliştirme sürecinde Python programlama diliyle makine öğrenmesi ve web ara yüzünün kodları yazılmıştır. Tüm kodlar yazılarak sistem çalışabilir hale getirildikten sonra altıncı adım olan test safhasına geçilmiştir. Karşılaşılan problemler giderilerek gerekli dersler alınmıştır.

Sistemin gerçekleştirilmesi ve değerlendirilmesi safhası sistemin kullanıcıya sunulması ve değerlendirme sürecine geçilmesidir. Bu aşamada da web tabanlı uygulamada sisteme bilgi girişleri yapılarak değerlendirmeler yapılmış, uygulamanın devamlılığı takip edilmiştir. Tez kapsamında geliştirilen uygulamada SGYD adımları takip edilmiş ve gerekli olduğunda önceki adımlara dönülerek döngü devam ettirilmiştir.

3.3. KISITLAR

Yapay Zeka Tabanlı Personel Seçim Sistemi Uygulaması bir bilişim şirketi için yapılacak veri bilimi uzmanı temininde, ön eleme yapacak bir uygulama olarak geliştirilmiştir. Yazar tarafından oluşturulan 3000 kişilik örnek veri setinde uygulanmıştır. Uygulamanın üzerinde çalışacağı aday havuzunun sayısında herhangi bir kısıt bulunmamaktadır. Ancak yapay zeka sisteminin eğitiminde kullanılan aday havuzunda kategorik olarak elendi, geçti değerlendirmeleri bulunması gerekmektedir.

Uygulama halihazırdaki makine öğrenmesi modelleriyle sadece veri bilimi uzmanı kadrosuna yapılacak başvurular için çalışmaktadır. Farklı kadrolarda veya kurum içerisinde bir göreve personel seçimi maksatlı kullanılmak istendiğinde uygun değerlendirme verisi temin edilmeli ve ilgili veri seti üzerinde makine öğrenmesi gerçekleştirilmelidir.

Sayısal olarak değerlendirilmiş bir veri seti bulunmadığı için uygulama kategorik tabanlı olarak eleme tahminlerinde bulunmaktadır. Sayısal puanlama etiket değerleri bulunan bir veri seti kullanılması durumunda regresyon modelleri çalıştırılarak sayısal tahminlerde bulunulabilir.

Yapay Zeka Tabanlı Personel Seçim Sistemi Uygulaması insan kaynağı temininde yapay zeka kullanımının başvuru formu tabanlı nitelik değerlendirmesi yöntemini kullanmaktadır. Uygulamanın başvuru ekranı herkese açık olarak işletilmekte ancak yapay zeka tahminlerinin gerçekleştirildiği yönetici girişine parolayla girilebilmektedir.

DÖRDÜNCÜ BÖLÜM

UYGULAMA

4.1. VERİ SETİNİN DEĞERLENDİRİLMESİ

Uygulamanın ismi Yapay Zeka Tabanlı Personel Seçim Sistemidir. Uygulamanın mantığı bir iş yerinde geçmişte alan uzmanı kişiler tarafından gerçekleştirilmiş aday seçimlerinin ya da büyük bir başvuru havuzunda yeteri sayıdaki veri için yapılmış değerlendirmelerin makine öğrenmesi modellerine sokularak incelenmemiş başvurular için anlamlı sonuçlar çıkarılmasıdır.

Makine öğrenmesi modellerini çalıştırabilmek maksadıyla uygulamada kullanılacak veri setinin oluşturulma süreci üçüncü bölümde açıklanmıştı. Anlamlı bir veri seti oluşturulduktan sonra veri seti için çıktıları oluşturacak uzman karar vericiye ihtiyaç duyulmaktaydı. Bu çalışmada elimizdeki veri için karar verici yazardır. Ancak normalde bu kişi uzman personel ya da iş alımını gerçekleştirecek kişi olacaktır. İş yerlerinde insan kaynağı teminini gerçekleştirecek kişiler kararlarını aday havuzunu inceleyerek iş yeri politikası ve kadronun gerekliliklerine uygun olarak gerçekleştirmektedir.

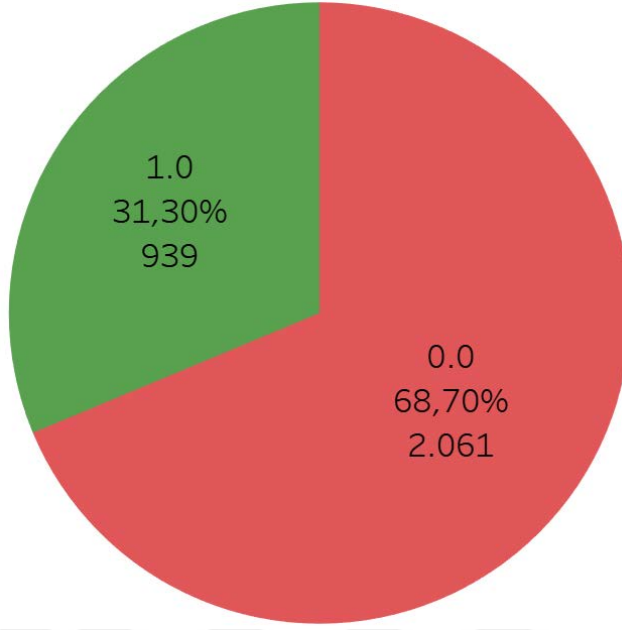
Tezde üzerinde çalışılan örnek işe alım süreci bir bilişim şirketinin veri bilimi uzmanı kadrosu için insan kaynağı temini olarak belirlenmiştir. Veri bilimi uzmanı sektörde aynı zamanda “Data Scientist” olarak da isimlendirilmektedir. 3000 kişiden oluşan örnek başvuru havuzunda adaylar yazar tarafından oluşturulan Tablo 2’deki şartlara uygun olarak değerlendirilmiştir. Genel olarak Tablo 2’deki şartları sağlayan kişilere mülakata çağırılacak şekilde geçti (1) etiketi verilmiş, şartları sağlamayan kişilere ise elendi (0) etiketi verilmiştir. Yapılan değerlendirme sonrasında veri setinde oluşan dağılım Şekil 28’de görüldüğü gibidir.

Tablo 2: Kadro Nitelikleri

	Veri Bilimi Uzmanının Sahip Olması Gereken Nitelikler	İstisnalar
1.	Yabancı Dil Bilgisi Seviye Tespit Sınavı (YDS) sınavında İngilizce'den en az 70 notunun alınmış olması gerekmektedir.	• Yoktur.
2.	Python veya R programlama dillerinden en az bir tanesinin bilinmesi gerekmektedir.	• Şirket tarafından verilen yapay zeka kursunu almış olanlar hariçtir.
3.	Adli sicil kaydı bulunmaması gerekmektedir.	• Referansı olan veya doktora yapmış olan adaylar ön görüşmeye çağırılarak durumları ayrıca değerlendirilecektir.
4.	Yapay zeka bilgisi minimum 2 seviyesinde olması gerekmektedir.	• Şirketin yapay zeka kursunu almış olan ve maaş beklentisi 4000 TL'den az olanlar, • Bilişim teknolojileri alanında alan tecrübesi 4 yıldan fazla olanlar, • Yüksek lisans veya doktora eğitimini tamamlamış olanlar hariçtir.
5.	Maaş beklentisi 6500 TL altında olmalıdır.	• Maaş beklentisi 6500-7000 TL arasında olanlar için; yüksek lisans veya doktora eğitimini tamamlayan ve son çalıştığı yerdeki ücreti 6000 TL ve altında olanlar hariçtir.
6.	Son çalışılan işteki ücret 8000 TL altında olmalıdır.	• Yoktur.
7.	Aday vardiyalı çalışabilecek olmalıdır.	• Maaş beklentisi 4000 TL ve altı olanlar hariçtir.
8.	Seyahat engeli bulunmamalıdır.	• Alan tecrübesi 3 yıl ve üzerinde olan ve vardiyalı çalışabilenler hariçtir.
9.	Ön lisans mezunlarının yapay zeka bilgi seviyesi 3 ve üstünde olmalıdır.	• Yoktur.
10.	Yaş 35 ve altı olmalıdır.	• Doktora veya yüksek lisans eğitimini tamamlamış olanlar hariçtir.

Kaynak: Yazar tarafından derlenmiştir.

Şekil 28: Karar Oranları



Kaynak: Yazar tarafından derlenmiştir.

Tablo 2’deki şartlar her kadro ve iş kolu için değişebilmektedir. Bu şartların ne olduğunun makine öğrenmesi modeli için fazla önemi bulunmamaktadır. Makine öğrenmesi şartlar üzerinden değil alınan kararlar üzerinden anlamlı bilgiler edinmektedir. Örneğin; insan kaynakları uzmanı şartların yazılı olmadığı şekilde kararlar da alabilirdi. Bu durumda sistem, uzmanın verdiği kararları takip eden çıktılar verirdi.

4.2. VERİNİN MAKİNE ÖĞRENMESİ MODELİNE HAZIRLANMASI

4.2.1. Modele Girmeyecek Sütunların Düşürülmesi

Yapay Zeka Tabanlı Personel Seçim Sistemi Uygulamasında bazı bilgiler özellikle makine öğrenmesi modelinde eğitilmeyecektir. Bunlar veri setindeki cinsiyet ve medeni durum sütunlarıdır. Buradaki amaç insanlar tarafından oluşturulabilen ön yargı veya pozitif/negatif ayrımların önüne geçerek en doğru adaylara ulaşmaktır. Bunun haricinde isim, soy isim, ikamet ili, ikamet ilçesi, askerlik durumu ve doğum tarihi (yaş sütunu modele girmektedir) sütunları da karar için anlamlı bilgi içermediğinden makine öğrenmesi modeline girmeyecektir.

4.2.2. Kategorik Verilerin Sayısallaştırılması

Elimizdeki verilerin makine öğrenmesi modeli tarafından tanınabilmesi ve üzerinde matematiksel işlemlerin gerçekleştirilebilmesi için sayısal olmayan kategorik veri sütunlarının sayısallaştırılması gerekmektedir.

Bunun için Python'un Pandas kütüphanesinde bulunan "get_dummies" metodu kullanılmaktadır. Bu yöntem sayesinde kategorik veri içeren bilinen diller, mezuniyet dalı gibi bilgiler ilgili verinin bulunduğu satırın karşılığı 1, bulunmadığı satırın karşılığı 0 olacak şekilde ayarlanmaktadır (McKinney ve diğerleri, 2020: 984-985). Bu konuya dair yabancı dil bilgisi içeren kategorik sütunun sayısala dönüştürülmesi Şekil 29'da gösterilmiştir. Tüm düzenlemeleri yaptıktan sonra oluşan ve makine öğrenmesi modeline girecek olan veri setinin örnek bir bölümü Şekil 30'da görüldüğü gibidir.

