



AKSARAY ÜNİVERSİTESİ

SOSYAL BİLİMLER ENSTİTÜSÜ

YÖNETİM BİLİŞİM SİSTEMLERİ ANABİLİM DALI

**BİLİŞİM TEKNOLOJİLERİ ÖĞRETMENLERİNİN SCRATCH
YAZILIMINA İLİŞKİN ÖZYETERLİK İNANÇLARININ MAKİNE
ÖĞRENMESİ VE DERİN ÖĞRENME YÖNTEMLERİ İLE
SINIFLANDIRILMASI**

YÜKSEK LİSANS TEZİ

BURAK KOCA

DANIŞMAN

DOÇ. DR. KEMAL ADEM

AKSARAY 2022

AKSARAY ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
YÖNETİM BİLİŞİM SİSTEMLERİ ANABİLİM DALI

**BİLİŞİM TEKNOLOJİLERİ ÖĞRETMENLERİNİN SCRATCH
YAZILIMINA İLİŞKİN ÖZYETERLİK İNANÇLARININ MAKİNE
ÖĞRENMESİ VE DERİN ÖĞRENME YÖNTEMLERİ İLE
SINIFLANDIRILMASI**

YÜKSEK LİSANS TEZİ

BURAK KOCA

DANIŞMAN
DOÇ. DR. KEMAL ADEM

AKSARAY 2022

TELİF HAKKI VE TEZ FOTOKOPİ İZİN FORMU

Bu tezin tüm hakları saklıdır. Kaynak göstermek koşuluyla tezin teslim tarihinden itibaren(....) ay sonra tezden fotokopi çekilebilir.

YAZARIN

Adı : Burak

Soyadı : KOCA

Bölümü : Yönetim Bilişim Sistemleri

İmza :

Teslim tarihi :

TEZİN

Türkçe Adı: Bilişim Teknolojileri Öğretmenlerinin Scratch Yazılımına İlişkin Özyeterlik İnançlarının Makine Öğrenmesi ve Derin Öğrenme Yöntemleri ile Sınıflandırılması

İngilizce Adı: Classification of Self Efficacy Beliefs of Information Technology Teachers About Scratch Software by Machine Learning and Deep Learning Methods

ETİK İLKELERE UYGUNLUK BEYANI

Tez yazma sürecinde bilimsel ve etik ilkelere uyduğumu, yararlandığım tüm kaynakları kaynak gösterme ilkelerine uygun olarak kaynakçada belirttiğimi ve bu bölümler dışındaki tüm ifadelerin şahsıma ait olduğunu beyan ederim.

Yazar Adı Soyadı: Burak KOCA

İmza:

T.C.
AKSARAY ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
JÜRİ ONAY SAYFASI

Burak KOCA tarafından hazırlanan “Bilişim Teknolojileri Öğretmenlerinin Scratch Yazılımına İlişkin Özyeterlik İnançlarının Makine Öğrenmesi ve Derin Öğrenme Yöntemleri ile Sınıflandırılması” başlıklı tez çalışması aşağıdaki jüri tarafından oy birliği ile Aksaray Üniversitesi Yönetim Bilişim Sistemleri Anabilim Dalı’nda Yüksek Lisans Tezi olarak kabul edilmiştir.

Danışman Doç. Dr. Kemal ADEM

İmza

Bilgisayar Mühendisliği/Sivas Bilim ve Teknoloji Üniversitesi

.....

Üye: Prof. Dr. Zeliha SEÇKİN

Yönetim Bilişim Sistemleri/Aksaray Üniversitesi

.....

Üye: Dr. Öğür. Üyesi Cengiz COŞKUN

Bilgisayar Mühendisliği/Sivas Bilim ve Teknoloji Üniversitesi

.....

Tez Savunma Tarihi: 09/09/2022

Sosyal Bilimler Enstitüsü Yönetim Kurulu’nun tarih ve sayılı kararı ile onaylanmıştır.

Sosyal Bilimler Enstitüsü Müdürü

Doç. Dr. Sevilay USLU DİVANOĞLU

İmza

TEŞEKKÜR

Bu çalışmanın gerçekleşmesinde yardımlarını hiçbir zaman esirgemeyen ve yol gösteren danışman hocam Sayın Doç. Dr. Kemal Adem'e, ölçek uyarlama sürecinde katkılarından dolayı Sayın Doç. Dr. Alev ELÇİ'ye ve Ayşe Gamze ÇAM'a teşekkürleri sunarım. En büyük teşekkür ise tezimin her aşamasında desteklerini hiçbir zaman esirgemeyen biricik eşim Esra KOCA'ya teşekkür ediyorum Bu tez çalışmasını eşim ve biricik oğlum İsmail Ege'ye atfediyorum.

2022

Burak KOCA

AKSARAY ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
BİLİŞİM TEKNOLOJİLERİ ÖĞRETMENLERİNİN SCRATCH
YAZILIMINA İLİŞKİN ÖZYETERLİK İNANÇLARININ MAKİNE
ÖĞRENMESİ VE DERİN ÖĞRENME YÖNTEMLERİ İLE
SINIFLANDIRILMASI
(Yüksek Lisans Tezi)

Burak KOCA, Aksaray 2022

ÖZET

Teknolojinin gelişmesiyle bireylerin sahip olması beklenen beceriler de değişime uğramıştır. Bilgi işlemsel düşünme becerisi 21.yy. becerileri içerisinde en önemli becerilerdendir. Bu becerinin bireylere kazandırılabilmesi için nitelikli bir kodlama eğitime ihtiyaç vardır. Bu düzlemde MEB bilişim teknolojileri öğretim programı kapsamına kodlama eğitimini dahil etmiştir. Program kapsamında temel düzey blok tabanlı kodlama araçlarından Scratch kullanılarak bu eğitim verilmektedir. Bu sebeple bilişim teknolojileri öğretmenlerinin kodlama araçları konusunda kendilerini ne kadar yeterli gördükleri son derece önemlidir. Bu bağlamda, bu çalışmada bilişim teknolojileri öğretmenlerinin Scratch'e yönelik kullanımına ilişkin öz yeterlik inançlarının klasik makine öğrenmesi ve derin öğrenme yöntemleri ile sınıflandırılması amaçlanmıştır. Veri toplama aracı olarak ise daha önceden geliştirilmiş T-SECT anketi kullanılmıştır. Bu ölçek likert tipinde 39 maddeden oluşmaktadır. Bilişim teknolojileri öğretmenlerinden oluşan 192 örnek ve 39 öznitelik kullanılarak veri seti oluşturulmuştur. Örnek sayısının sınırlılığı nedeniyle dengesiz veri sorununu ortadan kaldırmak için SMOTE yöntemi ile veri çoğaltılmış ve örnek sayısı 262 ye yükseltilmiştir. Veri seti WEKA yazılımına aktararak üzerinde klasik makine öğrenmesi yöntemleri ve Google

Colab platformu ile Evriřimli Sinir Ađı (ESA) yöntemleri kullanılmıřtır. Bu amaç dođrultusunda Zero-R, J48, Random Forest (RF), K-Star, Multilayer-Perceptron (MLP), NaivesBayes, SMO, Logistic, IBK ve Random Tree yöntemleri ile sınıflandırma başarıısı hesaplanmıřtır. Veri seti üzerinde en yüksek sınıflandırma performansı elde edilen klasik makine öğrenmesi yöntemleri SMO, MLP, IBK, J48 ve RF olarak bulunmuřtur. ESA ile yapılan sınıflandırma klasik makine öğrenmesi yöntemlerinden daha başarılı sonuç vermiřtir. Biliřim teknolojileri öğretmenlerinin öz yeterlik inançları ESA kullanılarak %99.30 dođruluk oranıyla başarılı bir biçimde sınıflandırılmıřtır.

Bilim Kodu: 92431, 92432, 11302

Anahtar Kelimeler: Bilgi işlemsel düşünme, Evriřimli Sinir Ađları, Makine öğrenmesi, Özyeterlik inancı, Scratch

Sayfa Adedi: 92

Danışman: Doç. Dr. Kemal ADEM

AKSARAY UNIVERSITY
GRADUATE SCHOOL OF SOCIAL SCIENCES
CLASSIFICATION OF SELF EFFICACY BELIEFS OF
INFORMATION TECHNOLOGY TEACHERS ABOUT SCRATCH
SOFTWARE BY MACHINE LEARNING AND DEEP LEARNING
METHODS
(M.A. Thesis)
Burak KOCA, Aksaray 2022

ABSTRACT

With the development of technology, the skills that are expected from individuals have also changed. Computational thinking is one of the most important skills of 21st century. A qualified coding education is needed in order to upskill the individuals for computational thinking. Concordantly, the Ministry of National Education has included coding education in the scope of the information technologies curriculum. In this context, this training is given by using Scratch which is one of the basic block-based coding tools. Therefore, it is extremely important how competent Information Technologies teachers consider themselves about coding tools. In this study, it is aimed to classify the self-efficacy beliefs of Information Technologies teachers regarding the use of Scratch by using classical machine learning and deep learning methods. T-SECT scale which was previously developed was used as a data collection tool. This scale consists of 39 Likert type items. The data set was created by using 192 samples consisting of Information Technology teachers and 39 attributes. In order to eliminate the problem of unbalanced data due to the limited number of samples, the data was amplified by the SMOTE method and the number of samples was increased to 262. The data set was transferred into WEKA software and classical machine learning methods,

Google Colab platform and Convolutional Neural Network (CNN) methods were used on it. In line with this purpose, classification success was calculated with Zero-R, J48, Random Forest(RF), K-Star, Multilayer-Perceptron(MLP), NaivesBayes, SMO, Logistic, IBK and Random Tree methods. The classical machine learning methods with the highest classification performance on the data set were found as SMO, MLP, IBK, J48 and RF. Classification which is made with CNN received better results than classical machine learning methods. Self-efficacy beliefs of Information Technologies teachers were successfully classified using CNN with an accuracy rate of 99.30%.

Science Code: 92431, 92432, 11302

Keywords: Computational thinking , Convolutional Neural Networks , Machine learning, Scratch, , Self-efficacy belief.

Page Number: 92

Supervisor: Doç. Dr. Kemal ADEM

İÇİNDEKİLER

TELİF HAKKI VE TEZ FOTOKOPİ İZİN FORMU	i
ETİK İLKELERE UYGUNLUK BEYANI.....	ii
JÜRİ ONAY SAYFASI.....	iii
TEŞEKKÜR.....	iv
ÖZET	v
ABSTRACT.....	vii
TABLolar LİSTESİ.....	xii
ŞEKİLLER LİSTESİ	xiii
SİMGELER VE KISALTMALAR LİSTESİ	xv
1. GİRİŞ	1
1.1 Araştırmanın Önemi.....	2
2. KURAMSAL TEMELLER	4
2.1 Kodlama Nedir	4
2.2 Kodlama Eğitimi	5
2.3 Blok Tabanlı Kodlama Araçları.....	6
2.3.1 Scratch Nedir?.....	6
2.4 Özyeterlik.....	7
2.5 Kaynak Araştırması	7
3. MATERYAL VE YÖNTEM.....	12
3.1 Veri Setinin Oluşturulması	12
3.2 Synthetic Minority Oversampling Technique (SMOTE)	13
3.3 Yapay Zekâ	14
3.4 Makine Öğrenmesi.....	16

3.4.1 Denetimli Makine Öğrenmesi.....	18
3.4.2 Yarı Denetimli Makine Öğrenmesi.....	18
3.4.3 Denetimsiz Makine Öğrenmesi.....	19
3.4.4 Takviyeli (Pekiştirmeli) Makine Öğrenmesi	19
3.4.5 Makine Öğrenmesi Algoritmaları	20
3.4.6 Kullanılan Klasik Makine Öğrenmesi Algoritmaları.....	22
3.5 Derin Öğrenme.....	29
3.5.1 Evrişimli Sinir Ağı (ESA).....	29
3.6 Değerlendirme Ölçütleri	35
3.6.1 Değerlendirme Kriterleri.....	35
3.7 Geliştirme Ortamları	38
3.7.1 Çalışmada Kullanılan IDE/Programlama Dili/Kütüphane.....	38
4. BULGULAR.....	40
4.1 SMO Parametreleri ve Deneysel Sonuçları	41
4.2 J48 Parametreleri ve Deneysel Sonuçları	42
4.3 MLP Parametreleri ve Deneysel Sonuçları.....	43
4.4 Random Forest Parametreleri ve Deneysel Sonuçları	45
4.5 IBK Matrisi	46
4.6. Evrişimli Sinir Ağı Deneysel Sonuçları ve Parametreler	47
5. SONUÇ	51
6. KAYNAKÇA.....	53
EKLER.....	64
TEZ DEĞERLENDİRME FORMU	73



TABLÖLAR LİSTESİ

Tablo 1. Özniteliklere ait bilgiler	13
Tablo 2. Karmaşıklık (Hata) Matrisi	36
Tablo 3. Makine Öğrenmesi Yöntemlerinin Deneysel Sonuçları	40
Tablo 4. SMO Parametreleri Değerleri	41
Tablo 5. SMO Karmaşıklık Matrisi.....	42
Tablo 6. J48 Parametre Değerleri.....	43
Tablo 7. J48 Karmaşıklık (Hata) Matrisi	43
Tablo 8. MLP Parametre Değerleri	44
Tablo 9. MLP Karmaşıklık (Hata) Matrisi.....	45
Tablo 10. Random Forest Parametre Değerleri.....	45
Tablo 11. Random Forest Karmaşıklık(Hata) Matrisi	46
Tablo 12. IBK Parametre Değerleri	47
Tablo 13. IBK Karmaşıklık(Hata) Matrisi	47
Tablo 14. Evrişimli Sinir Ağı Deneysel Sonuçları.....	49
Tablo 15. Klasik makine öğrenmesi yöntemleri ve Evrişimli Sinir Ağı Sonuçları.....	49

ŞEKİLLER LİSTESİ

Şekil 1. Kodlama Türleri	4
Şekil 2. Yapay zekâ gelişim kronolojisi (Sargın ve Göçen, 2020)	15
Şekil 3. Yapay Zekâ Çalışma Alanları	15
Şekil 4. Yapay Zekâ, Makine Öğrenmesi ve Derin Öğrenme ilişkisi (Güngör ve Yardımcı, 2020)	16
Şekil 5. Makine Öğrenmesi Yöntemleri	17
Şekil 6. Denetimli Makine Öğrenmesi (Turkhost, 2021)	18
Şekil 7. Yarı Denetimli Makine Öğrenmesi (Turkhost, 2021)	18
Şekil 8. Yarı Denetimsiz Makine Öğrenmesi (Turkhost, 2021)	19
Şekil 9. Takviyeli (Pekiştirmeli) Makine Öğrenmesi (Turkhost, 2021)	20
Şekil 10. Makine Öğrenmesi Algoritmaları (Cankıran,2019)	21
Şekil 11. SMO Algoritması	22
Şekil 12. Çok katmanlı ileri beslemeli yapay sinir ağı (Kaynar ve Taştan, 2009).	23
Şekil 13. Random Forest Algoritması (Kirasich, Smith ve Sadler, 2018).....	26
Şekil 14. One-R Algoritması (Holte, 1993).....	28
Şekil 15. Evrişimli Sinir Ağı Mimarisi (Sayracı, 2021)	30
Şekil 16. Evrişim Katmanı (Kızrak, 2018)	31
Şekil 17. Havuzlama (Pooling) Katmanı (Kızrak, 2018)	32
Şekil 18. Tam Bağlantılı Katman (Kızrak, 2018).....	33
Şekil 19. Aktivasyon Fonksiyonları Girdi ve Çıktı Değerleri (Karakuş, 2018)	35
Şekil 20. Çapraz Geçerlilik Prensipleri.....	38

Şekil 21. Çalışmada Kullanılan Makine Öğrenmesi ve ESA yöntemlerinin ROC eğrisi....50



SİMGELER VE KISALTMALAR LİSTESİ

ESA	Evrişimli Sinir Ağı
CNN	Convolutional Neural Network
SMO	Sıralı Minumum Optimizasyon
MLP	MultilayerPerceptron(Çok Katmanlı Algılayıcı
RF	Random Forest
MEB	Milli Eğitim Bakanlığı
PISA	Uluslararası Öğrenci Değerlendirme Programı
SMOTE	Synthetic Minority Oversampling Technique
ROC	Receiver Operating Characteristic
DT	Karar Ağacı
kNN	K-En Yakın Komşu
NB	Naive Bayes
SVM	Destek Vektör Makinesi
YÖK	Yüksek Öğretim Kurumu
TYT	Temel Yeterlilik Testi
WEKA	Waikato Environment for Knowledge Analysis
YSA	Yapay Sinir Ağları
USBS	Uyarlamalı Sinir-Bulanık Sınıflayıcı
MATLAB	Matrix Laboratory
TSECT	Teachers' Self-Efficacy in Computational Thinking
ReLU	Doğrultulmuş Doğrusal Aktivasyon Fonksiyonu
<i>f</i>	Aktivasyon fonksiyonu
OneR	Very Simple Classification Rules
TP/DP	Doğru Pozitif
TN/DN	Doğru Negatif
FP/YP	Yanlış Pozitif

FN/YN	Yanlış Negatif
ARFF	Attribute-Relation File Format
CVS	Comma-separated values
XLS	File name extension of MS Excel
CPU	Central Process Unit
GPU	Graphics Processing Unit
RAM	Random Access Memory



1. GİRİŞ

Teknolojide meydana gelen hızlı deęişimler başta ekonomi olmak üzere saęlık, eęitim, üretim gibi pek çok alanda etkili olmuştur (Ersoy ve Özden, 2011). Bahsedilen alanlarda teknolojinin daha çok kendine yer bulması insanların daha farklı becerilere sahip olmasını zorunlu kılmıştır. Bu beceriler son yıllarda 21. yy. becerileri olarak adlandırılmaya başlanmıştır. 21. yy. becerileri çocuklarımızı geleceęe nasıl hazırlamamız gerektiğini ifade etmektedir (Sing, 1991). Problem çözme, bilgi-işlemsel düşünme, yaratıcılık, varsayımsal düşünme, eleştirel düşünme, girişimcilik ve iletişim gibi üst bilişsel beceriler 21. yy. becerileri kapsamında yer almaktadır.