Şekil 29: Kategorikten Sayısala Dönüşüm

```
veri[["yabancidilbilgisi"]].head()
```

	yabancidilbilgisi
0	rusça
1	bilmiyor
2	ingilizce
3	fransızca
4	ingilizce

```
veri["yabancidilbilgisi"].str.get_dummies(sep="_").add_prefix("dil_").head()
```

	dil_almanca	dil_arapça	dil_bilmiyor	dil_fransızca	dil_ingilizce	dil_rusça
0	0	0	0	0	0	1
1	0	0	1	0	0	0
2	0	0	0	0	1	0
3	0	0	0	1	0	0
4	0	0	0	0	1	0

Kaynak: Yazar tarafından derlenmiştir.

Şekil 30: Modele Hazır Veri Seti

	0	1	2	3	4	5
yas	22.0	31.0	30.0	33.0	27.0	28.0
alantecrübesi	1.0	6.0	5.0	7.0	3.0	4.0
maaşbeklentisi	5000.0	4000.0	5000.0	4000.0	4000.0	5000.0
ensonücret	4000.0	4000.0	4000.0	5000.0	5000.0	5000.0
yzbilgi	1.0	0.0	0.0	0.0	0.0	0.0
karar	0.0	0.0	0.0	0.0	0.0	0.0
egitim_doktora	0.0	0.0	0.0	0.0	0.0	0.0
egitim_lisans	0.0	1.0	1.0	1.0	1.0	0.0
egitim_yükseklisans	0.0	0.0	0.0	0.0	0.0	0.0
egitim_önlisans	1.0	0.0	0.0	0.0	0.0	1.0
lisansbölümü_Bilgisayar	1.0	0.0	0.0	0.0	0.0	1.0
lisansbölümü_Bilgisayar Mühendisliği	0.0	0.0	1.0	0.0	1.0	0.0
lisansbölümü_Endüstri Mühendisliği	0.0	0.0	0.0	0.0	0.0	0.0
lisansbölümü_Matematik	0.0	0.0	0.0	0.0	0.0	0.0

Kaynak: Yazar tarafından derlenmiştir.

4.3. MAKİNE ÖĞRENMESİNİN ÇALIŞTIRILMASI

Tezde kullanılan makine öğrenmesi modellerinin çalıştırılabilmesi ve değerlendirmeler için öncelikle Şekil 31'deki kütüphanelerin yüklenmesi gerekmiştir. Bu tezde makine öğrenmesi için kullanılan temel kütüphane Scikit-learn'dür. Scikit-learn orta düzey denetimli ve denetimsiz öğrenme problemleri için makine öğrenmesi algoritmaları sağlayan kullanımı kolay bir Python kütüphanesidir (Pedregosa ve diğerleri, 2011: 2826).

Kütüphane yüklemeleri sonrasında veri seti makine öğrenmesi için veri sütunları ve karar sütunu olmak üzere ikiye ayrılmıştır. Veri sütunları modelin karar sütunundaki çıktıları anlamlandırmak için kullanacağı verilerdir.

Şekil 31: Kütüphaneler

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from xgboost import XGBClassifier
from yellowbrick.classifier import ConfusionMatrix,
ClassificationReport
import pickle
```

Kaynak: Yazar tarafından derlenmiştir.

Denetimli makine öğrenmesi modellerinde mantık belirli oranlarda ikiye ayrılan etiketli verinin bir kısmının modelin eğitimi için kullanılması ve sonrasında eğitilmiş olan model tarafından daha önce görmediği veriler üzerinde tahmin gerçekleştirilmesidir. Tahmin sonrasında test verisindeki gerçek etiketler ile modelin tahminleri karşılaştırılarak modelin başarı derecesine ulaşılmaktadır. Veri setini bölme kodu Şekil 32’de görüldüğü gibidir.

Şekil 32: Verinin Bölünmesi

```
X_egitim, X_test, y_egitim, y_test = train_test_split(X, y, test_size=0.20, stratify=y, random_state=42)
```

Kaynak: Yazar tarafından derlenmiştir.

4.3.1. Lojistik Regresyon Modeli

Elimizdeki veri setinde veriler farklı ölçeklerde bulunmaktadır. Örneğin ücretler binli değerler ile gösterilirken yaş ve alan tecrübesi gibi haneler tekli hanededirler. Bu nedenle lojistik regresyon modeli çalıştırılmadan önce verinin Scikit-learn kütüphanesinden indirilen StandartScaler() komutuyla standardize edilmesi analizi iyileştirmektedir (Buitnick ve diğerleri, 2013: 6-7).

Öncelikle ölçeklendirme yapılmadan Şekil 33’de gösterildiği şekilde Scikit-learn kütüphanesinin LogistikRegression() modeli boş olarak kurulmuştur. Sonrasında

“fit” işlemiyle model eğitim verisiyle eğitilmiştir (Scikit-learn, 2020b). Son olarak tahmin edilen çıktılarla gerçek çıktılar Tablo 3’te gösterilen hata matrisi ve sınıflandırma raporu yardımıyla değerlendirilmiştir.

Şekil 33: Lojistik Regresyon Modeli

```
log_model = LogisticRegression()  
log_model.fit(X_egitim, y_egitim)  
y_tahmin = log_model.predict(X_test)
```

Kaynak: Yazar tarafından derlenmiştir.

Tablo 3: Lojistik Regresyon Modeli Değerlendirme Sonuçları

		TAHMİN DEĞERİ	
		0	1
GERÇEK DEĞER	0	364	48
	1	55	133
Doğruluk		Hassasiyet	Geri Çağırma
		F-1 Değeri	
0.83		0.73	0.71
			0.72

Kaynak: Yazar tarafından derlenmiştir.

Sonrasında standardizenin etkisini görebilmek için veri Şekil 34’teki gibi standardize edilmiştir. Sonuç olarak Tablo 4’te görüleceği üzere yaklaşık 20 tane daha doğru tahmin yapılmıştır. Bu nedenle uygulamada Lojistik Regresyon Modeli için öncelikle veri standardize edilecektir.

Şekil 34: Standardize Lojistik Regresyon Modeli

```
olcek = StandardScaler()  
X_egitim = olcek.fit_transform(X_egitim)  
X_test = olcek.transform(X_test)  
scaleloj = LogisticRegression()  
scaleloj.fit(X_egitim, y_egitim)  
y_tahmin = scaleloj.predict(X_test)  
confusion_matrix(y_test, y_tahmin)
```

Kaynak: Yazar tarafından derlenmiştir.

Tablo 4: Standardize Model Değerlendirme Sonuçları

		TAHMİN DEĞERİ		
		0	1	
GERÇEK DEĞER	0	375	37	
	1	45	143	
Doğruluk		Hassasiyet	Geri Çağırma	F-1 Değeri
0.86		0.79	0.76	0.78

Kaynak: Yazar tarafından derlenmiştir.

4.3.2. KEYK Modeli

KEYK modelinde de verilerin eğitim öncesinde standardize işlemine tabi tutulması sonuçları daha başarılı hale getirmektedir. Bu nedenle eğitim ve test verisi standardize edilmiştir.

Şekil 35'te gösterildiği gibi Scikit-learn kütüphanesinin KNeighborsClassifier() modeli komşu eleman sayısı tercih edilerek boş olarak kurulmuştur. Sonrasında “.fit” işlemiyle model eğitim verisiyle eğitilmiştir (Scikit-learn, 2020c). Son olarak tahmin edilen çıktılarla gerçek çıktılar Tablo 5'te gösterilen hata matrisi ve sınıflandırma raporu yardımıyla değerlendirilmiştir.

Şekil 35: KEYK Modeli

```
olcek = StandardScaler()
X_egitim = olcek.fit_transform(X_egitim)
X_test = olcek.transform(X_test)
keyk_model = KNeighborsClassifier(n_neighbors = 11)
keyk_model.fit(X_egitim, y_egitim)
y_tahmin = keyk_model.predict(X_test)
```

Kaynak: Yazar tarafından derlenmiştir.

Tablo 5: KEYK Modeli Değerlendirme Sonuçları

		TAHMİN DEĞERİ	
		0	1
GERÇEK DEĞER	0	333	79
	1	28	160
Doğruluk		Hassasiyet	Geri Çağırma
		F-1 Değeri	
0.82		0.67	0.85
			0.75

Kaynak: Yazar tarafından derlenmiştir.

4.3.3. DVM Modeli

DVM modelinde eğitim ve test verisi öncelikle standardize edilmiştir. Şekil 36’da gösterildiği gibi Scikit-learn kütüphanesinin SVC() modeli kurulmuştur. Sonrasında “.fit” işlemiyle model eğitim verisiyle eğitilmiştir (Scikit-learn, 2020d). Tablo 6’da gösterilen hata matrisi ve sınıflandırma raporu incelendiğinde DVM’nin lojistik regresyon ve KEYK modellerine göre daha başarılı olduğu görülmektedir.

Şekil 36: DVM Modeli

```

olcek = StandardScaler()
X_egitim = olcek.fit_transform(X_egitim)
X_test = sc.transform(X_test)
svm_model = SVC()
svm_model.fit(X_egitim, y_egitim)
y_tahmin = svm_model.predict(X_test)

```

Kaynak: Yazar tarafından derlenmiştir.

Tablo 6: DVM Modeli Değerlendirme Sonuçları

		TAHMİN DEĞERİ	
		0	1
GERÇEK DEĞER	0	379	33
	1	37	151
Doğruluk		Hassasiyet	Geri Çağırma
		F-1 Değeri	
0.88		0.82	0.80
			0.81

Kaynak: Yazar tarafından derlenmiştir.

4.3.4. Karar Ağaçları Modeli

Karar ağaçları modelinin çalışma yapısı gereği verilerin ölçeklendirilmesine ihtiyaç duyulmamaktadır. Şekil 37’de gösterildiği gibi Sklearn kütüphanesinin DecisionTreeClassifier() modeli boş olarak kurulmuştur. Sonrasında “.fit” işlemiyle model eğitim verisiyle eğitilmiştir (Scikit-learn, 2020e). Son olarak tahmin edilen çıktılarla gerçek çıktılar Tablo 7’deki hata matrisi ve sınıflandırma raporu yardımıyla değerlendirilmiştir. Açıkça görüleceği üzere karar ağaçları modeli önceki modellere oranla yüksek bir başarı getirmiştir.

Şekil 37: Karar Ağaçları Modeli

```
agac_model = DecisionTreeClassifier()  
agac_model.fit(X_egitim, y_egitim)  
y_tahmin = agac_model.predict(X_test)
```

Kaynak: Yazar tarafından derlenmiştir.

Tablo 7: Karar Ağaçları Modeli Değerlendirme Sonuçları

		TAHMİN DEĞERİ		
		0	1	
GERÇEK DEĞER	0	401	11	
	1	5	183	
Doğruluk		Hassasiyet	Geri Çağırma	F-1 Değeri
0.97		0.94	0.97	0.96

Kaynak: Yazar tarafından derlenmiştir.

4.3.5. Rastgele Orman Modeli

Rastgele Orman Modeli için Şekil 38’de gösterildiği gibi Sklearn kütüphanesinin RandomForestClassifier() modeli içerisindeki parametreler belirlenerek boş olarak kurulmuştur. Sonrasında “.fit” işlemiyle model eğitim verisiyle eğitilmiştir (Scikit-learn, 2020f). Tablo 8’deki hata matrisi ve sınıflandırma raporu incelendiğinde Karar Ağaçları modeline yakın sonuçlar elde edildiği görülmektedir.

Şekil 38: Rastgele Orman Modeli

```
orman_model = RandomForestClassifier()  
orman_model.fit(X_egitim, y_egitim)  
y_tahmin = orman_model.predict(X_test)
```

Kaynak: Yazar tarafından derlenmiştir.

Tablo 8: Rastgele Orman Modeli Değerlendirme Sonuçları

		TAHMİN DEĞERİ		
		0	1	
GERÇEK DEĞER	0	409	3	
	1	11	177	
Doğruluk		Hassasiyet	Geri Çağırma	F-1 Değeri
0.98		0.98	0.94	0.96

Kaynak: Yazar tarafından derlenmiştir.

4.3.6. XGBoost Modeli

XGBoost modeli için Şekil 39’da gösterildiği üzere xgboost kütüphanesinin XGBClassifier() modeli boş olarak kurulmuştur. Sonrasında “.fit” işlemiyle model eğitim verisiyle eğitilmiştir (XGboost, 2020). Tablo 8’deki hata matrisi ve sınıflandırma raporu incelendiğinde 600 adet test verisinde yalnızca bir tane hata yapıldığı görülmektedir.

Şekil 39: XGBoost Modeli

```
xgb_model = XGBClassifier()  
xgb_model.fit(X_egitim, y_egitim)  
y_tahmin = xgb_model.predict(X_test)
```

Kaynak: Yazar tarafından derlenmiştir.

Tablo 9: XGBoost Modeli Değerlendirme Sonuçları

		TAHMİN DEĞERİ		
		0	1	
GERÇEK DEĞER	0	411	1	
	1	0	188	
Doğruluk		Hassasiyet	Geri Çağırma	F-1 Değeri
1		0.99	1	1

Kaynak: Yazar tarafından derlenmiştir.

4.3.7. Modellerin Değerlendirilmesi

Makine öğrenmesi gerçekleştirilerek elde edilen modeller ve Tablo 10'daki değerlendirme sonuçları incelendiğinde en başarılı modelin XGBoost modeli olduğu görülmektedir. Model bilindiği üzere artırım topluluk öğrenim metodunu kullanmaktadır. Rastgele orman modeli de torbalama topluluk öğrenim metodunu kullanmaktadır. Ancak karar ağaçları ve rastgele orman modellerinin her ikisi de benzer çok yüksek başarı oranlarına sahip olduğundan net bir değerlendirme yapma imkanı bulunmamaktadır.

Tablo 10: Model Değerlendirmeleri

	Hassasiyet	Geri Çağırma	F-1 Değeri
Lojistik Regresyon	0.79	0.76	0.78
K En Yakın Komşular	0.67	0.85	0.75
DVM	0.82	0.80	0.81
Karar Ağaçları	0.94	0.97	0.96
Rastgele Orman	0.98	0.94	0.96
XGBoost	0.99	1	1

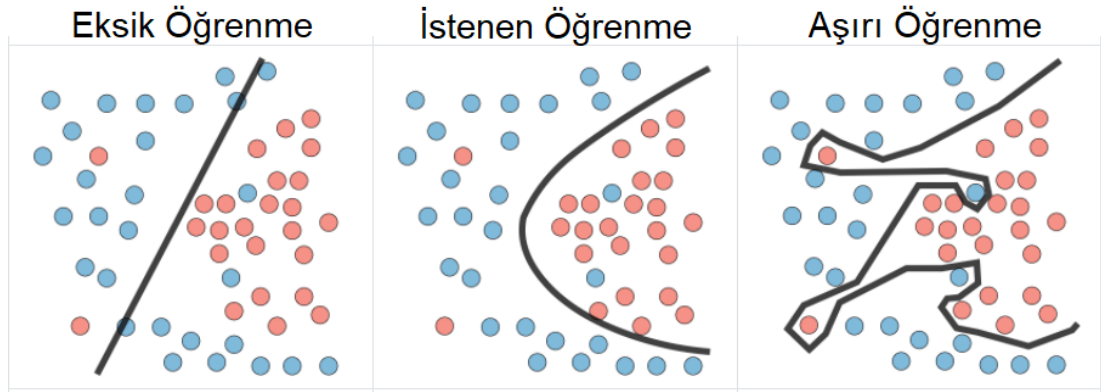
Kaynak: Yazar tarafından derlenmiştir.

XGBoost modelinin yaptığı tahminlerde sadece I. Tip Hata kapsamına giren normalde elenmesi gerekirken çağırılacak olarak değerlendirilmiş bir aday bulunmaktadır. Personel Seçim Sistemi Uygulamasında elenmesi gereken bir kişinin

yanlışlıkla mülakata çağırılması, mülakata gelmesi gereken bir kişinin elenmesine tercih edilecek bir hatadır.

Tablo 10'daki yüksek başarı seviyeleri memnuniyet verici gözükse de model değerlendirmelerinde dikkat edilmesi gereken bir husus aşırı öğrenme gerçekleşme ihtimalidir. Aşırı öğrenme modelin elindeki veri seti üzerinden ezber mantığıyla öğrenme gerçekleştirmesi ve bunun sonucunda görmediği tarzda bir girdide yanlış sonuçlar elde etmesidir. Şekil 40'ta görüleceği üzere aşırı öğrenme genel öğrenme eğrisinin dışına çıkarak ezberci bir patern çizmektedir. İstenen öğrenme şekli ise genel bir öğrenme paterni çizilmesidir (Amidi, 2020; Kocakafa, 2020).

Şekil 40: Öğrenim Türleri



Kaynak: Amidi, 2020.

Unutulmaması gereken diğer bir husus ise kullanılan verinin yapay olmasıdır. O nedenle Tablo 10'daki yüksek değerlendirme puanları tek başına, modellerin mükemmel öğrenme gerçekleştirdiğini ve sistemin halihazırdaki haliyle gerçek hayatta kullanılabileceğini göstermemektedir. Nitekim gerçek hayata ait verilerde bu kadar yüksek oranlarda başarı elde edilmesi beklenen bir durum değildir. Yazar tarafından uygulama üzerinden yapılan yeni başvuru denemelerinde özellikle XGboost modelinin çoğu tahmininde başarılı olduğu ancak nadir de olsa bazı noktalarda istenen çıktılara ulaşamadığı gözlemlenmiştir.

Sonuç olarak bu çalışmada elimizdeki veri üzerinde karar ağaçları tabanlı modellerin yüksek başarı oranı verdiği söylenebilir. Modeller ile yeni başvurular üzerinde doğru tahminler yapılabildiği görülmekte ve bu da uygulamanın insan

kaynağı temin faaliyetlerinde kullanımı konusunda heyecan uyandırmaktadır. Ancak kullanılan veri yapay olduğundan daha ileri değerlendirmeler için uygulamanın gerçek hayata ait işe alım verileri üzerinde çalıştırılması ve yapılacak değerlendirmelere göre geliştirilmesi hedeflenmektedir.

4.4. UYGULAMANIN GELİŞTİRİLMESİ

Yapay zeka değerlendirmeleri yapıldıktan sonra sıra uygulamanın geliştirilmesine gelmiştir. Uygulamada istenen şirket için web tabanlı olarak yapılacak başvuruların, aday başvurusunu tamamladığı anda kabul veya ret olarak insan kaynağı temini uzmanının önüne düşmesidir.

Bunu gerçekleştirebilmek için açık kaynak bir Python kütüphanesi olan Streamlit ile web uygulaması geliştirilmesine karar verilmiştir. Streamlit özellikle makine öğrenmesi ve veri bilimi uygulama geliştiricileri tarafından kullanılan bir kütüphanedir (Streamlit, 2020).

Streamlit kodları Python Programlama diliyle .py uzantılı olarak Pycharm kod geliştirme uygulamasıyla yazılmıştır (Jetbrains, 2020). Uygulamamızda sol taraftaki seçim menüsü üzerinden seçilen Başvuru Girişi, Yönetici Girişi ve İletişim olmak üzere 3 menü bulunmaktadır.

4.4.1. Başvuru Girişi

Şirkete dışarıdan başvuruda bulunacak adayların erişim yeri başvuru girişi bölümü olacaktır. Uygulamanın ana program fonksiyonu tanımlandıktan sonra eğer yapılarıyla sekme ayrıntılarına girilmeye başlanmıştır.

Başvuru girişinde İsim, Soy İsim, Doğum Tarihi, İkamet İli, İkamet İlçesi bölümlerine ait kodlamalar Şekil 41’de, kodlama sonucunda oluşturulan uygulama ekranı ise Şekil 42’de gösterilmiştir.

Şekil 41: Başvuru Sekmesi Kodları

```
def anaprogram():  
    secim = st.sidebar.selectbox("MENÜ", ("Başvuru Girişi", "Yönetici Girişi",  
    "İletişim"), key="a")  
  
    if secim == "Başvuru Girişi":  
        st.title("PERSONEL BAŞVURU SİSTEMİ")  
  
        cift_engelleyici = 0  
        st.subheader("Başvuru Girişi")  
        left, right = st.beta_columns(2)  
        with left:  
            isim = st.text_input("İsim")  
        with right:  
            soyisim = st.text_input("Soyisim")  
  
        dogumtarihi = st.date_input("Doğum Tarihiniz", min_value= datetime.date(1980,  
            1, 1), max_value=datetime.date(2000, 1, 1))  
  
        def yas_hesapla(dogumtarihi):  
            bugün = datetime.date.today()  
            return bugün.year - dogumtarihi.year - ((bugün.month, bugün.day) <  
                (dogumtarihi.month, dogumtarihi.day))  
        yas = yas_hesapla(dogumtarihi)  
  
        ikametili= st.text_input("İkamet İli")  
        ikametilcesi= st.text_input("İkamet İlçesi")
```

Kaynak: Yazar tarafından derlenmiştir.

Şekil 42: Uygulama Ekranı 1

PERSONEL SEÇİM SİSTEMİ

Başvuru Girişi

İsim	Soyisim
Doğum Tarihiniz	
2020/12/16	
İkamet İli	
İkamet İlçesi	

Kaynak: Yazar tarafından derlenmiştir.

Sonrasında cinsiyet, medeni durum ve askerlik durumu girdileri gelmektedir. Bu bilgiler işaretleme kutusu olacak şekilde ayarlanmıştır. Başvuranın hata sonucu iki kutuyu da işaretlemesine engel olmak için bir kontrol kodu yazılarak uyarı verilmesi sağlanmıştır.

Şekil 43: Cinsiyet, Medeni Durum ve Askerlik Durumu Kodları