Geleceğin iş gücünü oluşturacak 21. yy. becerilerine sahip bireylerin ekonomik kalkınmada kilit rol oynayacağı bir gerçektir. Ekonomik kalkınmanın gereklilięi eğitimle ilgili politikalar üzerinde de deęişimi zorunlu kılmaktadır. Küreselleşme ve küresel işgücü niteliğinin önemli bir hale gelmesi eğitim-öğretim süreçlerinde teknolojinin sıklıkla işe koşulmasını gerekli bir hale getirmiştir (Sayın ve Seferoęlu, 2016). Sanayi, savunma, saęlık gibi önemli alanlarda bilgisayar tabanlı teknolojilerin kullanılması, eğitim-öğretim sürecinde bilgi işlemsel düşünme becerilerine daha çok önem verilmesini gerektirmektedir. Bu bağlamda bilgi işlemsel düşünmenin ana basamaklarından olan kodlama eğitimi öne çıkmaktadır (Sayın ve Seferoęlu, 2016).

Pek çok ülke bilgi-işlemsel düşünmenin; bireye kazandıracığı beceriler, toplumsal fayda ve ekonomik katkı gibi önemli çıktılarının olması sebebiyle ders içeriklerine seçmeli/zorunlu olarak kodlama eğitimini dahil etmişlerdir (Bocconi, Chiocciariello, Dettori, Ferrari ve Engelhardt, 2016; Göncü, Çetin ve Top, 2018). Ülkemizde de bilişim teknolojileri ve yazılımı öğretim programında revizyona gidilerek birçok kademede kodlama eğitime yer vermeye başlanmıştır. Öğretim programıyla hedeflenen kodlama becerilerini öğrencilerin kazanabilmeleri için öğretmenlerin kodlamanın gerektirdiğı alan bilgisine ve pedagojik bilgiye ihtiyaçları vardır (Göncü, Çetin ve Şendurur, 2020).

Literatürde yer alan çalışmalarda, bilişim teknolojileri öğretmenlerinin büyük bir kısmı kodlama eğitimi hakkında kendilerini yeterli bulmamakta veya sadece temel seviyede yeterli gördüklerini düşünmektedirler (Türker ve Pala, 2018). Bunun sebebi öğretmenlerin lisans eğitiminde kodlama eğitimi ile ilgili eğitim almamaları veya geçerliliğini yitirmiş programlama dillerinde eğitim almış olmaları olabilir (Ozoran, Çağıltay ve Topalli, 2012). Öğretmenlerin düşüncelerine ilişkin çalışmalar literatürde yer almaktadır ancak kodlamanın gerektirdiği beceriler ile bu becerileri öğrencilere öğretme konusunda sahip olmaları gereken yeterliklere ne ölçüde sahip olduklarına ilişkin az sayıda çalışma bulunmaktadır. Özellikle bilişim teknolojileri öğretim programında yer alan kodlama araçlarından blok tabanlı araçları kullanabilme becerilerine ne derecede sahip oldukları bu konunun öğretimi açısından son derece önemlidir. Bu doğrultuda bu çalışmada bilişim teknolojileri öğretmenlerinin özyeterlik düzeylerinin belirlenmesi amaçlanmaktadır.

Öğretmenlerin blok tabanlı kodlama araçlarına ilişkin özyeterlik düzeylerinin sınıflandırılmasında derin öğrenme ve klasik makine öğrenmesi yöntemlerine başvurulmuştur. Makine öğrenmesi çok büyük verileri analiz etme ve yüksek seviye bilgileri ayıklama işlemlerinde kullanılmaktadır (Breiman, 2001). Derin öğrenme, insan beynine benzer şekilde kompleks problemleri çözmek için kullandığı yöntem ve yeteneklerini taklit eden, büyük miktarda veriden yararlanarak öznitelik çıkarma, sınıflandırma ve dönüştürme işlemlerini gerçekleştiren bir makine öğrenmesi tekniğidir (Albannai, 2019). Makine öğrenmesi ve derin öğrenme yöntemleri eğitimde sınıflandırma ve tahmin etme amaçlı kullanılabilir.

Bu çalışmada bilişim teknolojileri öğretmenlerinin Blok tabanlı programlama araçlarına yönelik özyeterlik düzeylerinin klasik makine öğrenmesi ve derin öğrenme yöntemleriyle sınıflandırılması amaçlanmaktadır.

1.1 Araştırmanın Önemi

21. yüzyılda oluşan teknolojik ve ekonomik değişimler her alanda kişilerin yeni becerilerle donanmış olmasını gerekli kılmıştır. Bilgi işlemsel düşünme becerisi bu beceriler arasında önemli yere sahiptir. Bilgi işlemsel düşünme becerisi bağlamında pek çok ülkenin eğitim sisteminde kodlama eğitimine yer vermeye başlanmıştır. Kodlama yapabilme becerisi 21. yüzyılda herkesin sahip olunması beklenen beceriler arasında yer almaktadır (Aytekin, Çakır, Yücel ve Kulaöz, 2018) .

Ülkelerin içerisinde bulunduğu küresel rekabet kodlama alanında da ülkeleri en iyi olmaya teşvik etmektedir. Bu sebeple gelişmiş ve gelişmekte olan ülkeler eğitim sistemine farklı seviyelerde farklı dersler kapsamında kodlama eğitimini entegre etmeye çalışmaktadırlar (Tucker vd., 2003). Kodlamanın erken yaşlarda öğretilmesinin bireye bilgi işlemsel düşünme becerisi anlamında büyük oranda katkı sağlayacağı düşünülmektedir (Yükseltürk ve Altıok, 2016).

Eğitim kurumlarında programların uygulayıcısı olan öğretmenlerin kodlama konusunda donanımlı ve kodlamanın öğretiminde izlenecek adımlar konusunda deneyimli olması önemlidir (Göncü, Çetin ve Top, 2018). Ülkemizde de ortaokul ve lise düzeyinde bilişim teknolojileri ve yazılım dersi öğretim programına kodlama eğitimi eklenmiştir (MEB). Ancak kodlama çoğu bilişim teknolojileri öğretmenlerin lisans eğitimlerinde karşılaşmadıkları bir konudur. Bu bağlamda öğretmenler hizmet içi eğitimlerle ve kendi kişisel çabalarıyla kodlama konusunda kendilerini geliştirmeye çalışmaktadırlar (Tekerek, Ercan, Udum ve Saman, 2012). Öğretmenlerin kendilerini yeterli gördükleri alanda daha rahat öğretim yapabilecekleri açıktır. Bu çalışma ile bilişim teknolojileri öğretmenlerinin blok tabanlı kodlama araçlarına yönelik özyeterlik düzeylerinin yeterliliği belirlenmek istenmiştir. Bu amaçla TSECT anketi yardımıyla toplanan veriler, derin öğrenme yöntemlerinden Evrişimli Sinir Ağı (ESA) ve klasik makine öğrenmesi yöntemleri ile öğretmenlerin özyeterlik düzeyleri sınıflandırılmaya çalışılmıştır. Literatür incelendiğinde Evrişimli Sinir Ağları kullanılarak genellikle görüntü işleme ve metin üzerine çalışıldığı görülmüştür. Çalışmada kullandığımız 5’li likert tipinde ölçek ile özyeterlik düzeyi belirlenmeye çalışılmıştır. Bu sonuçlar ile ilgili literatürde ESA çalışmasına rastlanmamıştır. Bu çalışmanın literatüre bu açıdan katkı sağlayacağı düşünülmektedir.

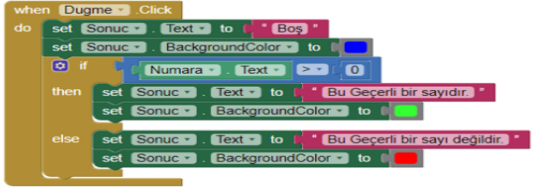
Araştırmacı Pandemi sürecinde yüz yüze etkileşim imkânı bulunmadığından veriler toplanırken dijital ortamlar kullanmak zorunda kalmıştır. Bu durum örneklemin az sayıda olmasına sebep olmuştur.

2. KURAMSAL TEMELLER

2.1 Kodlama Nedir

Kodlama, elektronik ve mekanik cihazlar, bilgisayar, telefon, tablet ve diğer cihazlar ile insanlar arasında iletişim sağlamak ve bunun sonucunda verilen görevleri adım adım yaptırmak için yazılan komutlar serisinin tamamıdır (Sayın ve Seferoğlu, 2016). Başka bir deyişle kodlama, kullanıcı ile bilgisayar arasında bir veya birden fazla problemi çözmek için programlama dilleri, algoritma yöntemleri, kodlar ve kod blokları kullanılarak yazılan programlar bütünüdür (Akçay ve Çoklar, 2016; Sterling, 2015)

Kodlamada öncelikle bir veya birden fazla problem esas alınmaktadır. Bu problemin çözümü için bilgisayar kullanıcıya geri dönüt vermek zorundadır. Bu geri dönütü vermek için kodlama ile yazılmış programlar vasıtasıyla iletişim kurulmaktadır. Problemin çözümü için algoritma yöntemlerini kullanarak adım adım program çözülür. Daha sonra herhangi bir programlama dili yardımı ile kodlama yapılır ve probleme çözüm bulunur (Sterling, 2015). Şekil 1’de görüldüğü gibi kodlama araçları metin tabanlı ve blok tabanlı olmak üzere ikiye ayrılır. Metin tabanlı kodlamada programı yazan kişiler kullandıkları dile uygun olarak programı metin şeklinde yazarlar. Blok tabanlı kodlamada ise her koda karşılık gelen bloklar vardır. Bu bloklar sürekli bir mantığıyla sıraya koyularak program oluşturulur.

<pre>#include<stdio.h> #include<conio.h> main() { int a,b,c; printf("Birinci sayiyi giriniz "); scanf("%d",&a); printf("ikinci sayiyi giriniz "); scanf("%d",&b); c=a+b; printf("Sayilarin Toplami = %d",c); getch(); }</pre>	
Metin Kodlama	Blok Kodlama

Şekil 1. Kodlama Türleri

2.2 Kodlama Eğitimi

Bilgi işlemsel düşünme içinde bulunduğumuz yüzyılın ekonomik ve endüstriyel gerekliliklerinden biri haline gelmiştir (Gander, 2013). Bilgi işlemsel düşünme becerisi, bilişim kavramlarından faydalanan bir probleme çözüm bulma, tasarım yapma ve insan davranışlarını algılama olarak ifade edilebilir (Gülbahar, Kert & Kalelioğlu, 2018). Bilgi işlemsel düşünme becerisini kazandırmada kodlama eğitimi ön planda tutulmaktadır. Kodlama eğitiminin ayrıca bireylere bilgi işlemsel düşünme becerisine temel oluşturacak üstbilişsel düşünme, analitik düşünme, yaratıcılık gibi pek çok becerinin kazanılmasında da etkilidir (Çetin, 2012). Bu bağlamda pek çok ülke kodlama eğitimine son derece önem vermektedir.

Geçmişte de tüm dünyada kodlama eğitimi ile ilgili farklı seviyelerde yapılan araştırmalar, kodlama programlarında yer alan kuralların soyut olması ve kullanılan programlama dilinin İngilizce olması nedeniyle kodlamanın zor öğrenildiğini ortaya koymuştur (Bayman ve Mayer, 1983; Özmen ve Altun, 2014). Geleneksel programlama dillerinin öğrenilmesinde yaşanan bu sıkıntılar çocuktan yetişkine herkesin kullanabileceği yeni nesil araçların kullanılmasını zorunlu kılmıştır. Dünya genelinde soyut kuralları bulunan geleneksel kodlama araçlarının yerini öğrenilmesi daha kolay olan blok tabanlı kodlama araçları almıştır (Çatlak, Tekdal ve Baz, 2015). Blok tabanlı kodlama araçları ile sürükleyip bırak yöntemi ile kodlama yapılabilmekte ayrıca ses, resim gibi medyalar da projelere eklenebilmektedir (Sırakaya, 2018).

Ülkemizde ve tüm dünyada kodlama eğitiminin üzerinde durulmakta ve öğretim programları revize ederek kodlama eğitimi programlara dahil edilmektedir (Bers, Flannery, Kazakoff ve Sullivan, 2014; Kalelioğlu, Gülbahar, Akçay ve Doğan, 2014). Ülkemizde kodlama eğitimine 2012 yılında alınan kararla Bilişim teknolojileri dersinin kapsamına yazılım konularının eklenmesi ile ilk adım atılmıştır (Alp, 2019). Bunun sonucunda dersin adı Bilişim teknolojileri ve yazılım olarak değişmiştir. 2017 yılında güncellenen öğretim programı ile blok tabanlı kodlama araçlarına ağırlık verilecek şekilde kapsamına alınmıştır (MEB, 2017).

Ülkeler gelişen çağa ayak uydurmak amacıyla kodlama eğitimini küçük yaşlardan itibaren vermeye başlamışlardır (Demirer ve Sak, 2016). Kuzey ülkelerinden Estonya 2012 yılında 7 yaş ve üzeri çocuklara kodlama eğitimi vermeye başlamıştır. Güney Kore 2017 yılında ilkokulda ve 2018'de liselerde yazılım derslerini zorunlu tutacağını belirtmiştir. İngiltere

ise mevcut bilişim teknolojileri dersinin içeriğinin sıkıcı olduğunu ve çağın gereklerine uymadığını belirtmiştir (Burns, 2012). Bu ders içeriğinin ülke ekonomisine zarar verebileceği bildirilmiştir. Çağın gereklerine uygun olarak bilişim teknolojileri dersin içeriğine kodlama eğitiminin ve gerçek hayata uygun simülasyonların ekleneceğini duyurmuştur. Ayrıca 2014 yılını kodlama yılı ilan ederek öğrencilerde kodlama farkındalığını yükseltmeyi amaçlamışlardır (Demirer ve Sak, 2016).