```
a, b, c = st.beta_columns(3)
with a:
    st.write("Cinsiyet")
with b:
    erkek = st.checkbox('Erkek')
with c:
    kız = st.checkbox('Kız')
if erkek+kız>1:
    st.warning("Lütfen Tek Kutucuğu Seçiniz")
    cift_engelleyici = 1

a, b, c = st.beta_columns(3)
with a:
    st.write("Medenidurum")
with b:
    medenidurum_evli = st.checkbox('Evli')
with c:
    medenidurum_bekar = st.checkbox('Bekar')
if medenidurum_evli+medenidurum_bekar>1:
    st.warning("Lütfen Tek Kutucuğu Seçiniz")
    cift_engelleyici = 1

a, b, c = st.beta_columns(3)
with a:
    st.write("Askerlik Durumu")
with b:
    askerlikdurumu_tamam = st.checkbox('Muaf/Tamamlandı')
with c:
    askerlikdurumu_tamamlanmadı = st.checkbox('Tamamlanmadı')
if askerlikdurumu_tamam+askerlikdurumu_tamamlanmadı>1:
    st.warning("Lütfen Tek Kutucuğu Seçiniz")
    cift_engelleyici = 1
```

Kaynak: Yazar tarafından derlenmiştir.

Şekil 44: Uygulama Ekranı 2

Cinsiyet	☐ Erkek	☐ Kız
Medenidurum	☐ Evli	☐ Bekar
Askerlik Durumu	☐ Muaf/Tamamlandı	☐ Tamamlanmadı

Kaynak: Yazar tarafından derlenmiştir.

Eğitim durumunun belirlenmesinde işaretleme kutusu yöntemi izlenmiştir. Detaylar için ise işaretlemeye göre aşağıda açılan sekmeler düzenlenmiştir. Örneğin; yüksek lisans ve doktora bölümü detay sekmesi ancak eğitim durumunda işaretlendikleri takdirde ortaya çıkmaktadır. Eğitim durumuna bağlı olarak mezun olunan bölümler için seçeneklerden seçme veya elle girme, okul için yalnızca elle girme yöntemi izlenmiştir. Kodlamanın bir bölümü Şekil 45'te ve uygulama ekranı Şekil 46'da görüldüğü gibidir.

Şekil 45: Eğitim Durumu Kod Örneği

```
st.write("Eğitim Durumunuz")
a, b, c, d = st.beta_columns(4)
with a:
    egitim_önlisans = st.checkbox('Önlisans')
with b:
    egitim_lisans = st.checkbox('Lisans')
with c:
    egitim_yükseklisans = st.checkbox('Yükseklisans')
with d:
    egitim_doktora = st.checkbox('Doktora')
if egitim_önlisans+egitim_lisans+egitim_yükseklisans+egitim_doktora>1:
    st.warning("Lütfen Son Mezuniyetinize Göre Tek Kutucuğu Seçiniz")
    cift_engelleyici = 1

if egitim_önlisans==1 or egitim_lisans==1 or egitim_yükseklisans==1 or
egitim_doktora==1:
    lisansbölümü = st.selectbox("Lisans/Önlisans Bölümü", ("Bilgisayar Önlisans",
"Bilgisayar Mühendisliği", "Endüstri Mühendisliği", "Matematik", "Matematik
Mühendisliği", "Diğer"), key="lsbl")
    lisansokul = st.text_input('Lütfen Lisans Eğitiminizi Tamamladığınız Okulu
Yazınız:')
    if lisansbölümü == "Bilgisayar":
        lisansbölümü_Bilgisayar=1
    elif lisansbölümü == "Bilgisayar Mühendisliği":
        lisansbölümü_Bilgisayar_Mühendisliği=1
    elif lisansbölümü == "Endüstri Mühendisliği":
        lisansbölümü_Endüstri_Mühendisliği=1
    elif lisansbölümü == "Matematik":
        lisansbölümü_Matematik=1
    elif lisansbölümü == "Matematik Mühendisliği":
        lisansbölümü_Matematik_Mühendisliği=1
    elif lisansbölümü == "Diğer":
        lisans_diger = st.text_input('Lütfen Lisans Bölümünüzü Belirtiniz:')
```

Kaynak: Yazar tarafından derlenmiştir.

Şekil 46: Uygulama Ekranı 3

Eğitim Durumunuz

☐

Önlisans

☐

Lisans

☐

Yükseklisans

☒

Doktora

Lisans/Önlisans Bölümü

Diğer

Lütfen Lisans Eğitiminizi Tamamladığınız Okulu Yazınız:

Lütfen Lisans Bölümünüzü Belirtiniz:

Yüksek Lisans Bölümü

Bilgisayar Mühendisliği

Lütfen Yüksek Lisans Eğitiminizi Tamamladığınız Okulu Yazınız:

Doktora Bölümü

Bilgisayar Mühendisliği

Kaynak: Yazar tarafından derlenmiştir.

Programlama dili bilgisi için işaretleme kutusu yöntemi izlenmiştir. Seçenekler arasında yer almayan bilgiler için ise diğer seçeneğinin işaretlenmesine bağlı olarak açılan sekme düzenlenmiştir. Şirket tarafından 2018-2020 yılları arasında bir yapay zeka kursu verildiği varsayılmış ve bu durum eleme sisteminde değerlendirmeye alınmıştır. Bu nedenle burada katıldım, katılmadım olarak iki seçenek konmuştur. Bu sekmelerin kodlama örnekleri Şekil 47’de ve uygulama ekranı Şekil 48’de görüldüğü gibidir.

Şekil 47: Programlama Dilleri, Yapay Zeka Kursu Kodları

```
diger_prog_dil = 0
st.write("Bildiğiniz Programlama Dillerini İşaretleyiniz")
a, b, c, d, e = st.beta_columns(5)
with a:
    program_python = st.checkbox('Python')
with b:
    program_R = st.checkbox('R')
with c:
    program_C = st.checkbox('C')
with d:
    program_Cplusplus = st.checkbox('C++')
with e:
    program_sql = st.checkbox('SQL')
a, b, c, d, e = st.beta_columns(5)
with a:
    program_javascript = st.checkbox('Javascript')
with b:
    program_scala = st.checkbox('Scala')
with c:
    program_julia = st.checkbox('Julia')
with d:
    program_PHP = st.checkbox('PHP')
with e:
    dig = st.checkbox('Diğer', key="dig_prog")
if dig == 1:
    dig_prog_dil = st.text_input('Lütfen Bildiğiniz Diğer Dilleri Belirtiniz:')

st.write("2018-2020 Yılları Arasında Şirketimizin Düzenlediği Yapay Zeka Kursuna Katıldınız mı?")
a, b, c = st.beta_columns(3)
with a:
    q = 0
with b:
    kurs_aldim = st.checkbox('Katıldım')
with c:
    kurs_almadim = st.checkbox('Katılmadım')
```

Kaynak: Yazar tarafından derlenmiştir.

Şekil 48: Uygulama Ekranı 4

Bildiğiniz Programlama Dillerini İşaretleyiniz

☐ Python	☐ R	☐ C	☐ C++	☐ SQL
☐ Javascript	☐ Scala	☐ Julia	☐ Tableau	☒ Diğer

Lütfen Bildiğiniz Diğer Dilleri Belirtiniz:

2018-2020 Yılları Arasında Şirketimizin Düzenlediği Yapay Zeka Kursuna Katıldınız mı?

☐ Katıldım	☐ Katılmadım
-----------------------------------	-------------------------------------

Kaynak: Yazar tarafından derlenmiştir.

Yapay zeka bilgisi 0-4 arasında sürüklemeli seçenek ile, yabancı dil bilgisi işaretleme kutucukları ile, alan tecrübesi, maaş beklentisi ve en son alınan ücret kullanıcı girişine müsaade edecek şekilde ayarlanmıştır. Kodlamalar Şekil 49’da ve uygulama ekranı Şekil 50’de görüldüğü gibidir.

Şekil 49: Yapay Zeka Bilgisi, Dil, Alan Tecrübesi Kodları

```
yzbilgi = st.slider("Yapay Zeka Bilginizi 0-4 Arasında Puanlayınız", 0, 4)

diger_yabancı_dil = 0
st.write("Bildiğiniz Yabancı Dilleri İşaretleyiniz")
a, b, c, d, e = st.beta_columns(5)
with a:
    dil_ingilizce = st.checkbox('İngilizce')
with b:
    dil_almanca = st.checkbox('Almanca')
with c:
    dil_fransızca = st.checkbox('Fransızca')
with d:
    dil_arapça = st.checkbox('Arapça')
with e:
    dil_rusça = st.checkbox('Rusça')
a, b = st.beta_columns(2)
with a:
    dil_bilmiyorum = st.checkbox('Yabancı Dil Bilmiyorum')
with b:
    diger_yb_dil = st.checkbox('Diğer', key="dig_dil")
if diger_yb_dil == 1:
    diger_yabancı_dil = st.text_input('Lütfen Bildiğiniz Diğer Dilleri Belirtiniz:')

alantecrübesi= st.text_input("IT Alanındaki İş Tecrübeniz Kaç Yıldır?")
maasbeklentisi= st.text_input("Maaş Beklentiniz Aylık Kaç TL'dir?")
ensonücret= st.text_input("En Son Çalıştığınız Şirketteki Maaşınız Aylık Kaç TL'ydi?")
```

Kaynak: Yazar tarafından derlenmiştir.

Şekil 50: Uygulama Ekranı 5

The screenshot displays the application's user interface. At the top, there is a slider titled "Yapay Zeka Bilginizi 0-4 Arasında Puanlayınız" (Rate your AI knowledge on a scale of 0-4). Below this, a section titled "Bildiğiniz Yabancı Dilleri İşaretleyiniz" (Mark the foreign languages you know) contains five checkboxes for "İngilizce" (English), "Almanca" (German), "Fransızca" (French), "Arapça" (Arabic), and "Rusça" (Russian). There is also a checkbox for "Yabancı Dil Bilmiyorum" (I don't know foreign language) and a "Diğer" (Other) checkbox. Below these, there are three text input fields: "IT Alanındaki İş Tecrübeniz Kaç Yıldır?" (How many years of work experience do you have in the IT field?), "Maaş Beklentiniz Aylık Kaç TL'dir?" (What is your monthly salary expectation in TL?), and "En Son Çalıştığınız Şirketteki Maaşınız Aylık Kaç TL'ydi?" (What was your monthly salary in the last company you worked for?).

Kaynak: Yazar tarafından derlenmiştir.

Adli sicil kaydı, seyahat engel durumu, referans ve vardiyalı çalışma durumu bilgileri işaretleme kutusu olacak şekilde ayarlanmıştır. Referansım vardır seçeneği işaretlendikten sonra altta şirket içerisindeki referansın kim olduğunun yazılacağı doldurma kutusu çıkmaktadır. Kodlamalar Şekil 51’de ve uygulama ekranı Şekil 52’de görüldüğü gibidir.

Şekil 51: Adli Sicil, Seyahat, Referans, Vardiya Kodları