2.3 Blok Tabanlı Kodlama Araçları

Öğrencilere kodlama eğitimi verilirken programlama araçlarının yanlış bir şekilde kullanılması öğrencilerin programlamaya karşı ön yargılı olmasına neden olmaktadır (Ünsal, 2020). Bu ön yargının oluşmaması için her yaş seviyesine uygun bir programlama aracının seçilmesi gereklidir. Kodlamayı öğrencilere kolay bir şekilde kazandırmak için basit programlama araçları kullanılmalıdır.

Programlama metin tabanlı (text-based) ve blok tabanlı (block-based) olarak ikiye ayrılmaktadır. Kodlama eğitimi yeni bir dil öğrenmeye benzetildiğinden dolayı öğrenciler ilk başta ön yargı ile yaklaşabilmektedir. Bunun için ilk etapta kodlamada görsel ağırlıklı programlama araçlarının kullanılması tercih edilir. Blok tabanlı programlama araçları öğrenmeyi zevkli hale getirirken aynı zamanda öğrencilerin motivasyonunu artırır. Soyut kavramlar blok tabanlı araçlarla görselleştirilerek kodlamaya yeni başlayan herkes için programlamayı kolaylaştırmaktadır (Saygıner, 2017). Özellikle ilköğretim seviyesinde kod yazmak yerine blok kullanmak daha kolaydır. Blok tabanlı araçlar öğrencilerin somutlaştırılmadığı bazı durumları ve problemleri görselleştirilerek anlamasını sağlarlar. Kod yazmayı daha eğlenceli ve anlaşılır hale getirir.

Başlıca blok tabanlı kodlama araçları arasında Scratch, Alice, Code.org ve kodu game lab yer almaktadır. Bu araçların en önemli özellikleri de ücretsiz olması ve her seviyede insanın kodlama öğrenmesini kolaylaştıracak görsellerin bulunmasıdır.

2.3.1 Scratch Nedir?

MIT laboratuvarlarında Lifelong Kindergarten grubunun bir projesi olup 8 ve 16 yaş arası gruplar için oluşturulmuş bir görsel kodlama aracıdır. 8 yaş ve üzeri hedeflenmesine rağmen daha alt yaş grupları için de kullanılabilir. Herhangi bir programlama dilini

bilmeden ve kod yazmadan her yaştan öğrencinin yapabileceği görsel bir programlama aracıdır. Öğrenciler programdaki kod bloklarını bir yerden bir yere sürükleyerek kodlama yapabilmektedir. Scratch sürükle-bırak mantığıyla çalışmaktadır.

Kullanıcı dostu ve kurulumu tamamen ücretsiz olan bir yazılımdır. Bilgisayara kurularak veya online (çevrimiçi) ortamlarda da kullanılabilir. Öğrenciler projelerin Scratch programının sitesine yükleyerek 150 den fazla ülkede diğer öğrencilerle iletişim halinde olabilirler. Sitedeki diğer projeler öğrencilere ilham vererek yaratıcılıkları arttırmaktadır.

Scratch yardımıyla öğrenciler oyun, hareketli ve sesli hikâyeler, animasyonlar, benzetimler ve birçok dijital içerik geliştirilebilir. Scratch yardımı ile farklı derslere içerik üretilerek disiplinler arası etkileşim sağlanabilir.

2.4 Özyeterlik

Bandura tarafından ilk defa ortaya konulan özyeterlik kavramı; bireyin belli bir konuda başarılı olmak gayesiyle lüzumlu gördüğü etkinlikleri bir araya getirerek düzenli bir şekilde uygulamasına olan inancı şeklinde tanımlanmıştır (Bandura, 1997). Eğitim-öğretim sürecindeki özyeterlik ise öğretmenin belli bir duruma ilişkin gerekli olan öğretim etkinliklerini nitelikli bir biçimde yürütme inancı olarak tanımlanmaktadır (Taşkın ve Hacıömeroğlu, 2010).

2.5 Kaynak Araştırması

Özyeterliğin tahminine yönelik yapılan bir çalışmada üniversite öğrencilerinin bir derse yönelik performansları makine öğrenmesi yöntemleriyle tahmin edilmeye çalışılmıştır (Jamjoom, Rhodes, Andrews ve Grant, 2021). Veri seti 244 öğrenciden elde edilen verilerle oluşturulmuştur. Öğrencilerin ders içindeki özyeterliklerini gösteren ön performans düzeyleri dikkate alınarak makine öğrenmesi modellerinden karar ağacı (DT), k-en yakın komşu (kNN), Naïve Bayes (NB) ve destek vektör makinesi (SVM) kullanılmıştır. Çalışma sonucunda, %99.18 doğruluk oranıyla DT ve SVM modellerinin en yüksek doğruluk ile sonuç verdiği görülmüştür.

Ersöz (2017), tarafından 2016 yılında bir üniversitede kayıtlı öğrenciler ile gerçekleştirilen benzer bir çalışmada da her öğrenci için tanımlayıcı bir öğrenci profili oluşturulmak istenmiştir. Öğrencilerin sistemdeki bütün özellikleri kullanılarak belirli profillere göre

sınıflandırılmaya çalışılmıştır. Sınıflandırma analizi sonucu makine öğrenmesi yöntemlerinden karar ağacı tercih edilmiştir. Veriler arasında belirli kurallar oluşturulmuş ve bazı özellikler düğüm noktası olarak belirlenmiştir. Sonuçta geniş bir veri yığını öğrenciyi yansıtan bir profile dönüştürülerek anlamlı hale getirilmiştir.

Üniversite öğrencileri ile gerçekleştirilen başka bir çalışmada Wirojcharoenwong, Luangnaruedom, Rattanasiriwongwut ve Tiantongve (2014) öğrencilerin bilgisayar özyeterliklerinin sınıflandırılmasını hedeflemişlerdir. Bu amaçla CfsSubsetEval altküme değerlendiricisi ile ve en iyi İlk öznitelik seçim algoritmasını kullanarak uygun öznitelikler seçilerek Karar Ağacı (C 4.5) yöntemi ile sınıflandırılmıştır. Veriler anket yardımıyla toplanmış analiz edilmiştir. Öznitelik seçimi neticesinde 12 özniteliğe ulaşılmış ve en iyi doğruluk değeri % 82.5766 olarak bulunmuştur.

Çeşmeli, Bozkurt, Kalkan ve Pençe (2015), tarafından yapılan bir çalışmada da Yönetim Bilişim Sistemleri bölümü öğrencilerinin yönetim ve bilişim derslerindeki başarı durumlarının veri analizi yöntemleriyle incelenmesi amaçlanmıştır. Yönetim Bilişim Sistemleri Bölümü 3. ve 4. Sınıf öğrencilerinin cinsiyet, yaş, öğrenim türü, sınıf, disiplin ceza durumu, hazırlık sınıfı okuma durumu, hazırlık notu, yönetim (11 ders) ve bilişim dersleri (14 ders) ve öğrencinin üniversite öncesi ikametgâh yeri bilgileri olmak üzere toplam 11 öznitelik belirlenmiştir. Yapay Sinir Ağları (YSA) ve Uyarlamalı Sinir-Bulanık Sınıflayıcı (USBS) yöntemlerin kullanılmıştır. YSA'nın eğitim setinde %90.4 başarı elde edilirken, test setinde ise %83.5 başarı elde edilmiştir. USBS ile yapılan sınıflandırmada ise eğitim verisi için %86.6, test verisi için %86.3 başarıya ulaşılmıştır. USBS ile yapılan öznitelik seçiminin test verisinin başarı oranını azaltmadığı belirlenmiştir. Yapılan çalışmalar öğrencinin başarısında hazırlık sınıfı okuma ile dersin kategorisinin en önemli faktörler olduğunu ortaya koymuştur.

Tankuş (2019), Şanlıurfa ilinde 2018 yılında YÖK tarafından yapılan TYT sınavına giren 12080 öğrencinin test puanları analiz edilerek ilçe bazında kategorilere ayrılmıştır. Buna ek olarak TYT puanlarının aritmetik ortalaması hesaplanmış ve okul türüne göre dört kategoriye ayrılmıştır. İstatiksel metot, Çok Katmanlı Algılayıcı 5, 10, 15 gizli katmanlı, Doğrusal Regresyon, Destek Vektör Regresyonu ve epsilon parametrelili, Rastgele Orman Regresyonu işe koşularak korelasyon katsayısı değerlerinin TYT puanına etkisi belirlenmeye çalışılmıştır. Sonuç olarak Türkçe, Sosyal Bilimler, Fen Bilimler, Matematik netleri ile diploma notları kullanılarak TYT puanı tahmin edilmeye çalışılmıştır. Diploma

notunun, ilçenin ve okul türünün TYT puanı üzerindeki etki düzeyleri de ortaya konulmuştur.

Merkezi bir sınavın incelendiği bir çalışmada da 2012 PISA sınavına ilişkin verilerle matematik okuryazarlığı bakımından başarılı ve başarısız öğrencileri derse ilişkin ilgi, tutum, motivasyon, algı, öz yeterlik, kaygı ve çalışma disiplini değişkenlerine göre sınıflandırmak ve bu değişkenlerin sınıflandırmada etkisini ortaya koymak amaçlanmıştır (Ferhan, 2016). Çalışmanın örneklemini Türkiye’den PISA sınavına katılan öğrenciler oluşturmuştur. Verilerin analizinde SPSS, WEKA ve karar ağacı uygulamalarından CHAID ile veri madenciliğinden yararlanılmıştır. Çalışma sonucunda başarılı ve başarısız öğrencileri sınıflandırmada en önemli duyuşsal özelliklerin özyeterlik algısı, derse ilişkin tutum ve çalışma disiplini olduğu ortaya konulmuştur. Çalışma sonuçları ayrıca veri madenciliği yöntemlerinden J.48 karar ağacı yardımıyla elde edilen doğru sınıflandırma yüzdesinin CHAID analizi yöntemiyle elde edilen değere çok yakın olduğu ve alternatif bir yöntem olarak onun da rahatça kullanılabileceği belirlenmiştir. Elde edilen bulgular Türk öğrenciler için matematik okuryazarlığında özyeterlik algısı, derse ilişkin tutum ve kaygı durumları ile çalışma disiplini konuları üzerinde özellikle durulması gerektiğini ortaya koymuştur. Bu alanlarda alınacak tedbirler ve geliştirmeler öğrencilerin başarı durumunu arttırarak ülkemizin PISA sınavlarında daha üst sıralara yükselebilmesini sağlayacaktır.

Başka bir çalışmada da PISA 2015 sınavı ile öğrencilerin matematik okuryazarlık düzeylerini tahmin etmek ve matematik okuryazarlığı üzerinde etkili olan değişkenleri tespit etmek istenmiştir (Aksu, 2019). Bu amaçla farklı yeterlik düzeylerinde olan altı farklı ülkenin (Singapur, Japonya, Norveç, Amerika, Türkiye ve Dominik Cumhuriyeti) öğrencilerinin matematik okuryazarlık düzeyleri üzerinde etkili olan değişkenler ayrı ayrı incelenmiştir. 34.565 öğrencinin verisi ilk aşamada WEKA programında analiz edilerek M5P algoritması kullanılmıştır. İkinci aşamada ise M5P tahminleme yöntemiyle elde edilen karar ağaçlarının ne düzeyde geçerli sonuçlar verdiğini ortaya koyabilmek için yapay sinir ağları yöntemi kullanılmıştır. Böylece girdi değişkenleri sayesinde çıktı değişkeni tahmin edilmeye çalışılmıştır. MATLAB programında yapılan analizler sonucu öğrencilerin gerçek matematik okuryazarlık puanları ile girdi değişkenleri yardımıyla tahmin edilen okuryazarlık puanları arasındaki ilişki incelenmiştir. Her ülke için matematik okuryazarlığı üzerinde etkili olan değişkenlerin farklılık gösterdiği tespit edilmiştir. Bu değişkenleri yordam gücünün de ülkelere göre değiştiği sonucuna varılmıştır.

Nasution (2018) yaptığı bir çalışmada mezunlar için girişimcilik niyetini tahmin etmeye çalışmıştır. Veriler, çevrimiçi bir iş piyasası veri tabanından, mezun izleyicilerinden ve mezunlara yönelik anket sonuçları kullanılarak elde edilmiştir. Girişimciliğin niyetini gösteren bir tahmin modeli elde etmek için C4.5 karar ağacı algoritması kullanılmıştır. Özyeterlik, Başarı İhtiyacı, Danışmanlık Bölümü, Kontrol Odağı ve Tutku gibi pek çok göstergenin girişimcilik niyetinde etkili olduğu düşünülmüştür. Tahmine dayalı model, bu değişkenler içerisinde girişimcilik niyetini en iyi yordayıcının, yüzde 79.7' lik bir değerle katkıda bulunan özyeterlik olduğunu tespit etmiştir.

Çiftçi, Kaleli ve Günal (2018) ise eğitmen performansının makine öğrenmesi yöntemleriyle tespit edilebilmesi üzerine çalışmışlardır. Çalışma kapsamında üniversitesi öğrencilerinin eğitmenleri hakkında doldurdıkları bir Likert tipi bir anket ile veriler toplanmıştır. Oluşturulan veri seti üzerinde çeşitli öznitelik indirgeme algoritmaları ile farklı makine öğrenme yöntemleri kullanılmıştır. Yapılan analizler sonucu en iyi sonuç öznitelik indirgeme algoritmalarından genetik algoritma ile elde edilmiş ve 19 öznitelik kullanılmıştır. Genetik algoritma ile birlikte makine öğrenmesi yöntemiyle %97.70 doğruluk performansına ulaşılmış ve eğitmenlerin performanslarının sıralanması sağlanmıştır.

Dokuzuncu sınıf öğrencilerinden toplanan eğitsel veriler üzerinde makine öğrenmesi yöntemleri kullanılarak bir içgörü elde edilmeye çalışılmıştır (Yıldız ve Börekçi, 2020). Kullanılan veri setinde öğrenciler ve aileleri hakkında demografik bilgiler, çalışma rutinleri, öğrenme etkinliklerine katılma davranışlarını ve bilim hakkındaki epistemolojik inançları yer almıştır. Toplanan verilerden iki sınıflı (sınav sonucuna göre başarılı veya başarısız) bir tahmin yapılmaya çalışılmıştır. Makine öğrenmesi yöntemlerinden en iyi tahmin doğruluğuna % 98.6 ile Sinir Ağı algoritması kullanılarak ulaşılmıştır. Akademik başarı tahmin doğruluğunu, ailenin demografik değişkenlerinin, öğrencinin bilimsel epistemolojik inançlarının, çalışma rutinlerinin ve bazı derslere yönelik tutumlarının büyük oranda etkilediği görülmüştür.

Ünal (2019), 5'li Likert tipi ölçeklerle üretilen yapay veri setlerinin derin öğrenme tekniklerinin performanslarının belirlenmesi amaçlamıştır. Veri seti olarak müşteri memnuniyet tahminine ilişkin verilerin çoğaltılması amacıyla yamuk bulanık sayılar kullanılarak bulanık forma dönüştürülerek kullanılmıştır. Verilerin analizinde ilk olarak lojistik regresyon yöntemi ile pilot çalışma yapılmıştır. Daha sonra bulanık mantık

kapsamında derin öğrenme teknikleri değerlendirilerek karşılaştırılmıştır. Derin öğrenme tekniklerinden Relu ve SoftPlus algoritmaları kullanılmıştır. Derin öğrenme ile yapılan analizlerde tepe, maksimum ve minimum değerler için ayrı ayrı sonuçlarla durulaştırmaya gidilmiş hem de bulanık sayıların tek sonuç dizisi üretmesi ile derin öğrenme modelinin performansları araştırılmıştır (Ünal, 2019).