```
a, b, c = st.beta_columns(3)
with a:
    st.write("Adli Sicil Kaydı")
with b:
    adlisicilkaydı_yoktur = st.checkbox('Adli Sicil Kaydım Yoktur')
with c:
    adlisicilkaydı_vardır = st.checkbox('Adli Sicil Kaydım Vardır')
if adlisicilkaydı_yoktur + adlisicilkaydı_vardır > 1:
    st.warning("Lütfen Tek Kutucuğu Seçiniz")
    cift_engelleyici = 1

a, b, c = st.beta_columns(3)
with a:
    st.write("Seyahat Engel Durumu")
with b:
    seyahatengel_yoktur = st.checkbox('Seyahat Engelim Yoktur')
with c:
    seyahatengel_vardır = st.checkbox('Seyahat Engelim Vardır')
if seyahatengel_yoktur + seyahatengel_vardır > 1:
    st.warning("Lütfen Tek Kutucuğu Seçiniz")
    cift_engelleyici = 1

a, b, c = st.beta_columns(3)
with a:
    st.write("Referans")
with b:
    referans_yoktur = st.checkbox('Referansım Yoktur')
with c:
    referans_vardır = st.checkbox('Referansım Vardır')
if referans_yoktur+referans_vardır>1:
    st.warning("Lütfen Tek Kutucuğu Seçiniz")
    cift_engelleyici = 1
if referans_vardır == 1:
    referans = st.text_input('Şirketimizdeki Referansınız Kimdir?:')

a, b, c = st.beta_columns(3)
with a:
    st.write("Vardiyalı Çalışma Durumunuz?")
with b:
    vardiyaimkanı_çalışabilirim = st.checkbox('Çalışabilirim')
with c:
    vardiyaimkanı_çalışamam = st.checkbox('Çalışamam')
if vardiyaimkanı_çalışabilirim+vardiyaimkanı_çalışamam>1:
    st.warning("Lütfen Tek Kutucuğu Seçiniz")
    cift_engelleyici = 1
```

Kaynak: Yazar tarafından derlenmiştir.

Şekil 52: Uygulama Ekranı 6

Adli Sicil Kaydı	☐ Adli Sicil Kaydım Yoktur	☐ Adli Sicil Kaydım Vardır
Seyahat Engel Durumu	☐ Seyahat Engelim Yoktur	☐ Seyahat Engelim Vardır
Referans	☐ Referansım Yoktur	☐ Referansım Vardır
Vardiya Çalışma Durumunuz?	☐ Çalışabilirim	☐ Çalışamam

Tamamla

Kaynak: Yazar tarafından derlenmiştir.

Aday bilgilerini tamamlayarak başvuru tamamlama butonuna bastığında bilgilerin makine öğrenmesine girecek bölüm ve tüm bilgiler olarak iki farklı veri setine kaydedilmesi gerekmektedir. Bu nedenle Şekil 53'te bir bölümü gösterildiği gibi, bilgi girişleri değişkenlere göre ayarlanan iki adet Python sözlüğü tanımlanmıştır.

Şekil 53: Veri Seti Sözlüğünün Bir Kısım

```
if st.button("Tamamla") and cift_engelleyici!=1:
    yz_haric = {"isim":isim, "soyisim":soyisim, "dogumtarihi":dogumtarihi,
    "ikametili":ikametili, "ikametilcesi":ikametilcesi, "cinsiyet_erkek":erkek,
    "cinsiyet_kiz":kiz, "medenidurum_evli":medenidurum_evli, "medenidurum_bekar":
    medenidurum_bekar, "askerlikdurumu_tamam":askerlikdurumu_tamam,
    "askerlikdurumu_tamamlanmadı":askerlikdurumu_tamamlanmadı, "lisans_diger":lisans_diger
    , "yükseklisans_diger":yükseklisans_diger, "doktorabölümü_diger":doktorabölümü_diger,
    "diger_prog_dil":diger_prog_dil, "diger_yabancı_dil":diger_yabancı_dil,
    "lisansokul":lisansokul, "yükseklisansokul":yükseklisansokul,
    "doktoraokul":doktoraokul, "referans":referans}
    listem = {'yas': yas, 'alantecrübesi': alantecrübesi, 'maaşbeklentisi':
    maaşbeklentisi...}
```

Kaynak: Yazar tarafından derlenmiştir.

Oluşturulan sözlükler iki farklı veri setine dönüştürülerek CSV formatında bilgisayara kaydedilmiştir. Makine öğrenmesi tahminleri, yönetici girişi üzerinden, kaydedilen veri setleri modele sokularak gerçekleştirilecektir. Kodlamalar Şekil 54'te olduğu gibidir.

Şekil 54: Başvuru Tamamlama Kodları

```
yz_haric_verisi = pd.DataFrame.from_dict(yz_haric, orient="index").T
yz_verisi = pd.DataFrame.from_dict(listem, orient="index").T
tum_veriler = yz_haric_verisi.join(yz_verisi)

try:
    basvurular = pd.read_csv("basvurular_yz.csv")
    basvurular = basvurular.append(yz_verisi, ignore_index=True)
    tum_veri = pd.read_csv("basvurular_tum_veri.csv")
    tum_veri = tum_veri.append(tum_veriler, ignore_index=True)
except:
    basvurular = yz_verisi
    tum_veri = tum_veriler

basvurular.to_csv("basvurular_yz.csv", index=False)
tum_veri.to_csv("basvurular_tum_veri.csv", index=False)
st.write("BAŞVURUNUZ TAMAMLANMIŞTIR")
elif cift_engelleyici == 1:
    st.warning("LÜTFEN BİLGİLERİNİZİ KONTROL EDİNİZ")
```

Kaynak: Yazar tarafından derlenmiştir.

4.4.2. Yönetici Girişi

Yönetici girişi bölümü uygulamada makine öğrenmesi değerlendirmelerinin ve tahminlerin sergilendiği kısımdır. Yalnızca insan kaynakları uzmanları ve yöneticilerin kullanımına açıktır. Giriş belirlenen parolaya göre olmaktadır.

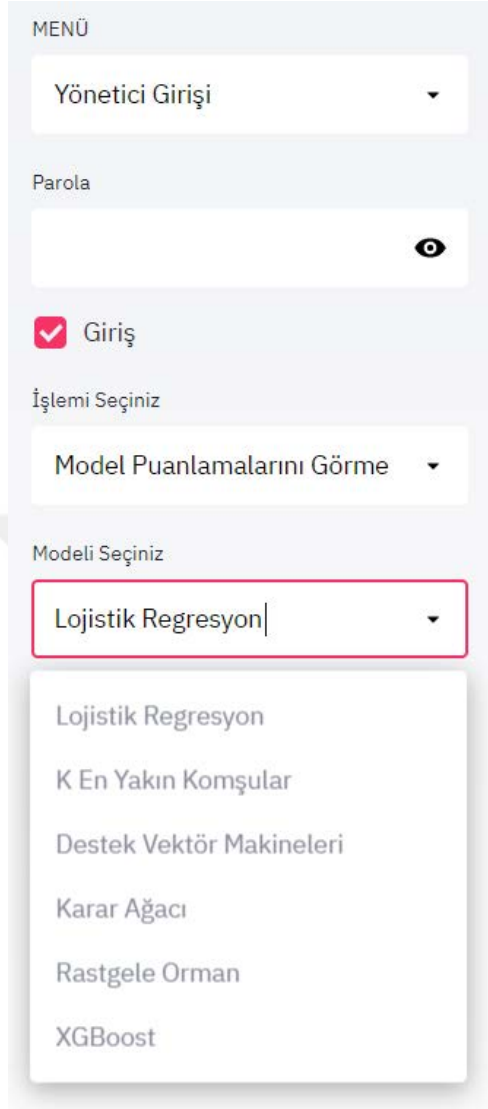
Parola girildikten sonra makine öğrenmesi modellerimizin değerlendirilebileceği ve aynı modellere göre yeni başvuru yapan adayların tahminlerinin incelenebileceği bölümler açılmaktadır. Yanlış parola girişi yapıldığı takdirde ekranda uyarı verilmektedir. Kodlamalar Şekil 55'te ve uygulama ekranı Şekil 56'da görüldüğü gibidir.

Şekil 55: Yönetici Girişi Kodları

```
elif secim == "Yönetici Girişi":
    st.title("PERSONEL SEÇİM SİSTEMİ")
    sifre = st.sidebar.text_input("Parola", type="password", key="b")
    if st.sidebar.checkbox("Giriş"):
        if sifre == "":
            islem = st.sidebar.selectbox("İşlemi Seçiniz", ("Model Puanlamalarını Görme", "Başvuru Değerlendirme"), key="c")
            st.header(islem)
            secilen_model = st.sidebar.selectbox("Modeli Seçiniz", ("Lojistik Regresyon", "K En Yakın Komşular", "Destek Vektör Makineleri", "Karar Ağacı", "Rastgele Orman", "XGBoost"), key="d")
```

Kaynak: Yazar tarafından derlenmiştir.

Şekil 56: Yönetici Yan Sekme Ekranı



The screenshot displays a web interface for an administrator. At the top, there is a 'MENÜ' (Menu) section with a dropdown menu currently set to 'Yönetici Girişi' (Admin Login). Below this is a 'Parola' (Password) field with a toggle icon. A red checkmark icon is next to the 'Giriş' (Login) button. Underneath is an 'İşlemi Seçiniz' (Select Operation) dropdown menu set to 'Model Puanlamalarını Görme' (View Model Scores). At the bottom, there is a 'Modeli Seçiniz' (Select Model) dropdown menu, which is currently open, showing a list of machine learning models: 'Lojistik Regresyon' (Logistic Regression), 'K En Yakın Komşular' (K-Nearest Neighbors), 'Destek Vektör Makineleri' (Support Vector Machines), 'Karar Ağacı' (Decision Tree), 'Rastgele Orman' (Random Forest), and 'XGBoost'. The 'Lojistik Regresyon' option is highlighted with a red border.

Kaynak: Yazar tarafından derlenmiştir.

Yönetici girişi bölümünde iki adet yan sekme açılmaktadır. Üstteki sekmeden model puanlarını görme veya başvuruları değerlendirme seçenekleri belirlenmektedir. Alttaki sekmede iki işlem için de çalışan model seçimi sekmesi bulunmaktadır.

İşlem seçimi gerçekleştirildikten sonra makine öğrenmesinin eğitilmesi maksadıyla veri seti yüklenerek nitelik verileri ve etiket olarak ikiye ayrılması gerekmektedir. Başvuru değerlendirme sekmesi için başvurular veri setinin yüklemesi yapılmıştır. Modellere göre girilebilecek parametreler için parametre_belirle, model yüklemesi için de modeli_çalıştır fonksiyonları tanımlanmıştır (Loeber, 2020). Lojistik

Regresyon, KEYK ve DVM modelleri için veriler standardize edilmiştir. Kodlamalar Şekil 57’de görüldüğü gibidir.

Şekil 57: Veri Setini Yükleme Kodları