Türk ve Tali (2020) tarafından yürütülen bir çalışmada somatizasyon ve duygusal öz farkındalık değişkenleri kullanılarak makine öğrenmesi yöntemleri ile bireylerin şiddet mağduriyetlerinin tahmin edilmesi amaçlanmıştır. Veriler, 552 katılımcıya Kişisel Bilgi Formu, Somatizasyon Ölçeği ve A Duygusal Öz Farkındalık Ölçeği-10 (A-DÖFÖ-10) uygulanarak toplanmıştır. Makine öğrenmesinde sıkça kullanılan sınıflandırma algoritmaları tek tek denenmiştir. Denenen yöntemlerden en doğru sınıflandırma performansını 0.74 doğruluk ve 0.82 f1-skoru değeri ile Lojistik Regresyondan elde edilmiştir. Bulgular, somatizasyon ve duygusal öz farkındalık kavramları üzerinden makine öğrenmesi yöntemleri ile bireylerin şiddet mağduriyetleri tahmin edilmiştir (Türk ve Tali, 2020).

3. MATERYAL VE YÖNTEM

3.1 Veri Setinin Oluşturulması

Çalışmanın veri seti 192 Bilişim Teknolojileri öğretmeninden TSECT anketi aracılığıyla toplanmıştır. Bean, Weese, Feldhausen ve Bell (2015) tarafından geliştirilen TSECT anketi gerekli izinler alındıktan sonra kullanılmıştır. Bu anket blok tabanlı kodlama araçlarına ilişkin özyeterlik düzeyinin belirlenmesini hedefleyen likert tipindeki maddelerden oluşmaktadır. Anket verileri 39 öznitelikten oluşmuştur. Likert tipi ölçekler sosyal bilimlerde en çok kullanılan yöntemlerdendir (Turgut ve Baykul, 1992). Her maddeye yanıtlayıcı tarafından verilen cevaplar yanıtlayıcının maddeye ilişkin tutumunun derecesini verir (Tezbaşaran,1997).

Veriler öncelikle .xlsx formatından .arff formatına dönüştürülmüştür. Veri setinde yer alan öznitelikler ağırlıklandırılarak toplam sonuç puanı elde edilmiştir. Toplam sonuç puanı 2800'den küçükse 'yeterli değil' ve 2800'den büyükse 'yeterli' olmak üzere yeterlik_düzeyi özniteliği belirlenmiştir. Anketteki maddelere verilen cevaplardan oluşan puanlama sonucunda 2800 değeri elde edilmiştir. Bu değer ölçek maddeleri ağırlıklandırılarak oluşturulmuştur. Özniteliklere ait bilgiler Tablo 1'de görülmektedir.

Veri seti Covid-19 pandemisi nedeniyle Google Forms aracılığı ile katılımcıların cevapları ile oluşturulabilmiştir. Bu yüzden veri seti daha sınırlı sayıda kalmıştır. Verinin sınıflandırılması aşamasında küçük dengesizlikler olduğu için SMOTE yöntemi kullanılarak veri 192'den 262'ye çoğaltılmıştır. Sınıflar arasındaki dengesizlik giderilmiştir.

Tablo 1. Özniteliklere ait bilgiler

Etiket	Veri Tipi	Tanımı
A1	Numerik	Bilgisayar programları yazımında kendime güvenirim.
A2	Numerik	Programlama kavramlarını etkili bir biçimde öğretmeyi bilirim.
A3	Numerik	Öğrencilerimin programlamaya karşı olumlu tutumunu destekleyebilirim
A4	Numerik	Farklı konuları işlerken öğrencilerimin programlamayı bir araç olarak kullanmalarını sağlamak için rehberlik edebilirim.
A5	Numerik	Sınıfımda programlamayı eğitim aracı olarak kullanma konusunda kendime güvenirim.
A6	Numerik	Programlamayı bir öğretim aracı olarak kapsayan ders planlarını uyarlayabilirim.
A7	Numerik	Programlamayı bir öğretim aracı olarak kapsayan orijinal ders planları oluşturabilirim.
A8	Numerik	Programlama kavramlarının Yeni Nesil Bilim Standartları ile nasıl ilişkili olduğunu açıklayabilirim.
A9	Numerik	komut dizisini adım-adım çalıştırılmak
A10	Numerik	komutları yinlemek için döngü kullanmak
A11	Numerik	klavye üzerinde bir tuşa basarak bir olaya müdahale etmek
A12	Numerik	bir anda birden fazla iş yapmak
A13	Numerik	özel koşulları sağlayacak komutları çalıştırmak
A14	Numerik	toplama ve çıkartma benzeri işlemleri yapmak
A15	Numerik	değerleri yüklemek, düzeltmek ve bulmak
A16	Numerik	kullanıcıya bir soru sormak
A17	Numerik	programlara yönelik olumlu tutum geliştirmeye destek
A18	Numerik	farklı konularda araştırırken programlamayı bir araç olarak kullanmalarına rehberlik
A19	Numerik	bir eğitim aracı olarak programlama kullanımı
A20	Numerik	ders programlarını öğrencilerimin öğrenme seviyesine göre ayarlama
A21	Numerik	bir eğitim aracı olarak programlama kullanımı için yeniden ders programı oluşturma
A22	Numerik	Programlamanın Yeni Nesil Bilim Standartları ilişkisi hakkında farkındalığımı artırma
A23	Numerik	şekil çizimi
A24	Numerik	animasyon (canlandırma filmi yapımı)
A25	Numerik	bir hikâye anlatmak
A26	Numerik	müzik yapmak
A27	Numerik	modelleme yapmak
A28	Numerik	adım adım komutları çalıştırmak
A29	Numerik	komutları yinlemek için döngü kullanmak
A30	Numerik	klavye üzerinde bir tuşa basarak bir olaya müdahale etmek
A31	Numerik	bir anda birden fazla iş yapmak
A32	Numerik	özel koşulları sağlayacak komutları çalıştırmak
A33	Numerik	toplama ve çıkartma benzeri işlemleri yapmak
A34	Numerik	değerleri yüklemek, düzeltmek ve bulmak
A35	Numerik	kullanıcıya bir soru sormak
A36	Numerik	geliştirmeleri adım adım yaparak, yeni fikirler geldikçe eklerim
A37	Numerik	istediğim veya bulduğum sorunları düzeltmeyi yaptığıma emin olmak için sık sık programımı çalıştırırım
A38	Numerik	programlarımı başkaları ile paylaşarak onların fikirlerini görmek isterim
A39	Numerik	değişik işlevleri yapmak için programımı birden fazla bölüme ayırırım
yeterlik düzeyi	Nominal	

3.2 Synthetic Minority Oversampling Technique (SMOTE)

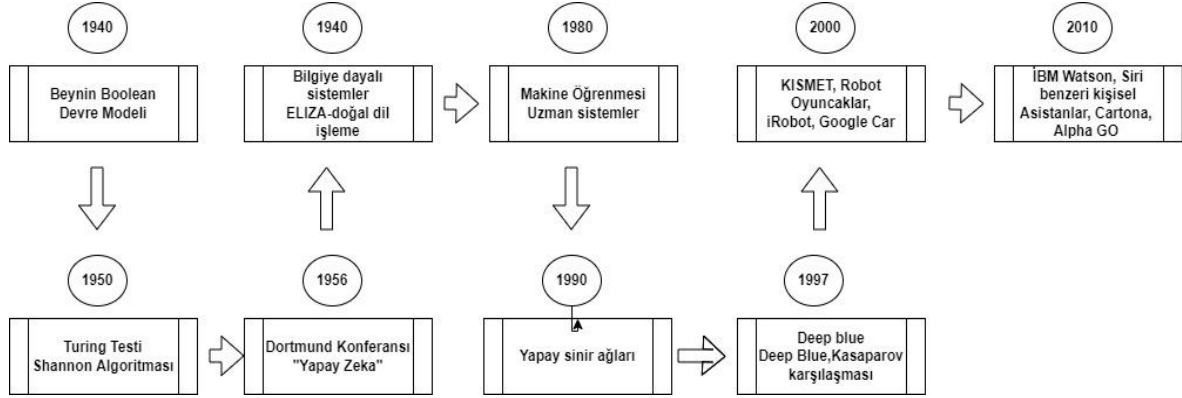
Makine öğrenmesinde kullanılan algoritmalar veri setlerinin dengeli dağıldığını varsayarak işlemleri yürütür (He ve Garcia, 2009). Sınıflandırma yapılırken sınıflar arasında dengesiz veri kümesi ile karşılaşıldığında, algoritmalar dağılımın özellikleri sakıncalı doğruluk ve duyarlılık değerleri elde edebilirler (Kartal ve Özen, 2017). Dengesiz veri kümelerinde, örneklem sayısı bakımından baskın gelen sınıflarla azınlıkta kalan sınıflar arasında dengesizlikler açığa çıkabilmektedir. Bu veri kümelerinin başarılı şekilde

sınıflandırabilmesi için öncelikle dengesiz veri kümelerini daha dengeli bir hale getiren veri örnekleme yöntemlerinin kullanılması önemlidir (Yavaş, Güran ve Uysal, 2020). Synthetic Minority Oversampling Technique (SMOTE) bu yöntemlerin en çok kullanılanlarından birisidir. SMOTE kullanılırken azınlık sınıf oluşturan veriler alınır ve o örneğe en yakın olan k komşu değerine veya tüm veriye bakılarak sentetik olarak çoğaltılır (Chawla, Bowyer, Hall ve Kegelmeyer, 2002). Diğer yöntemlerde kullanılan azınlık sınıf verilerinin kopyalanarak çoğaltılmasının yapılmaması SMOTE yönteminin kullanıldığı sınıflandırmaların başarımlarını daha da arttırmaktadır (Kartal ve Özen, 2017).

3.3 Yapay Zekâ

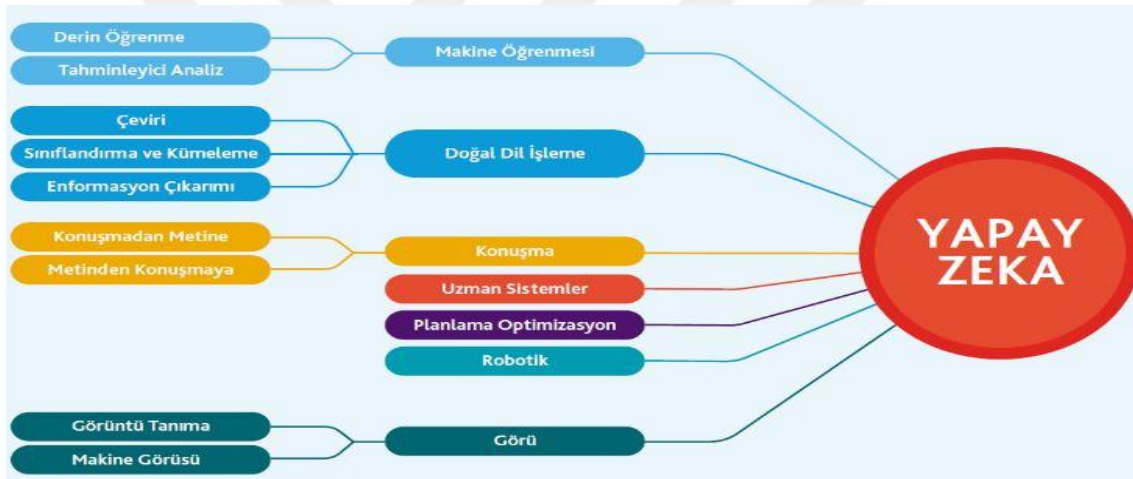
1956 yılında Dortmund Konferansı'nda McCarthy, Minsky, Rochester ve Shannon tarafından hazırlanan bir öneri mektubunda yapay zekâ kavramından bahsedilmiştir. Mektupta dört kişinin adı geçmesine rağmen yapay zekâ kavramının ilk ortaya atan kişi McCarthy olarak kabul edilir (Alaydın, 2013). McCarthy (2004, s.7) ilk olarak zekâyı şu şekilde tanımlamıştır: “Dünyada hedeflere ulaşma yeteneğinin hesaplayıcı bir parçasıdır. Değişen türde ve derecede zekâ insanlarda, birçok hayvanda ve bazı makinelerde görülür”. Yapay zekâ genel anlamda; bir bilgisayarın ya da bir makinenin, insanlara ait niteliklerden olan akıl yürütme, anlam çıkartma, genelleme ve geçmiş deneyimlerden öğrenme gibi üst seviye zihinsel süreçlere ilişkin görevleri yerine getirme yeteneği olarak tanımlanabilir (Karaduman, 2019). Ayrıca başka bir tanım da yapay zekâ, “insan benzeri zeki makineler özellikle de zeki bilgisayar programları yapma bilimi ve mühendisliği” olarak tanımlamıştır. Farklı bir açıdan bakacak olursak yapay zeka, bilgisayarların insanlara ait olan üst düzey bilişsel özellikleri kullanması olarak da tanımlanabilir.

Yapay zekâ günümüzde büyük veri sayesinde kendisine birçok alanda yer bulmuştur. Teknoloji, bankacılık ve eğlence başlıca yapay zekânın en çok kullanıldığı alanlardır (Elmas, 2007). Bunun yanı sıra akıllı asistan uygulamaları, sürücüsüz otonom araçlar, akıllı ev sistemleri, yüz ve görüntü tanıma ve oyun alanlarında yapay zekâ uygulamaları ile sıklıkla karşılaşılmaktadır. Yapay zekâ sürekli olarak güncellenerek günlük yaşamımızı kolaylaştırmaktadır. Yapay zekâyı tam olarak anlamamız için bu gelişim sürecinin önemli aşamaları Şekil 2’de görülmektedir (Arslan, 2020).



Şekil 2. Yapay zekâ gelişim kronolojisi (Sargın ve Göçen, 2020)

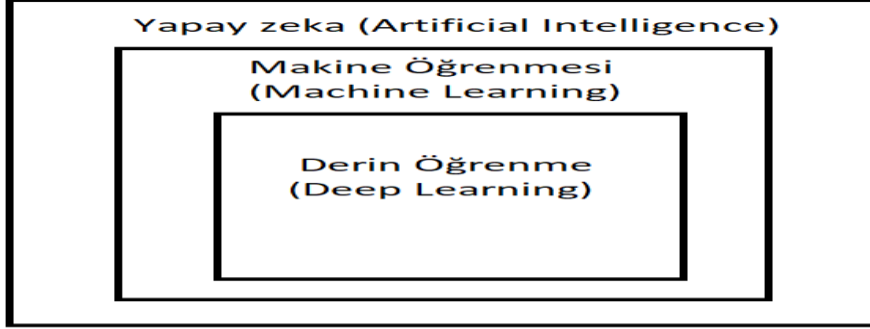
Yapay zekânın birçok çalışma alanı bulunmaktadır. Bu çalışma alanları Şekil 3’de görülmektedir.



Şekil 3. Yapay Zekâ Çalışma Alanları (Sanket, 2018)

Son dönemde en çok öne çıkan alt alanların başında makine öğrenmesi ve derin öğrenme gelmektedir. Makine öğrenmesi bir makinenin örneklerden ve önceki deneyimlerden kendi kendine öğrenebilmesidir. Bunun için makine, yeri geldiğinde çalışma algoritmasını değiştirme veya düzeltme eğilimine girmelidir. Derin öğrenme özelliklerden bilgi çıkararak ve bunları dönüştürerek doğrusal olmayan katmanlarda bunları işleme sokar (Deng ve Yu , 2014) . Her katman kendinden önceki katmanın sonucunu girdi olarak alır. Derin öğrenme verilerin farklı özelliklerini kullanarak veya bunların örneklerini kullanarak öğrenme sağlar (Bengio, 2009).

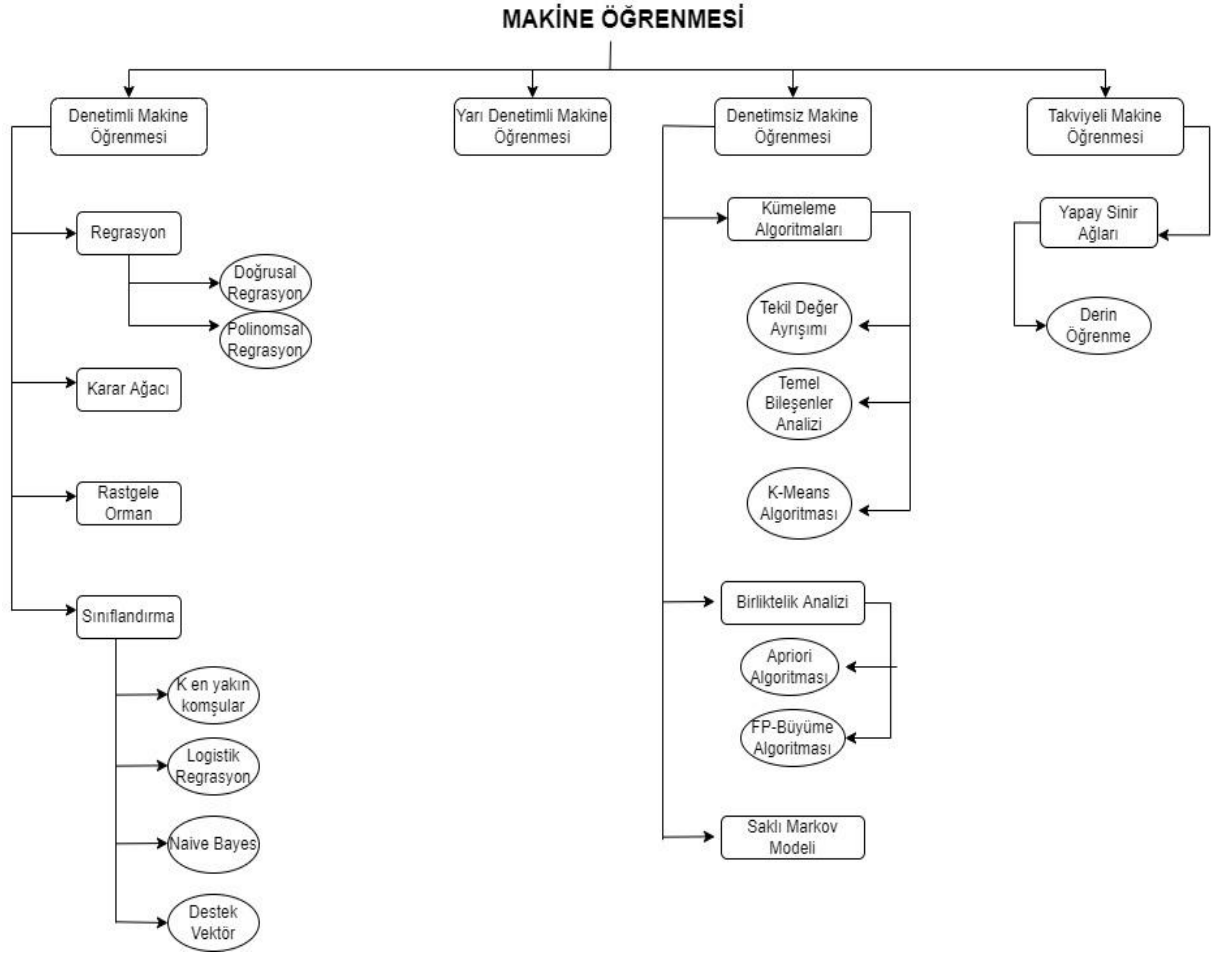
Yapay zekâ, makine öğrenmesi ve derin öğrenme kavram yanılgıları sonucunda akıllı uygulamalarda sıklıkla birbirlerinin yerine kullanılabilmektedir. Fakat bu üç kavram birbirinden farklıdır. Derin öğrenme, makine öğrenmesinin bir alt dalıdır. Makine öğrenmesi ise yapay zekânın alt başlıklarından biridir (Aylak, Oral ve Yazıcı, 2021). Sonuç olarak tüm makine öğrenmesi uygulamaları yapay zekâ olarak kabul edilebilir (Şekil 4).



Şekil 4. Yapay Zekâ, Makine Öğrenmesi ve Derin Öğrenme ilişkisi (Güngör ve Yardımcı, 2020)

3.4 Makine Öğrenmesi

Bir problemi aynı probleme ilişkin verilere dayanarak modelleyen bilgisayar algoritmalarına makine öğrenmesi denir (Nizam ve Akın, 2014). Makine öğrenmesi yöntemleri kullanılarak büyük verilerden sınıflandırma, tahmin veya kümeleme işlemler yapılabilmektedir. Bundan dolayı çok sayıda makine öğrenmesi yöntemi geliştirilmiş olup bunlardan bazıları; k-en yakın komşu algoritması, basit (naive) Bayes sınıflandırıcı, karar ağaçları, lojistik regresyon analizi, k-ortalamlar algoritması, destek vektör makinaları ve yapay sinir ağlarıdır. Şekil 5’de görüldüğü gibi makine öğrenme yöntemleri denetimli, yarı denetimli, denetimsiz ve pekiştirmeli (takviyeli) olmak üzere dört kategoride incelenmektedir (Kaynar, Arslan, Görmez ve Işık, 2018).

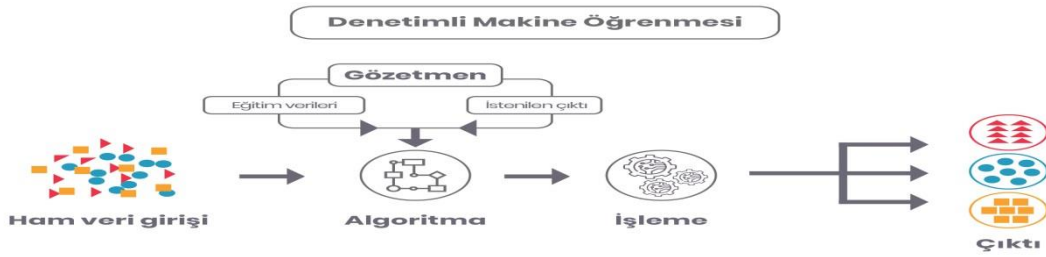


Şekil 5. Makine Öğrenmesi Yöntemleri

3.4.1 Denetimli Makine Öğrenmesi

Örnekler üzerinden gidilerek öğrenmenin sağlanması amaçlanmaktadır. Veri seti ile giriş ve çıkış bilgileri verilir ve makine buna göre çıkışa nasıl ulaşılacağına dair bir yöntem bulur. Denetimli makine öğrenmesi, herhangi bir girişe ait çıkışı bilmesine rağmen veride tanımlar, gözlemler ve tahminlerde bulunur. Tahminler yüksek seviyede doğruluk performansı gösterene kadar tekrar edilerek en iyi sonuçlara erişilmeye çalışılır (Atalay ve Çelik, 2017) .

En nihayetinde denetimli öğrenmede oluşturulan modelle, Şekil 6’da görüldüğü gibi bir grup girdi değerine karşılık bu girdiye ait hedef değerler verilerek aralarındaki ilişkiyi öğrenmesi ve hedef değerlere en yakın çıktıların üretilmesi amaçlanmaktadır (Aylak vd., 2021).



Şekil 6. Denetimli Makine Öğrenmesi (Turkhost, 2021)

3.4.2 Yarı Denetimli Makine Öğrenmesi

Denetimli makine öğrenmesine çok benzemekle beraber farkı hem etiketli hem etiketsiz verileri kullanmasıdır. Etiketli veriler anlamlı verilerdir. Makine öğrenmesi yöntemleri kullanılarak Şekil 7’de verildiği gibi yarı denetimli yöntem ile etiketsiz verileri etiketleme yapabiliriz (Turkhost, 2021).

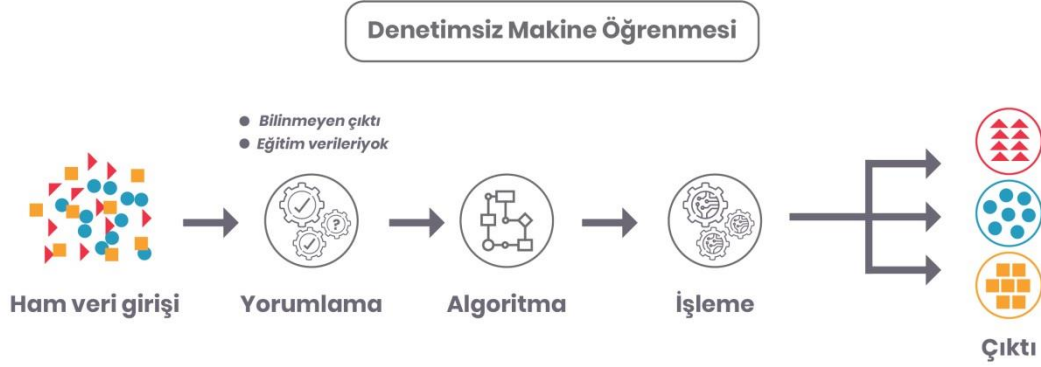


Şekil 7. Yarı Denetimli Makine Öğrenmesi (Turkhost, 2021)

3.4.3 Denetimsiz Makine Öğrenmesi

Denetimsiz makine öğrenmesi büyük veriler üzerinde yorumlama yapabilir. Bu veriler ilk olarak belli bir düzene sokulur. Veriler değerlendirildikçe diğer veriler ile ilgili karar verme yetenekleri artar (Turkhost, 2021).

Denetimsiz makine öğrenmesiyle Şekil 8’de görüldüğü gibi sadece girdi değerleri arasındaki ilişkiler ortaya çıkarılmaya çalışılır ve bu ilişkilerden yola çıkılarak birbirine yakın değerler gruplandırılarak kümeleme işlemi gerçekleştirilmiş olur (Marsland, 2011). Veri sayısı arttıkça makinenin karar verebilme performansı da artış gösterir ve daha başarılı bir biçimde verinin hangi kümeye ait olacağını daha doğru biçimde belirler.

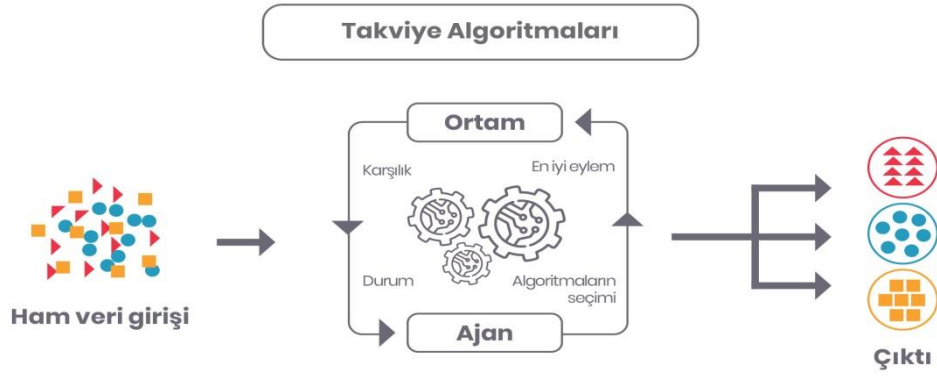


Şekil 8. Yarı Denetimsiz Makine Öğrenmesi (Turkhost, 2021)

3.4.4 Takviyeli (Pekiştirmeli) Makine Öğrenmesi

Bu makine öğrenmesi belirli kurallar çerçevesinde en iyi sonuca ulaşmayı amaçlar. Farklı seçenekler ve olasılıkları değerlendirerek en uygun sonuca ulaşır. Makine, denemeleri sonucunda hatalarından öğrenme gerçekleştirir ve en iyi sonuca ulaşır (Turkhost, 2021).

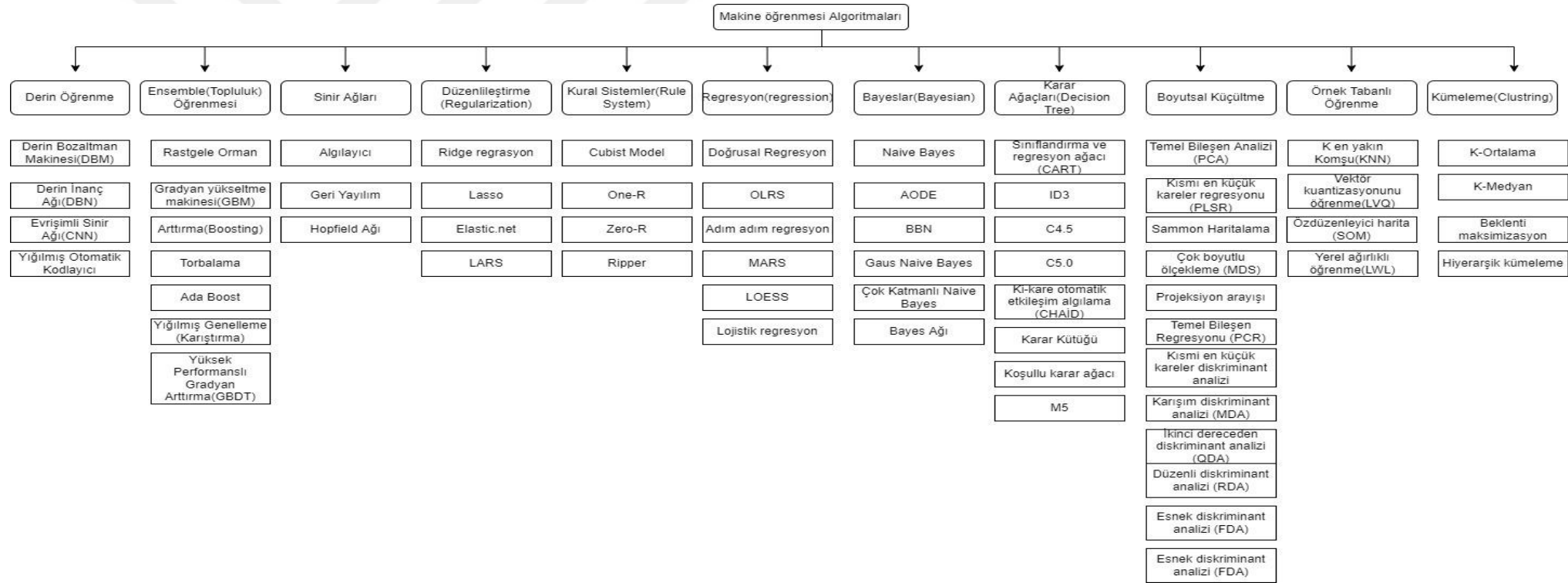
Pekiştirmeli(takviyeli) öğrenme yönteminde, birden fazla yöntemin birlikte kullanıldığı ve en etkili işlemin belirlenip bu işlemle makine öğrenme işlemleri yapılan yöntemdir (Yavuz, 2020). Bu makine öğrenmesi yönteminde, Şekil 9’daki gibi hedef çıktıyı elde etmek için bir danışman kullanmak yerine çıkışın verilen girişe karşılık iyi ya da kötü olarak değerlendirmesini yapan bir ölçüt kullanılmaktadır.



Şekil 9. Takviyeli (Pekiştirmeli) Makine Öğrenmesi (Turkhost, 2021)

3.4.5 Makine Öğrenmesi Algoritmaları

Makine öğrenmesi algoritmaları temel olarak derin öğrenme, topluluk öğrenmesi, sinir ağları, düzenlileştirme, kural sistemleri, regresyon, bayeslar, karar ağaçları, boyutsal küçültme, örnek tabanlı öğrenme ve kümeleme olmak üzere 11 alt başlıkta incelenmektedir. Bu başlıkları oluşturan algoritma ve yöntemler Şekil 10'da verilmiştir.



Şekil 10. Makine Öğrenmesi Algoritmaları (Cankıran,2019)

3.4.6 Kullanılan Klasik Makine Öğrenmesi Algoritmaları

Bu başlık altında yapılan çalışmada Bilişim Teknolojileri öğretmenlerinin öz yeterlik düzeylerinin sınıflandırılmasında kullanılan klasik makine öğrenmesi algoritmalarına yer verilmiştir.