```
df = pd.read_csv("yz.csv")
X = df.drop("karar", axis=1)
y = df["karar"]
try:
    basvuru_degerlendirme = pd.read_csv("basvurular_yz.csv")
except:
    pass
def parametre_belirle(secilen_model):
    param = dict()
    if secilen_model == "K En Yakın Komşular":
        K = st.sidebar.slider("K", 1, 15)
        param["K"] = K
    elif secilen_model == "Rastgele Orman":
        max_depth = st.sidebar.slider("max_depth", 2, 15)
        n_estimators = st.sidebar.slider("n_estimators", 1, 100)
        param["max_depth"] = max_depth
        param["n_estimators"] = n_estimators
    return param
parametreler = parametre_belirle(secilen_model)
def modeli_calistir(secilen_model, parametreler):
    if secilen_model == "Lojistik Regresyon":
        modelim = LogisticRegression()
    elif secilen_model == "K En Yakın Komşular":
        modelim = KNeighborsClassifier(n_neighbors=parametreler["K"])
    elif secilen_model == "Destek Vektör Makineleri":
        modelim = SVC()
    elif secilen_model == "Karar Ağacı":
        modelim = DecisionTreeClassifier()
    elif secilen_model == "Rastgele Orman":
        modelim = RandomForestClassifier(max_depth=parametreler["max_depth"],
                                         n_estimators=parametreler["n_estimators"], random_state=42)
    elif secilen_model == "XGBoost":
        modelim = XGBClassifier()
    return modelim
model = modeli_calistir(secilen_model, parametreler)
X_egitim, X_test, y_egitim, y_test = train_test_split(X, y, test_size=0.20,
stratify=y, random_state=42)
if secilen_model=="Lojistik Regresyon" or secilen_model=="K En Yakın Komşular" or
secilen_model=="Destek Vektör Makineleri":
    normalize = StandardScaler()
    X_egitim = normalize.fit_transform(X_egitim)
    X_test = normalize.transform(X_test)
    X = normalize.transform(X)
try:
    basvuru_degerlendirme = normalize.transform(basvuru_degerlendirme)
except:
    pass
```

Kaynak: Yazar tarafından derlenmiştir.

4.4.2.1. Model Puanlarını Görme

Model puanlarını görme sekmesinde modellerin değerlendirilmeleri için yapılacak görsel çizimler kodlanmıştır. Hassasiyet, Geri Çağırma ve F1 değerlendirmeleri sınıflandırma raporunda, tahminlerin durumu ise hata matrisinde görsel olarak gösterilmektedir.

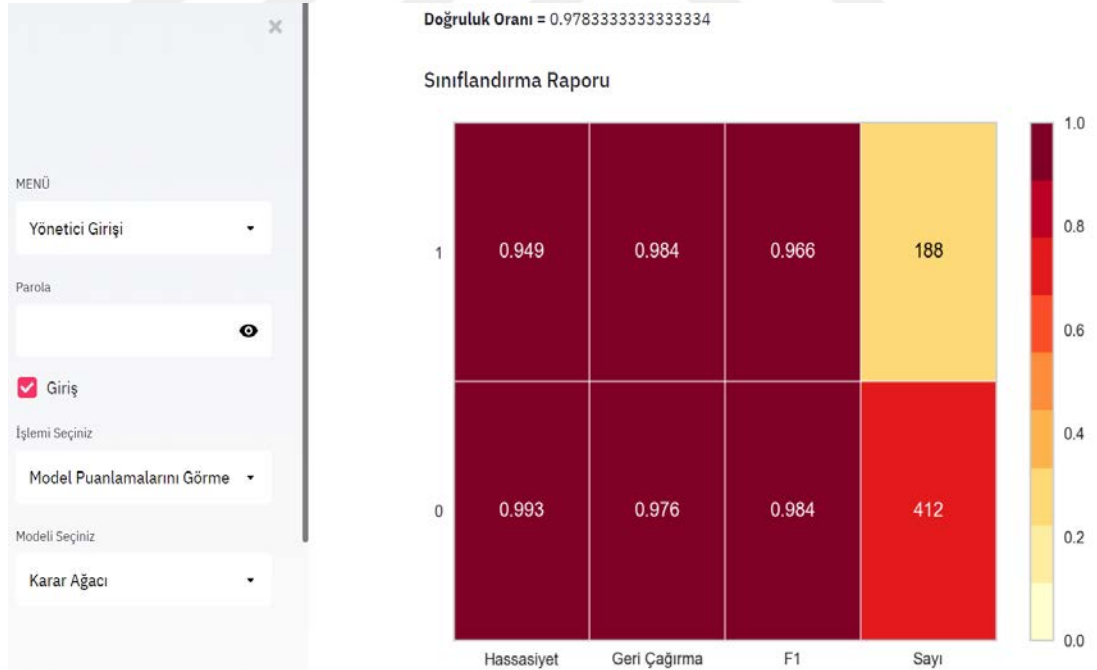
Şekil 58: Model Değerlendirme Kodları

```
if islem == "Model Puanlamalarını Görme":
    model.fit(X_egitim, y_egitim)
    y_tahmin = model.predict(X_test)
    dogruluk_oran = accuracy_score(y_test, y_tahmin)
    st.markdown(f"Model = **{secilen_model}**")
    st.write(f"Doğruluk Oranı = **{dogruluk_oran}**")
    st.subheader("Sınıflandırma Raporu")
    fig1= plt.figure()
    viz = ClassificationReport(model, support=True)
    viz.fit(X_egitim, y_egitim)
    viz.score(X_test, y_test)
    plt.xticks([0.5, 1.5, 2.5, 3.5], ['Hassasiyet', 'Geri Çağırma', 'F1', 'Sayı'])
    plt.yticks([0.5, 1.5], ['0', '1'])
    st.pyplot(fig1)
    cnf_matrix = confusion_matrix(y_test, y_tahmin)
    fig2=plt.figure()
    sns.heatmap(cnf_matrix, annot=True, cmap="YlGnBu", fmt='d')
    plt.ylabel('Asıl Etiket')
    plt.xlabel('Tahmin Edilen Etiket')
    st.pyplot(fig2)
```

Kaynak: Yazar tarafından derlenmiştir.

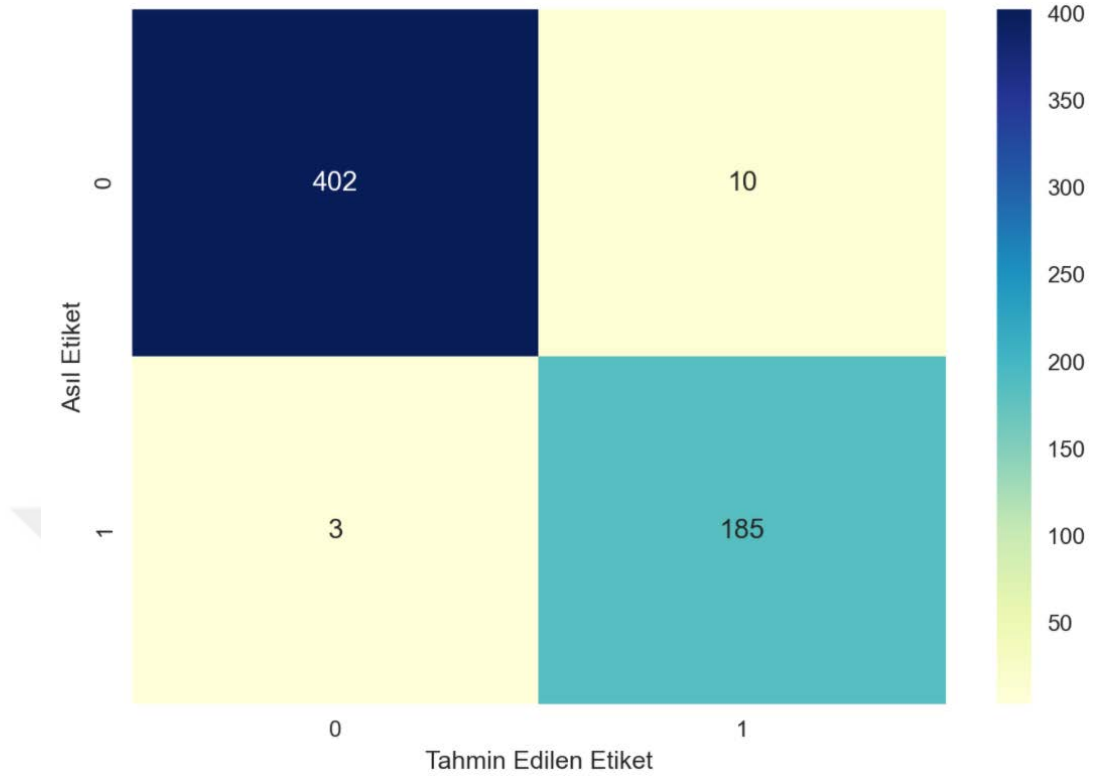
Model puanlarını görme sekmesinde ekranda gösterilen sınıflandırma raporu Şekil 59'da ve hata matrisi Şekil 60'ta görüldüğü gibidir.

Şekil 59: Uygulama Ekranı 7



Kaynak: Yazar tarafından derlenmiştir.

Şekil 60: Uygulama Ekranı 8



Kaynak: Yazar tarafından derlenmiştir.

4.4.2.2. Başvuru Değerlendirme

Başvuru değerlendirme bölümü seçildiğinde tüm veri setiyle eğitilen modeller yeni başvurular verisi üzerinde tahminlerini yapacaktır. Model ve parametre seçimlerine göre interaktif olarak tahminler de değişebilmektedir. Model değerlendirme bölümünde en başarılı model görüldükten sonra tahmin için o model seçilebilir.

Şekil 61: Model Değerlendirme Kodları

```
elif islem == "Başvuru Değerlendirme":  
    model.fit(X, y)  
    y_pred = model.predict(basvuru_degerlendirme)  
    cıktı = pd.read_csv("basvurular_tum_veri.csv")  
    for i in range(len(cıktı)):  
        sonuc = pd.DataFrame(cıktı.loc[i])  
        sonuc = sonuc.T  
        sonuc = sonuc[sonuc != False]  
        sonuc = sonuc.dropna(axis=1)  
        sonuc.insert(0, "karar", y_pred[i])  
        st.write(sonuc)
```

Kaynak: Yazar tarafından derlenmiştir.

Şekil 62: Başvuru Değerlendirme Ekranı

MENÜ

Yönetici Girişi

Parola

☒ Giriş

İşlemi Seçiniz

Başvuru Değerlendirme

Modeli Seçiniz

XGBoost

PERSONEL SEÇİM SİSTEMİ

Başvuru Değerlendirme

	karar	isim	soyisim	dogumtarihi	ikametili	ikametilcesi	cinsiyet
0	1	Mehmet		1988-05-25	İstanbul	Kadıköy	
1	1	Metin		1989-07-20	İstanbul	Kartal	
2	1	Meltem		1991-01-16	İstanbul	Beylikdüzü	
3	0	Ayşe		1995-04-26	İstanbul	Beylikdüzü	
4	0	Erdem		1981-05-14	İstanbul	Beykoz	

Kaynak: Yazar tarafından derlenmiştir.

4.4.3. İletişim Bölümü

Bir iş başvurusu uygulamasında yer alması gereken şirkete ait iletişim kurulabilecek kaynakların gösterildiği iletişim bölümü uygulamanın son sekmesine yerleştirilmiştir.

Şekil 63: İletişim Bölümü Kodları