3.4.6.1 Sıralı Minimum Optimizasyonu (SMO)

Sıralı Minimum Optimizasyon (SMO), bir matris depolamasına gerek duymadan ve sayısal ikinci dereceden programlama optimizasyon adımlarını kullanmadan SVM problemini basit ve hızlı bir şekilde çözebilen bir algoritmadır (Platt,1998). SMO Weka programında SVM algoritmasının karşılığıdır SMO'da hiçbir matris algoritması kullanılmadığından, sayısal hassasiyet sorunlarına en az rastlanır (Platt, 1998).

SMO ilk aşamada tüm çarpanlardaki kısıtlamaları hesaplayarak sınırlandırılmış maksimum değeri belirler (Kuan, Wang, Wang, Lin ve Gu, 2011). SMO'nun ana bileşeni olan iki Lagrange çarpanını çözmek için analitik bir yöntem sunar ve hangi çarpanların optimize edileceğini seçmek için bir buluşsal yöntem kullanır (Platt, 1998). İki çarpanın optimize edilmesinde, kısıtlamalar bir kutu yatılı sınırlandırılmış çapraz çizgi parçası olan iki boyutlu kareyi tanımlamak olarak görülebilir (Kuan vd, 2011). SMO algoritmasının gösterilmesi Şekil 11'deki gibidir.

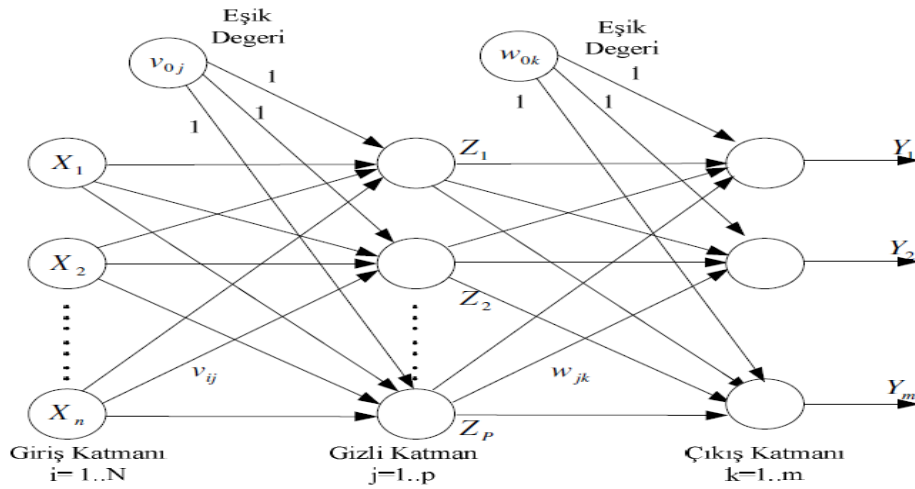
Initial
 $w=0, b=0$ and all $a_i=0, E=0$
loop
Step 1. Preprocess steps:
 $k_{ij}=x_i^T x_j, k_{ii}= x_i^T x_i, k_{jj}= x_j^T x_j$
 $\eta=2 k_{ij}-k_{ii}-k_{jj}$
calculate prediction error E_i, E_j and a_i^{old}, a_j^{old}, r .
Determine feasible range $[L,H]$ used for clipping, $a_j^{new,unc}$
Step 2. Lagrange Multiplier Updating steps:
2.1. $\alpha_j^{new,unc} = \alpha_j^{old} + \frac{y^j(E_j-E_i)}{\eta}$, and Clipping to L or H
2.2. $\alpha_i^{new} = \alpha_i^{old} + y_i y_j (\alpha_j^{old} - \alpha_j^{new})$
Step 3.SVM parameter updating steps:
3.1. update $w, b, \Delta b$.
3.2. update prediction error E .
end loop (if all a satisfied KKT condition)

Şekil 11. SMO Algoritması

3.4.6.2 Multilayer Perceptron (Çok Katmanlı Algılayıcı)

MLP'ler en popüler sinir ağı mimarileri arasındadır (Süt ve Çelik, 2011). Bunun sebebi MLP'nin öğrenme ve genelleme yeteneği, daha küçük eğitim seti gereksinimleri, hızlı çalışma ve uygulama kolaylığı gibi özelliklere sahip olmasıdır (Güler ve Ubeyli, 2005).

MLP ağlarında nöronlar girdi, çıktı ve gizli katman olmak üzere üç farklı katmanda yer alır. İlk katman olan girdi katmanı, çözüme ulaştırılması beklenen probleme ait bilgilerin yapay sinir ağına aktarılmasını sağlar (Güler ve Ubeyli, 2005). Çıktı katmanı ağ içerisinde işlenen bilginin ağ dışına iletiildiği katmandır (Kaynar ve Taştan, 2009). Gizli katman, girdi ve çıktı katmanlarının arasında bulunur ve gizli katman sayısı birden fazla bulunabilir (Süt ve Çelik, 2011). Standart bir MLP ağının katman yapısı Şekil 12'de yer almaktadır.



Şekil 12. Çok katmanlı ileri beslemeli yapay sinir ağı (Kaynar ve Taştan, 2009).

MLP' de, gizli katmandaki her bir nöron j , giriş sinyallerini x_i ilgili bağlantı ağırlıklarının w_{ji} 'lerinin gücü ile çarpıp toplar ve çıkış y_j 'sini toplamın bir fonksiyonu olarak Eşitlik 1'deki gibi hesaplanmış olur (Güler ve Ubeyli, 2005). Eşitlik 1'de yer alan aktivasyon fonksiyonu (f), bir nörona çarpan tüm sinyallerin ağırlıklı toplamını dönüştürmek için gereklidir. Aktivasyon fonksiyonu (f) basit bir eşik fonksiyonu veya sigmoidal, hiperbolik tanjant veya radyal temel fonksiyonu olabilir.

$$y_j = f(\sum(w_{ji} x_i)) \quad (1)$$

Eşitlik 2’de görülen çıkış nöronlarının istenen ve gerçek değerleri arasındaki kare farklılıkların toplamı E olarak tanımlanır (Güler ve Ubeyli, 2005). Üstte yer alan denklemde y_{dj} çıkış nöronunun istenen değeridir j ve y_j bu nöronun gerçek çıktısıdır. Her ağırlık E 'yi mümkün olduğunca çabuk azaltacak şekilde ayarlanmıştır. w_{ji} 'nin nasıl ayarlandığı, benimsenen eğitim algoritmasına bağlıdır (Basheer ve Hajmeer, 2000).

$$E = \frac{1}{2} \sum (y_{dj} - y_j)^2 \quad (2)$$

3.4.6.3 K-Star Algoritması

K-star, olabilecek bütün dönüşümler arasından rastgele seçim yaparak bir örneği diğerine dönüştürme olasılığını entropik ölçü kullanarak sınıflandıran bir algoritmadır (Madhusudana, Kumar ve Narendranath, 2016). Sembolik ve eksik değerler gerçek değerlerle tutarlı olamayabilir. Sınıflandırıcılarda oluşan bu karmaşıklığın giderilebilmesi için entropik ölçüm kullanılması büyük bir avantajdır (Painuli, Elangovan ve Sugumaran, 2014). Entropi özellikten özelliğe dönüşüm olarak tanımlanan sonlu sayıda dönüşümler kümesi oluşturulur. Ardından bir özelliğin (a) diğer bir özelliğe (b) dönüşümü, (a)’dan başlayıp (b)’de sonlanan sınırlı dönüşümler dizisi şeklinde belirlenir (Cleary ve Trig, 1995). Olasılık fonksiyonu P^* , t dönüşümleri ile a özelliğinden b özelliğine olan tüm olasılıklar,

$$P^*\left(\frac{b}{a}\right) = \sum_{\bar{t} \in P: \bar{t}(a)=b} P(\bar{t}) \quad (3)$$

belirlenir. Buna göre K-star (K^*) fonksiyonu

$$K^*\left(\frac{b}{a}\right) = -\log_2 P^*\left(\frac{b}{a}\right) \quad (4)$$

şeklinde ifade edilir.

K-star fonksiyonu sayısal veya sembolik öznelik değerine sahip veri setlerinin sınıflandırılmasında kullanılabilir. Sayısal veri değerleri için kullanımında x_0 , sembolik veri değerleri için kullanımında ise s parametreleri eğitim öncesinde belirlenmektedir (Çölkesen ve Kavzoğlu, 2010). Sayısal ve sembolik sınıflandırmalarda olasılık dağılımına dahil edilen örnek sayısı, en yakın komşuluk dağılımı olan 1 değeri ile

tüm örneklerin eşit ağırlığa sahip olduğu N değeri aralığında değişir (Cleary and Trig, 1995). Herhangi bir P^* fonksiyonu için etkili örnek sayısı

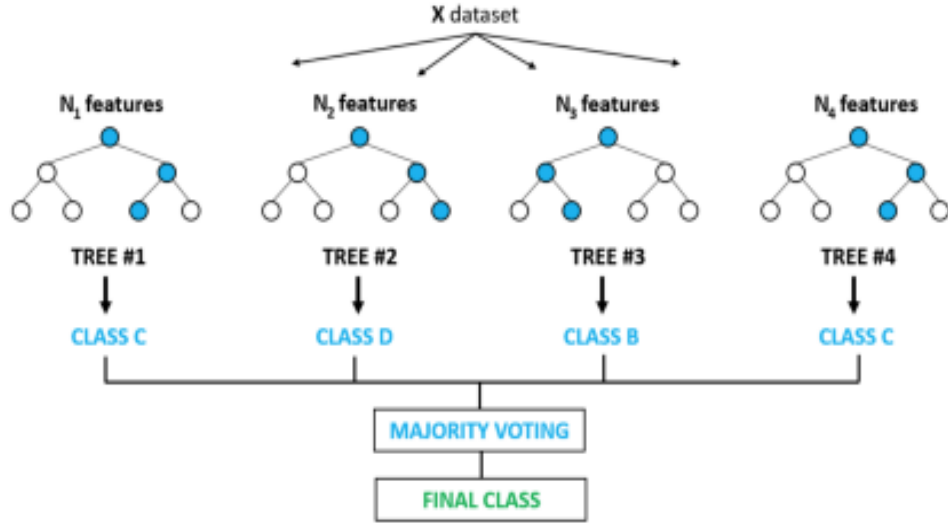
$$n_0 \leq \frac{(\sum_b P^*(b/a))^2}{\sum_b P^*(b/a)} \leq N \quad (5)$$

şeklinde gösterilir. Burada N eğitim örneklerinin tümünün sayısını, n_0 ise eğitim örneklerinden a örneğine en yakın mesafedeki örneklerin sayısını gösterir (Çölkesen ve Kavzoğlu, 2010). K-star algoritması x_0 ve s parametreleri için a değerini n_0 ve N arasındaki bir değer seçilmesiyle belirler. Böylece seçilen n_0 en yakın komşuluk algoritmasını ve seçilen N eşit ağırlıklı örneklerin sayısını verir. Belirlenen bu sayıların uygunluğu harmanlama parametresi (b) kullanılarak belirlenir. Harmanlama parametresi $b = \%0$ (n_0 için) ve $b = \%100$ (N için) arasında değişmektedir (Çölkesen, 2010).

3.4.6.4 Random Forest Algoritması

Random Forest (RF), Leo Breiman ve Adele Cutler tarafından geliştirilmiştir. RF, tek bir sınıflandırıcı ağaç kullanmak yerine birçok sınıflandırıcı ağacın beraber kullanılması ve ardından tahminlerinden alınan oylar ile yeni veriyi sınıflandırmayı sağlayan bir öğrenme algoritmasıdır (Nizam ve Akın, 2014). RF, tüm değişkenler arasında en iyi bölünmeyi kullanarak her düğümü bölmek yerine, her düğümü, o düğümde rasgele seçilen bir tahmin edici alt kümesi arasından en iyisini kullanarak böler (Akar ve Güngör, 2012).

Rastgele orman algoritması için orijinal veri setinden bootstrap yöntemi ile n tane örneklem belirlenir (Breiman, 2001). n dahil edilecek karar ağacı algoritması sayısını göstermektedir (Breiman, 2001). Daha sonra her örneklemin üçte ikisi eğitim verisi olarak bölünür. Daha sonra her örnekleme için farklı karar ağaçları yapılır. Ağaçlar hazırlanırken her düğümde tahmin değişkenleri içerisinde en iyi değişken yerine, rastgele m tane tahmin değişkeni seçilmesine dikkat edilmelidir (Korkem, 2013). m tane değişkenden, bilgi kazancı oranını optimum seviyede tutan değişken, dallandırma için belirlenir (Akar ve Güngör, 2012). Belirlenecek olan m değerinin regresyon ağaçlarında $\sqrt{p}$, sınıflama ağaçlarında ise $\sqrt{p}$ olarak önerilmiştir (Breiman, 2001). p değeri ise bağımsız değişken sayısını temsil etmektedir. Bu işlemlerden sonra sınıflandırma ağaçları içinde en fazla oylanan tahmin, Şekil 13'te verildiği gibi regresyon ağaçları için ise oylamanın ortalamasına bakılarak kesin tahmin yapılır (Breiman, 2001).



Şekil 13. Random Forest Algoritması (Kirasich, Smith ve Sadler, 2018).

Her karar düğümünde öznitelikler iki dala ayrılır ve bu işlem son tahminin yapılması için kullanılan yaprak düğümlere ulaşıncaya kadar tekrarlanır. Her bir düğümde hangi özelliğin ayrılacağını belirlemek için entropi hesaplanır. Entropi, alt küme verilerinin homojenliğini ölçer; entropi bire eşitse, sınıf etiketleri eşit olarak bölünür, sıfır entropi ise numunenin tamamen homojen olduğu anlamına gelir. Yalnızca iki etiketli ikili sınıflandırma durumunda olduğu gibi, bölme, sınıf etiketlerinin tümünün 1 veya 0 olmasıyla sonuçlanırsa, entropi sıfır olacaktır. Benzer şekilde, etiketlerin yarısı 1 veya 0 ise, daha yüksek bir entropi 1 gözlenir. Entropi, her değişken için hesaplanır ve daha sonra, her bir değişken üzerinde bölünmeden önceki (ör. üst düğüm) ve bölmeden sonraki (ör. alt düğüm) entropi arasındaki fark hesaplanır. Entropide en fazla fark veren öznitelik, sınıf etiketlerini belirlemede en önemli etken olacaktır.

$$E(D) = - \sum_{k=1}^m p_i \log_2(p_i) \quad (6)$$

3.4.6.5 J48 Algoritması

Ross Quinlan tarafından geliştirilen bu algoritma WEKA veri madenciliği aracındaki C4.5 algoritmasının açık kaynaklı bir uygulamasıdır (Gholap, 2012). J48 algoritması bir çeşit karar ağacı algoritmasıdır.

Karar ağacı algoritmasında amaç farklı öznitelikler üzerinden birkaç örnek için nasıl davrandığının bulunması ve eğitim örneklerine bakılarak yeni örnekler için sınıflandırma yapılabilmesidir (Kaur ve Chhabra, 2014). Bu yöntemle sınıflandırma sürecini göstermek için bir ağaç oluşturularak kritik dallanmalar belirlenir ve oluşturulan tüm örneklere uygulanarak sınıflandırma işlemi gerçekleştirilir (Eraldemir, Arslan ve Yıldırım, 2017). J48 karar ağacı eksik değerleri, karar ağaçlarının budanmasını, sürekli öznitelik değer aralıklarını, kuralların türetilmesini vb. hesaba katar. Bu algoritma, verilerin sınıflandırılması için belirli bir kimliğinin üretildiği kuralları oluşturarak esneklik ve doğruluk dengesi kurulana kadar bir karar ağacının aşamalı olarak genelleştirilmesidir. (Kaur ve Chhabra, 2014).

J.48 algoritmasının temel adımları şu şekilde sıralanabilir: (Quinlan, 1992)

- (i) Örnekler aynı sınıfa ait olursa ağaç bir yaprağı temsil eder, böylece o yaprak aynı sınıfla etiketlenerek döndürülür.
- (ii) Öznitelik üzerinde bir test ile verilen her nitelik için potansiyel bilgi hesaplanır. Daha sonra, öznitelik üzerindeki bir testten kaynaklanacak bilgi kazancı (Entropi) hesaplanır (Korting, 2006).

$$Entropi(\vec{y}) = - \sum_{j=1}^n \frac{|y_j|}{|\vec{y}|} \log \frac{|y_j|}{|\vec{y}|} \quad (7)$$

$$Entropi(j|\vec{y}) = \frac{|y_j|}{|\vec{y}|} \log \frac{|y_j|}{|\vec{y}|} \quad (8)$$

$$Kazanç(\vec{y}, j) = Entropi(\vec{y}) - Entropi(j|\vec{y}) \quad (9)$$

- (iii) Daha sonra mevcut seçim kriteri temelinde en iyi nitelik bulunur ve dallanma için o öznitelik seçilir.

3.4.6.6 Naive Bayes Sınıflandırma Algoritması

Naive Bayes sınıflandırıcısının temelini 1760 yıllarında ortaya atılan Bayes teoremi oluşturmaktadır (Solmaz, Günay ve Alkan, 2014). Sınıflandırılacak verilerin hangi olasılıkla hangi kümeye (sınıflara) ait olduklarını tespit eder (Nizam ve Akın, 2014). Sarıkoç'a (2012) göre Naive Bayes sınıflandırıcısının iki temel varsayımı bulunmaktadır:

1. Özniteliklerin tamamı aynı ölçüde önemlidir ve

2. Özniteliklerin tamamı birbirinden bağımsızdır.

Belli bir öznitelik vektörü ile bu vektörün herhangi bir sınıfa(küme) sahip olma olasılığı hesaplanabilir. Bu sınıflayıcı, en iyi olasılığa sahip bulunan sınıf hangisi ise onu karar sınıfı olarak belirler.

Bu yöntemle pek çok yönteme oranla veriler optimum seviyede sınıflandırılmaktadır. Maksimum olasılık ilkesi dikkate alınarak sınıfın üstün ve büyük olasılıklara bağlı olma özelliği dikkate alınarak formülize edilir (Nur ve Ülker, 2020). I ve M 'nin rastgele olaylar olmak üzere:

$$P(I|M) = \frac{P(M|I)P(I)}{P(M)} \quad (10)$$

Bu denklemde $P(I)$ I 'nin bağımsız bir olasılığıdır, $P(M)$ M 'nin bağımsız bir olasılığıdır, $P(M|I)$ olay verildiğinde M olayının olabilme olasılığıdır. $P(I)$ ve $P(M)$ önceki olasılığa eşdeğerdir, $P(I|M)$ arka olasılığa eşdeğerdir (Nur ve Ülker, 2020).

Gerçek hayat koşullarında sınıflandırma yapılırken giriş değişkenlerinin tamamının birbirinden bağımsız olması mümkün değilse bile, Naives Bayes sınıflandırıcısının başarılı tahminler yaptığı bilinmektedir (Sarıkoc, 2012).

3.4.6.7 OneR (Very Simple Classification Rules)

OneR doğru sınıflandırma olasılığını iyileştirmeye çalışan, basit hata tabanlı kural çıkarımı yaklaşımını kullanan bir algoritmadır (Uzun, 2005). Aynı zamanda bir seviyeli karar ağacı ile öğrenen sınıflandırma algoritmasıdır. OneR en çok farklılık gösterebilecek özelliği seçerek içinden en az hatalı olanı en iyi özellik olarak öğrenir ve kullanır. (Parsania, Jani ve Bhalodiya, 2014). Bu algoritmaya ilişkin adımlar Şekil 14'te görülmektedir.

*Her bir özellik için,
{Özelliğin her bir değeri için, aşağıdaki gibi kuralı oluştur:
 {Her bir sınıfın ne kadar görüldüğünü say
 Çok sık görülen sınıfı bul
 Oluşturulan kuralı tanımla, bu özellik-değeri sınıflandır
 }
 Kuralların hata oranını hesapla
 }
Kuralları en küçük hata oranıyla seç*

Şekil 14. One-R Algoritması (Holte, 1993)

Kolay anlaşılır bir yapıya sahip, maliyeti az ve iyi kurallar öğrenebilmesi avantaj oluştursa da fazla miktarda alt kategorilerin olduğu öğrenimlerde aşırı öğrenme (overfitting) durumları ortaya çıkabilmektedir (Çakır, 2005).

3.5 Derin Öğrenme

Derin öğrenme, özellikle doğrudan erişilen verilerden öğrenebilen makine öğrenmesi tekniğidir. Bir veya daha fazla gizli katman içeren yapay sinir ağlarından oluşur. Veri miktarından fazlalık arttıkça makine öğrenmesi yetersiz kalmaktadır. Bunun sonucunda derin öğrenme daha iyi performans vermektedir.

Derin öğrenme ile ilgili ilk çalışma 1980’de oyuncak problemlerini çözmek için yapılmıştır. Günümüzdeki şekline gelmesi için yeterli veriler olmadığından gelişmesi zaman almıştır. Derin öğrenme ile ilgili araştırmalar geçmişe dayansa da 2012 yılından sonra yaygın olarak kullanılmaya başlanmıştır. Bu yaygınlaşmada, nesne tanımlama alanında en büyük yarışma olan Büyük Ölçekli Görsel Tanıma 2012 yarışmasına ESA ile katılan Krizhevsky, Sutskever ve Hinton’un yarışmayı kazanması etkili olmuştur. Big data kavramının gelişmesi ile derin öğrenme algoritmaları da gelişmiş ve bugünkü şeklini almıştır.

Derin öğrenme makine öğrenmesi algoritmalarını kullanarak beynin yapısına benzeyen yapay sinir ağları oluşturur. Beyinde nöronlara nasıl bilgi girişi oluyorsa derin öğrenmede de yapay sinir ağlarına bilgi girişi olmaktadır. Bilgi sinyaller işlenerek çıkışa yönlendirilir. Derin öğrenme yönteminin kullanılması internetin artık çok aktif kullanılması sonucu büyük verilerin ortaya çıkmasıdır. Büyük verilerle çalışmak için makine öğrenmesi yetersiz kalmaktadır. Bunun sonucunda derin öğrenme yöntemleri geliştirilmiştir. Derin öğrenme insan öğrenmesini modelleyerek ve hatta kendini daha çok geliştirmek için yapay sinir ağlarını kullanır. Derin öğrenme ile veriler üzerinde sınıflandırma, tahmin etme ve birçok işlem hızlı ve güvenli bir şekilde yapılabilmektedir.

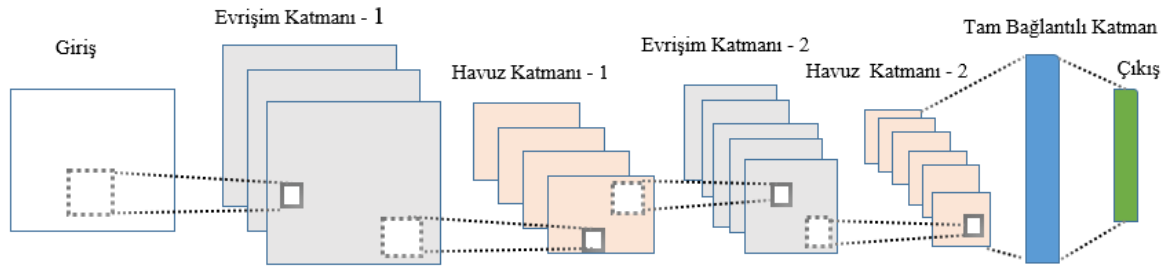
3.5.1 Evrişimli Sinir Ağı (ESA)

İlk olarak 1989 yılında Evrişimli Sinir ağı olarak LeCun tarafında ifade edilen, tek boyutlu zaman serileri ve iki boyutlu görsel verileri çözümlemek için kullanılan yapay sinir ağıdır. Evrişim kelimesi sinir ağında kullanılan matematiksel işlemi ifade eder. Ağdaki

katmanların birinde matris çarpımlarının yerine evrişim kullanılır (Goodfellow, Bengio ve Courville, 2016). Geleneksel evrişimli sinir ağlarında giriş, evrişim (convolution), havuzlama (pooling), tam bağlantı (full connected) ve çıkış katmanları bulunur. Bunlar genellikle bilgilerin analiz edilmesinde kullanılır.

Verilerin işlenmesinde ilk aşamada giriş değerleri belirli filtrelerle evrişim katmanında filtrelendirir. Daha sonra aktivasyon fonksiyonu kullanılarak öznitelik haritası oluşturulur. Evrişim katmanlarının her birinden sonra hesaplama işlemlerini daha basit yapmak için havuz katmanı kullanılır. Havuz katmanında önceki katmandaki nöronların hesapladığı değerler ile sonraki katmandaki tek nöron ile birleştirir. Bu şekilde boyut azaltma yapılarak örnekleme işlemi yapılmış olur. Havuz katmanında yöntem olarak maksimum ve ortalama işlemleri kullanılır. Maksimum yöntemde geçerli matrisin en yüksek değeri seçilerek algılanan özellikleri korunur.

Ortalama yönteminde ise matrisin ortalama değerleri alınır ve verilerin boyutu azaltılarak öznitelik haritası örneklendirilir. Evrişim ağının sonunda ve hedeflenen değerleri optimize ederken tam bağlantılı katman kullanılır. Evrişimli sinir ağının yapısı ve katmanları Şekil 15'te verilmiştir.



Şekil 15. Evrişimli Sinir Ağı Mimarisi (Sayracı, 2021)

Evrişimli sinir ağlarındaki katmanlarda (h) hesaplama işlemleri için giriş değerleri (x) ve ağırlıkları (w) çarpılıp toplanır. Daha sonra sapma değeri (b :bias) bu değere eklenerek aktivasyon fonksiyonu (h) hesaplanır.

$$h_i = f\left(\sum_i x_i + b\right) \quad (11)$$

Algoritmanın ağırlık güncellemesi ve optimizasyonu, gradyanların geri yayılmasıyla yapılır. Tam bağlantılı katmanlar softmax fonksiyonu ile çıktısı her bir sınıfın olasılığını oluşturur. En yüksek olasılığa (p) sahip sınıf çıktı olarak kabul edilir ve normalize üstel fonksiyon genellemesi aşağıdaki denklemle olduğu gibi hesaplanır:

$$p_i = \frac{e^{x_i}}{\sum_{j=1}^n e^{x_j}} \quad (12)$$

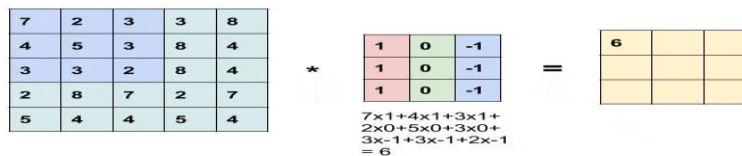
Fonksiyon, $x \in \mathbb{R}^n$ skorlarının bir vektörünü girdi olarak alıp en yüksek olasılığa sahip çıkış vektörünü oluşturur.

Evrişimli sinir ağı on katmanlı olan tam bağımlı katman hedeflenen değerleri optimize eder. Sınıflandırma yapmak için softmax kullanılır. Ayrıca regresyon sorunlarında sürekli bir değer bulmak için lineer fonksiyonları kullanılır. Sonuç olarak evrişimli sinir ağı mimarisi sürekli bir değeri kayıp (loss) fonksiyonu ile belirli bir hata oranıyla tahmin eder.

Genel olarak ESA modellerinin amacı problem çözmenin öğrenilmesinin sağlanmasıdır (Simard, Steinkraus ve Platt, 2003). Bu çalışmada da anket sonucunda oluşturulan veri setleri tek boyutlu 1DCONV kullanılarak sınıflandırma işlemi yapılmıştır.

3.5.1.1 Evrişim Katmanı

ESA algoritmalarının en temel katmanlarından birisidir. Görüntü üzerinde ilk işlem burada gerçekleşir. Gerçekleşen olay görüntü matrisleri üzerinde dolaşan, görüntü matrislerinden daha küçük boyutlu filtre matrisleri ile görüntülerin belirli özelliklerini elde etmektir. Elde edilen özellikler yüksek veya düşük seviyelerde olabilir. Filtre süzgeçlerinin kaymasıyla dönüşüm işlemi gerçekleştirilir. Evrişim katmanının çalışma prensibi Şekil 16'da görülmektedir (Kızırak, 2018).

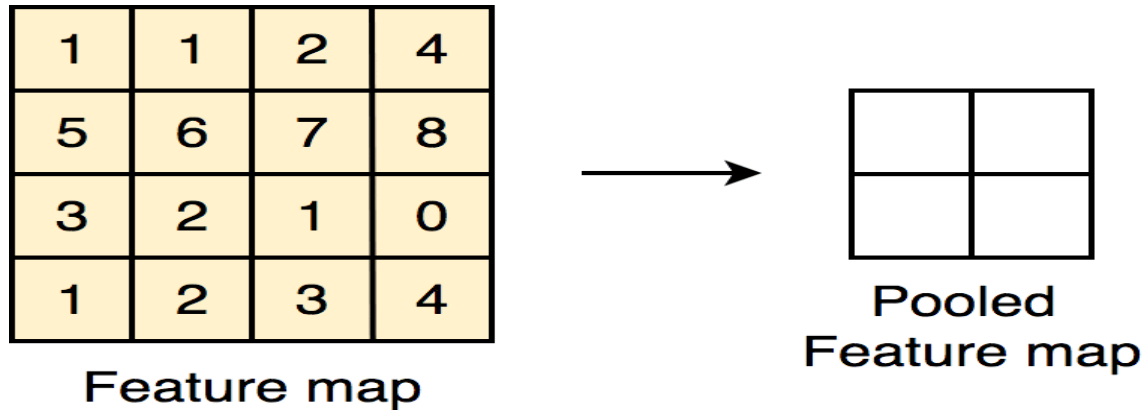


Şekil 16. Evrişim Katmanı (Kızırak, 2018)

Kullanacağımız görüntü 5x5 lik bir matristir. Uygulanacak filtrenin boyutu ise 3x3 lük bir matristir. Filtreleme işlemi yaparken ilk olarak filtre görüntüsünü sol üste koyarız. İkinci aşamada iki matrisin indisleri çarpılır ve tüm sonuçlar toplanır. Elde edilen sonuç çıktı matrisinin ilk elamanı olur. Üçüncü aşamada filtre matrisi bir piksel sağa kaydırılarak işlemler tekrarlanır. İlk satırdaki işlemler bitince ikinci satırdaki işlemler tekrar edilir. Tüm satırlardaki işlemler bitince çıktı matrisi oluşur. Bu işlem ile görüntünün üstündeki özellikler saptanmış olur. Evrişim ağlarında modellerin öğrendiği parametreler filtreleme sonucu oluşan değerlerdir. Modeller sürekli olarak bu değerleri güncelleyerek özellikleri daha iyi algılamaya çalışır. Görüntülerde tek bir özellik olmadığından dolayı birden fazla filtre kullanılır. Bundan dolayı evrişim ağları birçok katmandan oluşmaktadır (Kızrak, 2018).