```
elif secim == "İletişim":
    st.title("ŞİRKETİMİZE AİT İLETİŞİM BİLGİLERİ")
    st.markdown("Şirketimize ulaşmak ve ekstra bilgi edinmek için aşağıdaki "
                "iletişim yöntemleriyle müşteri temsilcimizle görüşebilirsiniz.")
    st.markdown("***Telefon Numaramız **: 02120000000")
    st.markdown("***Faks **: 02120000000")
    st.markdown("***Email **: bilişim@bilişim.com")
    st.markdown("***Çalışma Saatlerimiz** 09:00-18:00")
    st.markdown("***Adres **: Levent Mah. Deniz Cad. Bilişim Plaza No:1
                Kağıthane/İstanbul")
```

Kaynak: Yazar tarafından derlenmiştir.

Şekil 64: İletişim Uygulama Ekranı

MENÜ

İletişim

ŞİRKETİMİZE AİT İLETİŞİM BİLGİLERİ

Şirketimize ulaşmak ve ekstra bilgi edinmek için aşağıdaki iletişim yöntemleriyle müşteri temsilcimizle görüşebilirsiniz.

Telefon Numaramız : 02120000000

Faks : 02120000000

Email : bilişim@bilişim.com

Çalışma Saatlerimiz 09:00-18:00

Adres : Levent Mah. Deniz Cad. Bilişim Plaza No:1 Kağıthane/İstanbul

Kaynak: Yazar tarafından derlenmiştir.

SONUÇ

Kurumların sahip oldukları en kıymetli değer olan insan gücünü doğru şekilde belirleyebilmeleri için birçok çalışma yapılmaktadır. Özellikle son yıllarda bu alanda yapay zeka teknolojisinin kullanımı için çalışmalar hız kazanmıştır. Uluslararası boyutta büyük şirketler milyonlarca kişi arasından en doğru kişileri yapay zeka teknolojisinden faydalananarak seçmektedir. Ancak teknoloji yeni kullanılmaya başlandığından hala nihai kararlarda şirket karar vericileri tamamen kontrol sahibidir.

Yapay zeka teknolojisinin alanda kullanımının farklı versiyonları bulunmaktadır. Yapay zeka vasıtasıyla özgeçmiş formlarının değerlendirilmesi, sanal oyunlar ile değerlendirme, videolu online mülakat, sohbet robotlarıyla mülakat veya sosyal medya analizi kullanılan bazı yöntemlerdir.

Bu tez kapsamında ise yapay zeka vasıtasıyla özgeçmiş formlarının değerlendirilmesi yöntemine benzer olarak form tabanlı aday değerlendirmesi yöntemi izlenmiştir. Özgeçmiş formları değerlendirmesinde adayların veri formlarından bilgi çekerek analiz etme yöntemi izlenmektedir. Ancak özgeçmiş formları genellikle adaylar tarafından farklı formatlarda, faydalı olmayan bilgiler de ihtiva edebilecek şekilde hazırlanabilmektedir. Bundan dolayı başvuru formu tabanlı değerlendirmenin ilk değerlendirmeler için daha başarılı olacağı düşünülmektedir. Bu çalışmada ilk başta adaylardan direkt olarak hedefe yönelik bilgi talebi yapılarak sürecin hızlandırılması ve daha doğru sonuçların elde edilmesi hedeflenmiştir.

Yapay Zeka Tabanlı Personel Seçimi Sistemi Uygulaması makine öğrenmesi yöntemleri kullanılarak web ara yüzü ile gerçekleştirilmiştir. Tezin ilk aşamasında makine öğrenmesinde eğitilecek veri seti hazırlanmış ve adaylar karar verici yazar tarafından geçti/kaldı şeklinde iki farklı sonuçla etiketlenmiştir.

Etiketlenmiş veri seti ile Lojistik Regresyon, K En Yakın Komşular, Destek Vektör Makineleri, Karar Ağaçları, Rastgele Orman ve XGBoost denetimli makine öğrenmesi modelleri eğitilmiştir. Değerlendirme aşamasında veri seti için karar ağaçları tabanlı modellerin daha başarılı olduğu ve topluluk öğrenim metodunu kullanan XGBoost modelinin en iyi sonucu veren sınıflandırma algoritması olduğu tespit edilmiştir. Ancak kullanılan veri yapay olduğundan dolayı daha ileri seviye analizler için gerçek hayat verileri ile çalışma yapılması gerektiği değerlendirilmiştir.

Makine öğrenmesi modelleri geliştirildikten sonra başvuru formlarının doldurulup, verilerin kaydedileceği ve modellere göre başvuru değerlendirmelerinin yapılabilmesi için web ara yüzünde uygulama geliştirilmiştir. Uygulamanın başvuru bölümünde aday bilgilerinin girişi yapılarak tamamlama butonuna basıldığında veriler kaydedilmektedir. Sonrasında yönetici girişi bölümünden istenen makine öğrenmesi modeline göre aday için karar tahminleri izlenebilmektedir.

Bu çalışmada insan kaynağı temininde ön eleme aşamasında değerlendirmelerde bulunulmuş ancak sistem istendiği takdirde ufak düzenlemelerle sayısal çıktıların bulunduğu anlamlı bir veri setinde de çalıştırılabilecektir. Sayısal veriler kullanıldığı takdirde makine öğrenmesi regresyon modelleri kullanılacaktır.

Veri temizleme safhası hesaba katılmadığı takdirde geçmişte binlerce kişi üzerinde yapılmış değerlendirmeler, elimizdeki uygulama ile dakikalar içerisinde incelenerek, yeni başvuruların değerlendirilmesinde kullanılabilir.

Yönetim bilişim sistemleri, anlamsız veri topluluklarından anlamlı bilgiye ulaşmayı ve bu anlamlı bilgiyle karar vericileri doğru şekilde yönlendirmeyi hedefleyen bir disiplindir. Yapay Zeka Tabanlı Personel Seçim Sistemi Uygulaması da yöneticilerin kurumlarına yapacakları personel seçiminde hızlı ve doğru kararlar almalarına yardımcı olan bir yönetim bilişim sistemidir.

Bu tezdeki çalışmanın geçmiş yıllara ait personel alımı verilerini saklayan bir şirkette, gerçek veriler üzerinde denenmesi ve alınan sonuçlara göre gerekli geliştirmelerin yapılması hedeflenmektedir. İlerleyen çalışmalarda ön eleme safhası sonrasında, yapay zeka puanlandırması ile aday başarı sıralaması yapılması ve yapay sinir ağları kullanarak makine öğrenmesi modellerinin yanında derin öğrenme algoritmaları ile de tahminler yürütülmesi planlanmaktadır.

Sonuç olarak Yapay Zeka Tabanlı Personel Seçim Sistemi Uygulaması, iş alım faaliyetlerinde yapay zeka ve makine öğrenmesi tabanlı sistemlerin geliştirilmesinin maddi kaynaklar ve zaman açısından büyük tasarruflar sağlayabileceğini göstermektedir. Yalnızca insan kaynağı temin süreçlerinde değil karar verme faaliyetinin gerçekleştirildiği her alanda yapay zeka teknolojilerinden faydalanılması ve bu alanlarda daha fazla çalışma yapılması gelecek adına yapılabilecek en mantıklı yatırımlardan birisi olacaktır.

KAYNAKÇA

Amidi, A. ve Amidi, S. (2020). *Machine Learning Tips and Tricks Cheatsheet*. <https://stanford.edu/~shervine/teaching/cs-229/cheatsheet-machine-learning-tips-and-tricks#>, (28.12.2020).

Ayhan, S. ve Doğmuş, Ş. (2014) Destek Vektör Makineleriyle Sınıflandırma Problemlerinin Çözümü İçin Çekirdek Fonksiyonu Seçimi. *Eskişehir Osmangazi Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*. 9(1): 175-201.

Breiman, L. (2001) Random Forests. *Machine Learning*. 45: 5-32.

Buitnick, L., Louppe, G., Blondel, M., Pedregosa, F., Müller, A.C., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt B. ve Varoquaux G. (1 Eylül 2013). *API Design For Machine Learning Software: Experiences From The Scikit-learn Project*. <https://arxiv.org/abs/1309.0238>, (15.12.2020).

Uyargil C., Adal Z., Ataay İ.D., Acar A.C., Özçelik A.O., Sadullah Ö., Dünder G. ve Tüzüner V.L. (2008). *İnsan Kaynakları Yönetimi*. İstanbul: Beta Basım Yayın.

Chai, T. ve Draxler, R. (2014). Root Mean Square Error (RMSE) Or Mean Absolute Error (MAE)? - Arguments Against Avoiding RMSE In The Literature. *Geoscientific Model Development*. 7(3): 1247-1250.

Chen, T. ve Guestrin, G. (2016). XGBoost: A Scalable Tree Boosting System. *KDD'16: Proceedings Of The 22nd ACM SIGKDD International Conference On Knowledge Discovery and Data Mining* (ss.785-794), Düzenleyen Association For Computing Machinery. San Francisco. 13-17 Ağustos 2016.

Collins, L., Fineman, D.R. ve Tsuchida, A. (2017). People Analytics: Recalculating The Route. *Rewriting The Rules For The Digital Age 2017 Deloitte Global Human Capital Trends*. (ss.97-105). Deloitte University Press.

Cover, T.M ve Hart, P.E. (1967). Nearest Neighbor Pattern Classification. *IEEE Transactions On Information Theory*. 13(1): 21-27.

Dasgupta, A. ve Nath A. (2016). Classification of Machine Learning Algorithms. *International Journal of Innovative Research in Advanced Engineering*. 3(3): 6-11.

Dastin, J. (11 Ekim 2018). *Amazon Scraps Secret AI Recruiting Tool That Showed Bias Against Women*. <https://www.reuters.com/article/us-amazon-com-jobs-automation-insight-idUSKCN1MK08G>, (20.12.2020)

Davis, J. ve Mark G. (2006) The Relationship Between Precision-Recall and ROC Curves. *Proceedings Of The 23rd International Conference On Machine Learning* (ss. 233-240), Düzenleyen The International Machine Learning Society. Pittsburgh. 25-29 Temmuz 2006.

Department Of Justice. (2003) *Introduction*. <https://www.justice.gov/archive/jmd/irm/lifecycle/ch1.htm#para1.2>, (24.12.2020).

Dietterich, T.G. (2002). Ensemble Learning. *The Handbook Of Brain Theory and Neural Networks*. (ss. 405-408). Cambridge: The MIT Press.

Elouedi, Z., Mellouli K. ve Smets P. (2000) Classification With Belief Decision Trees. *Artificial Intelligence: Methodology, Systems, and Applications*. (ss.80-90) Berlin: Springer

Faliagka, E., Iliadis, L., Karydis, I., Rigou, M., Sioutas, S., Tsakalidis, A. ve Tzimas G. (2014) On-line Consistent Ranking On E-recruitment: Seeking The Truth Behind A Well-Formed CV. *Artificial Intelligence Review*. 42: 515–528.

Fogg, A. (05.07.2017). *A History of Machine Learning and Deep Learning*.
<https://www.import.io/post/history-of-deep-learning/>, (05.12.2020).

Foote, K.D. (07.02.2017). *A Brief History of Deep Learning*.
<https://www.dataversity.net/brief-history-deep-learning/>, (05.12.2020).

Hastie, T., Tibshirani, R. ve Friedman, J. (2009). *The Elements of Statistical Learning*.
New York: Springer.

James, G., Witten, D., Hastie, T. ve Tibshirani, R. (2013). *An Introduction To Statistical Learning*. New York: Springer.

Jetbrains, (2020). *Download PyCharm*.
<https://www.jetbrains.com/pycharm/download/#section=windows>, (20.12.2020).

Joshi, N. (9 Şubat 2019). *Recruitment Chatbots: Is The Hype Worth It?*
<https://www.forbes.com/sites/cognitiveworld/2019/02/09/recruitment-chatbot-is-the-hype-worth-it/?sh=7eda771d4083>, (26.12.2020).

Jupyter, (2020). *Installing The Jupyter Software*. <https://jupyter.org/install.html>,
(20.12.2020).

Kariyer.net, (2020). *Veri Bilimci Nasıl Olunur?*
<https://www.kariyer.net/pozisyonlar/veri+bilimci/nasil+olunur>, (01.12.2020).

Kavzaoğlu, T. ve Çölkesen, İ. (2010) Destek Vektör Makineleri ile Uydu Görüntülerinin Sınıflandırılmasında Kernel Fonksiyonlarının Etkilerinin İncelenmesi. *Harita Dergisi*. (144): 73-82.

Kaya, İ. ve Gözen, Ş. (2005). Personel Seçim Sürecinde Uzman Sistem Yaklaşımı ve Konya Büyükşehir Belediyesinde Bir Uygulama. *Selçuk Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*. (14): 355-376.

Khanna, S.K. (2019). Machine Learning V/S Deep Learning. *International Research Journal of Engineering and Technology*. 6(2): 455-458.

Kılınç, D., Borandağ, E., Yücalar, F., Özçift, A. ve Bozyiğit, F. (2015). Yazılım Hata Kestiriminde Kolektif Sınıflandırma Modellerinin Etkisi. *IX.Ulusal Yazılım Mühendisliği Sempozyumu Kitabı* (ss.113-121). Düzenleyen Yaşar Üniversitesi. İzmir. 9-11 Eylül 2015.

Kocakafa, T. (26 Şubat 2020). *Aşırı Öğrenme (Overfitting)*. <https://www.veribilimiokulu.com/blog/overfitting/>, (27.12.2020).

Leong, C. (2018). Technology & Recruiting 101: How It Works and Where It's Going. *Strategic HR Review*. 17(1): 50-52.

Lipton, Z.C., Elkan, C. ve Narayanaswamy, B. (2014). Optimal Thresholding Classifiers To Maximize F1 Measure. *Machine Learning and Knowledge Discovery In Databases Part II*. Berlin: Springer.

LinkedIn, (2020). *LinkedIn Recruiter Datasheet*. <https://business.linkedin.com/talent-solutions/recruiter/recruiter-datasheet>, (10.12.2020).

LinkedIn, (27 Mart 2018). *The Rise of Analytics in HR*. https://business.linkedin.com/content/dam/me/business/en-us/talent-solutions/talent-intelligence/workforce/pdfs/Final_v2_NAMER_Rise-of-Analytics-Report.pdf, (10.12.2020).

Loeber, P. (03 Haziran 2020). *Build A Beautiful Machine Learning Web App With Streamlit And Scikit-learn | Python Tutorial*. <https://www.youtube.com/watch?v=Klqn--Mu2pE>, (01.12.2020)

Marr, B. (14 Aralık 2008). *The Amazing Ways How Unilever Uses Artificial Intelligence To Recruit & Train Thousands Of Employees*. <https://bernardmarr.com/default.asp?contentID=1766>, (20.12.2020).

McKinney, W. ve Pandas Geliştirme Ekibi. (30 Ekim 2020). *Pandas: Powerful Python Data Analysis Toolkit*. <https://pandas.pydata.org/docs/pandas.pdf>, (15.11.2020).

Nizam, H. ve AKIN, S.S. (2014). Sosyal Medyada Makine Öğrenmesi ile Duygu Analizinde Dengeli ve Dengesiz Veri Setlerinin Performanslarının Karşılaştırılması. *19.Türkiye'de İnternet Konferansı Bildiriler Kitabı* (ss.129-136), Düzenleyen Yaşar Üniversitesi. İzmir. 27-29 Kasım 2014.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. ve Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal Of Machine Learning Research*. 12(85): 2825–2830.

Polat, S. (2017). Yazılım Hata Kayıtlarının Makine Öğrenmesi Yöntemleriyle Kümelenerek, Hataya Sebep Olan Bileşenlerin Tespit Edilmesi. *11.Ulusal Yazılım Mühendisliği Sempozyumu Kitabı* (ss. 444-453), Düzenleyen Atılım ve Alanya Hamdullah Emin Paşa Üniversitesi. Alanya. 18-20 Ekim 2017.

Polikar, R. (2012). Ensemble Learning. *Ensemble Machine Learning*. (ss.1-34). Boston: Springer.

Portilla J. (2020). *Python for Data Science and Machine Learning Bootcamp*. <https://www.udemy.com/course/python-for-data-science-and-machine-learning-bootcamp/>, (20.12.2020).

Samuel, A.L. (1959). Some Studies in Machine Learning Using The Game of Checkers. *IBM Journal of Research and Development*. 3(3): 211-229.

Sathya, R. ve Abraham, A. (2013). Comparison of Supervised and Unsupervised Learning Algorithms for Pattern Classification. *International Journal of Advanced Research in Artificial Intelligence*. 2(2): 34-38.

Scikit-learn, (2020a). *Nearest Neighbors Classification*, https://scikit-learn.org/stable/auto_examples/neighbors/plot_classification.html#sphx-glr-auto-examples-neighbors-plot-classification-py, (20.12.2020).

Scikit-learn, (2020b). *sklearn.linear_model.LogisticRegression*, https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html, (20.12.2020).

Scikit-learn, (2020c). *sklearn.neighbors.KNeighborsClassifier*, <https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html>, (20.12.2020).

Scikit-learn, (2020d). *sklearn.svm.SVC*, <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html>, (20.12.2020).

Scikit-learn, (2020e). *sklearn.tree.DecisionTreeClassifier*, <https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html>, (20.12.2020).

Scikit-learn, (2020f). *sklearn.ensemble.RandomForestClassifier*, <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>, (20.12.2020).

Sokolova, M. ve Lapalme, G. (2009). A Systematic Analysis Of Performance Measures For Classification Tasks. *Information Processing & Management*. 45(5):427-437.

Sönmez, H. (2012). Regresyon ve Korelasyon Çözümlemesi. *İstatistik*. (ss. 227-247). Eskişehir: Anadolu Üniversitesi.

Steinwart, I. ve Christmann, A. (2008) *Support Vector Machines*. New York: Springer.

Streamlit, (2020). *Welcome To Streamlit*. <https://docs.streamlit.io/en/stable/>, (20.12.2020).

Şeker, A., Diri, B. ve Balık, H.H. (2017). Derin Öğrenme Yöntemleri ve Uygulamaları Hakkında Bir İnceleme. *Gazi Mühendislik Bilimleri Dergisi*. 3(3): 47-64.

Sutton, R.S. ve Barto, A.G. (2018). *Reinforcement Learning: An Introduction*. Cambridge: The MIT Press.

Strohmeier, S. ve Piazza, F. (2015). Artificial Intelligence Techniques In Human Resource Management-A Conceptual Exploration. *Intelligent Techniques in Engineering Management*. (ss.149-169) Cham: Springer.

Tableau, (2020). *Tableau Desktop*.
<https://www.tableau.com/products/desktop/download>, (20.12.2020).

Taşar, B., Üneş, F., Demirci, M. ve Kaya Y.Z. (2017). Yapay Sinir Ağları Yöntemi Kullanılarak Buharlaştırma Miktarı Tahmini. *DÜMF Mühendislik Dergisi*. 9(1): 543-551.

Tecim, V. (2012). *Sistem Geliştirme Yaşam Döngüsü*.
<http://debis.deu.edu.tr/userweb//vahap.tecim/dosyalar/sgyd.pdf>, (20.12.2020).

Tonus H.Z. (2018). İnsan Kaynağı Bulma ve Seçme. *İnsan Kaynakları Yönetimi* (ss.63-70). Eskişehir: Anadolu Üniversitesi.

Turing A.M. (1950). Computing Machinery and Intelligence. *Mind*. 59(236): 433-460.

Türkiye Büyük Millet Meclisi, (07 Nisan 2016). Kişisel Verilerin Korunması Kanunu. *Resmi Gazete*. <https://www.resmigazete.gov.tr/eskiler/2016/04/20160407.pdf>, (20.11.2020).

Türkiye İstatistik Kurumu, (2020). *İl ve İlçelere Göre İl/İlçe Merkezi, Belde/Köy Nüfusu ve Yıllık Nüfus Artış Hızı 2019*.
<https://data.tuik.gov.tr/Bulten/DownloadIstatistikselTablo?p=0sulHkbXDJQtDqwXbWBPArErCfGJ54pigHlgegwwzRdIoG2V4RRDPz6In9Hd0eaNj>, (25.11.2020)

Türkiye İstatistik Kurumu, (2020). *Bitirilen Eğitim Durumuna ve Cinsiyete Göre Nüfus 2008-2019*.
<https://data.tuik.gov.tr/Bulten/DownloadIstatistikselTablo?p=NLpBAx1kMG2i5UnOCazV6mRjabovcRMFf/kUgigNi9SXYKX7waXPMQgo81Q7U7th>, (26.11.2020).

Upadhyay, A.K. ve Khandelwal, K. (2018). Applying Artificial Intelligence: Implications For Recruitment. *Strategic HR Review*. 17(5): 255-258.

Willmot C.J. ve Matsuura K. (2005). Advantages Of The Mean Absolute Error (MAE) Over The Root Mean Square Error (RMSE) In Assessing Average Model Performance. *Climate Research*. 30: 79-82.

XGBoost, (2020). *Python API Reference*.
https://xgboost.readthedocs.io/en/latest/python/python_api.html#module-xgboost.sklearn, (23.12.2020).