3.5.1.2 Havuzlama Katmanları (Pooling Layers)

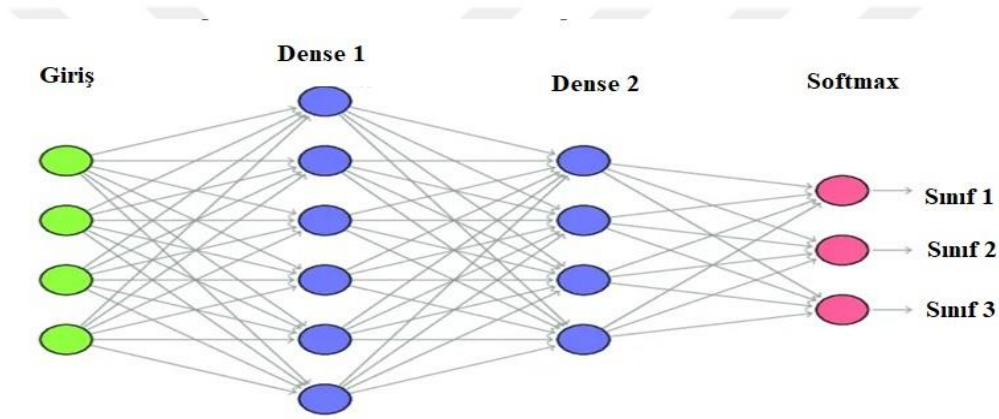
Evrişim katmanında işlemleri tamamlanan görüntüler havuzlama katmanına gönderilir. Görüntülerin çok büyük olduğu durumlarda parametre sayısını azaltarak görüntüyü küçültmeye yarayan katmana havuzlama katmanı denir. Maksimum havuzlama ve ortalama havuzlama olmak üzere iki türü vardır. Genel olarak maksimum havuzlama kullanılır. Maksimum havuzlamada evrişim katmanından sonra giriş verisinin uzamsal boyutu (yüksekliği ve genişliği) azaltılmak için kullanılır (Kızrak, 2018). Boyut azaltılarak baskın özellikler ortaya çıkarılır. Modelin etkili bir şekilde çalışması sağlanır. Havuzlama katmanının çalışma prensibi Şekil 17’de görülmektedir.



Şekil 17. Havuzlama (Pooling) Katmanı (Kızrak, 2018)

3.5.1.3 Tam Bağlantı Katmanları (Fully Connected Layers)

Evrişimli sinir ağlarında bulunan son katmandır. Bütün düğüm ve nöronların sonraki katmanlarla bağlandığı yapay sinir ağıdır. Yapay sinir ağındaki hesaplamaların karmaşık olmasında dolayı aşırı yüklenme işlemi gerçekleşebilir. Hesaplama sonucu elde edilen verilere Düzleştirme (Flattening) işlemi yapılarak sinir ağlarına gönderilir ve öğrenme işlemi uygulanır. Tam bağlantı katmanının amacı çok kez evrişimli katman ile havuzlama katmanlarından geçen matris biçimindeki veriler, düz bir vektör haline getirilir (Kızrak, 2018). Tam bağlantı katmanının çalışması Şekil 18’de belirtilmiştir.



Şekil 18. Tam Bağlantılı Katman (Kızrak, 2018)

3.5.1.4 Aktivasyon Fonksiyonları

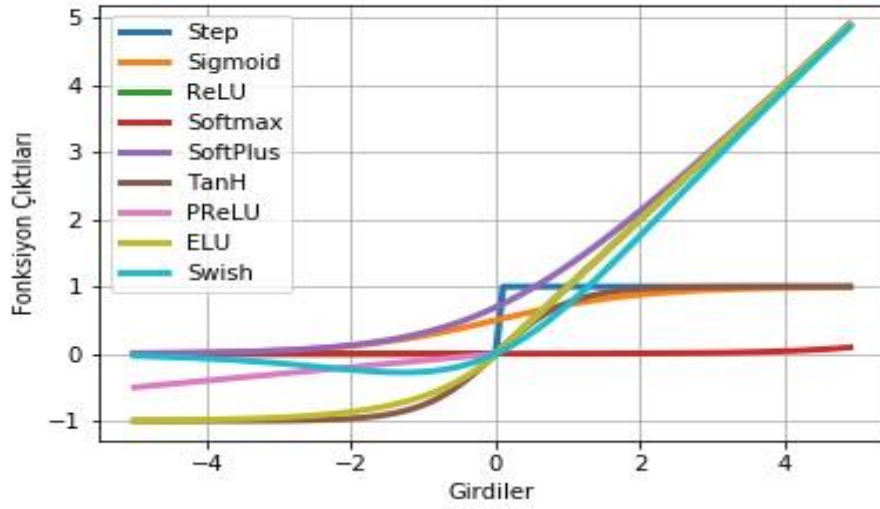
Yapay bir ağda n sayıda girdi ve tek bir çıktı veren ağ elemanına nöron denir. Bir yapay nöron x girdileri ve w ağırlıklarını toplar. Bias değerini ekleyerek y çıktısını hesaplar. y 'nin formülü Denklem 13’te gösterilmiştir.

$$y = \text{Aktivasyon} \left(\sum (w * x + b) \right) \quad (13)$$

Aktivasyon fonksiyonları y değerini kontrol etmek için kullanılır. Nöronların aktif olarak çalışıp çalışmayacağına karar verir. Evrişimli sinir ağlarının çalışmasında önemli bir yere sahiptir. Bu çalışmada ReLu ve Softmax aktivasyon fonksiyonları kullanılmıştır. En sık kullanılan aktivasyon fonksiyonları şu şekildedir (Karakuş, 2018):

- Step Fonksiyonu: Bir eşik değeri olarak ikili bir sınıflandırma çıktısı (0 ya da 1) oluşturur.
- Sigmoid Fonksiyonu: En sık kullanılan aktivasyon fonksiyonlarından birisidir ve $[0,1]$ aralığında bir çıktı oluşturur.
- Tanh Fonksiyonu: $[-1,1]$ aralığında çıktı üreten doğrusal olmayan bir aktivasyon fonksiyondur.
- ReLU Fonksiyonu: Doğrultulmuş lineer birim (rectified linear unit- RELU) doğrusal olmayan bir aktivasyon fonksiyondur. ReLU fonksiyonu negatif girdiler için 0 değerini kullanırken, x pozitif girdiler için x değerini kullanmaktadır.
- Softmax Fonksiyonu: Çoklu sınıflandırma problemleri için kullanılan bu fonksiyon, verilen her bir girdinin bir sınıfa ait olma olasılığını gösteren $[0,1]$ değerleri arasında çıktılar oluşturur.
- Softplus Fonksiyonu: Sigmoid ve Tanh gibi geleneksel aktivasyon fonksiyonlarına alternatif olarak sunulan bu fonksiyon $(0, +\infty)$ aralığında türevlenebilir bir çıktı oluşturmaktadır.
- ELU Fonksiyonu: Üstel lineer birim (Exponential Linear Unit-ELU), Negatif girdiler hariç ReLU ile aynıdır. Negatif girdilerde ise genellikle 1.0 alınan alfa parametresi kullanılmaktadır.
- PReLU Fonksiyonu: Parametrik ReLU olarak geçen bu fonksiyon da negatif girdiler için extra alfa sabiti ile verilen girdinin çarpım sonucu çıktı olarak kullanılmaktadır
- Swish Fonksiyonu: Google için araştırma yapan kullanıcılar tarafından yeni bulunan bu fonksiyon girdiler ile sigmoid fonksiyonunun çarpımını çıktı olarak kullanmaktadır.

Aktivasyon fonksiyonlarına göre girdi ve çıktı değerleri Şekil 19' da görülmektedir.



Şekil 19. Aktivasyon Fonksiyonları Girdi ve Çıktı Değerleri (Karakuş, 2018)

3.6 Değerlendirme Ölçütleri

3.6.1 Değerlendirme Kriterleri

Makine öğrenmesi yöntemlerinde sınıflandırma işlemlerinin daha iyi anlaşılmasını sağlamak amacıyla karmaşıklık (hata) matrisi kullanılarak görselleştirme işlemi yapılmaktadır (Feng vd., 2020). Hata matrisi, gerçek durumlar ile veri seti aracılığıyla programın yaptığı sınıflandırma tahminlerini içermektedir. Hata matrisleri, sınıflandırma yöntemlerinin performanslarını karşılaştırmak için doğru ve yanlış değerleri kullanmaktadır. Bu matriste genellikle satırlarda veriye ilişkin gerçek değerler bulunurken sütunlarda ise programın yaptığı tahminler yer almaktadır.

Sınıflandırma algoritmaları kullanıldığında hata matrisinde dört farklı sonuç oluşur (Fawcett ve Flach, 2005):

- Doğru pozitif (DP): Sınıflandırma algoritmasının doğru tahmin ettiği pozitif verileri gösterir.
- Doğru negatif (DN): Sınıflandırma algoritmasının doğru tahmin ettiği negatif verileri gösterir.

- Yanlış pozitif (YP): Gerçekte negatif (yanlış) olan bir verinin algoritma tarafından pozitif (doğru) olarak tahmin edildiğini söylemektedir. Bu durumda negatif (yanlış) örneğin yanlış sınıflandırılmasıdır.
- Yanlış negatif (YN): Gerçekte pozitif (doğru) olan bir verinin algoritma tarafından negatif (yanlış) olarak tahmin edildiğini söylemektedir. Bu durumda pozitif (doğru) örneğin yanlış sınıflandırılmasıdır.

Tablo 2’de görüldüğü gibi veri seti ve sınıflandırma sonucunda tahmin edilen sonuçların ne şekilde olacağı belirtilmiştir (Visa, Ramsay, Ralescu ve Van Der Knap, 2011).

Tablo 2. Karmaşıklık (Hata) Matrisi

		Tahmin Edilen Sınıf	
		Pozitif	Negatif
Gerçek Sınıf	Pozitif	Doğru Pozitif (DP)	Yanlış Negatif (YN)
	Negatif	Yanlış Pozitif (YP)	Doğru Negatif (DN)

Veri seti üzerinde yapılan deneysel çalışmaların daha nesnel değerlendirilmesi için cross validation (çapraz geçerlilik) yöntemi kullanılmıştır. Bu yöntem ile veri seti test ve eğitim olmak üzere rastgele iki kısma ayrılmıştır. Çapraz geçerliliğin başlıca yöntemlerinden biri olan k -fold yöntemi kullanılmıştır. Veri seti rastgele k gruba ayrılmıştır. Veri setinden seçilen grup validasyon seti olarak belirlenir. Diğer gruplar ise ($k-1$ grup) eğitim (train) seti olarak kullanılır. Eğitim seti ile model oluşturulur ve validasyon seti ile değerlendirilir. Modelin sonuçları bir listede tutulur. k farklı değerlendirme sonuçların ortalaması alınarak modelin performansı bulunur.

Modelin performans değerlendirmesi için ilk olarak doğruluk sonuçlarına bakıldı. Diğer kriterler olarak duyarlılık ve özgüllük sonuçlarına bakıldı. Doğruluk, duyarlılık ve özgüllük için 14, 15 ve 16’daki eşitlikler kullanılmıştır.

$$\text{Doğruluk} = \frac{TP+TN}{TP+FP+FN+TN} \times 100 \quad (14)$$

$$\text{Duyarlılık} = \frac{TP}{TP+FN} \times 100 \quad (15)$$

$$\text{Özgüllük} = \frac{TN}{TN+FP} \times 100 \quad (16)$$

3.6.1.1 Doğruluk

Veri seti üzerinde doğru sınıflandırılmış örneklerin toplamının verinin tamamına bölünmesi sonucunda doğruluk oranı oluşur. Hesaplama işlemi yapılırken tüm hata türleri dikkate alınarak aynı oranda işleme alınır. 14'deki eşitlikteki formülde de görüldüğü gibi tüm performansın değerlendirilmesi için doğruluk oranı kullanılır. Ayrıca veri setinin dengesiz dağıldığı durumlarda bu değer tam anlamıyla doğru sonuç vermez. Doğruluk oranı ne kadar yüksek ise o kadar yüksek performanslı bir sonuca ulaşıldığı söylenebilir.

$$\text{Doğruluk} = \frac{TP+TN}{TP+FP+FN+TN} \times 100 \quad (14)$$

3.6.1.2 Hassasiyet (Duyarlılık)

Veri seti üzerinde toplam pozitif olarak tahmin edilen örneklerin doğru tahmin edilme oranını vermektedir. Bundan dolayı hassasiyet yani duyarlılık doğru sınıflandırılan pozitif örneklerin pozitif olarak tahmin edilen toplam örnek sayısına bölünmesiyle ele edilir. Duyarlılık formülüne 15'deki eşitlikte yer verilmektedir.

$$\text{Duyarlılık} = \frac{TP}{TP+FN} \times 100 \quad (15)$$

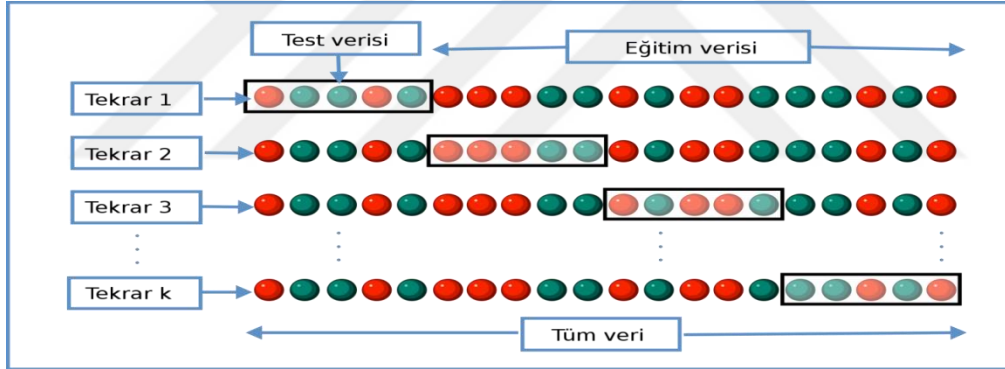
3.6.1.3 Özgüllük (Belirginlik)

Özgüllük veri setindeki negatif sınıflandırma başarısının oranını ölçer. Bundan dolayı doğru negatif değerinin doğru negatif ve yanlış negatif toplamına bölümünden sonra oluşan orandır. Özgüllük formülü 16'daki eşitlikte yer almaktadır.

$$\text{Özgüllük} = \frac{TN}{TN+FP} \times 100 \quad (16)$$

3.6.1.4 Çapraz Geçerlilik (Cross Validation)

Makine öğrenmesindeki modellerin görmediği veriler üzerinde nesnel ve doğru değerlendirme yapmak için istatistiksel örnekleme (resampling) yöntemidir (Şimşek, 2018). Diğer bir kullanım alanı ise modelde hiperparametre optimizasyonudur. Objektif bir şekilde değerlendirilme yapabilmek için her bir veri seti çapraz geçerlilik yöntemi ile eğitim ve test veri setlerine ayrılmıştır. Eğitim ve test veri setleri ilk olarak rastgele bir şekilde ayrılmıştır. Temel çapraz geçerlilik yöntemlerinden birisi olan k fold yönteminde veri seti rastgele karıştırılır (Durna, 2020). Ardından veri seti k sayıda gruba bölünür. Bu gruplardan biri validasyon seti olarak seçilir. Kalan gruplar ise ($k - 1$ grup) test seti olarak kullanılır. Test seti kullanılarak modelleme yapılır ve validasyon seti kullanılarak değerlendirilir (Durna, 2020). Modelin değerlendirme puanı bir listeye kaydedilir. Son aşamada k farklı değerlendirme puanının ortalaması alınarak modelin performansı belirlenir. Çapraz geçerlilik yöntemine ilişkin çalışma prensibi Şekil 20’ de verilmiştir.



Şekil 20. Çapraz Geçerlilik Prensibi

3.7 Geliştirme Ortamları

3.7.1 Çalışmada Kullanılan IDE/Programlama Dili/Kütüphane

Bu çalışmada klasik makine öğrenmesi yöntemleri için WEKA programı kullanılmıştır. WEKA (Attribute Relationship File Format) Yeni Zelanda’da bulunan Waikoto Üniversite tarafından Java programlama dili ile yazılmış ücretsiz bir yazılımdır. Genel olarak makine öğrenmesi için bu yazılım kullanılmaktadır. Herkese açık şekilde kullanılan bu yazılımda

sınıflandırma, kümeleme ve ilişkilendirme işlemleri yapılabilir. Weka yazılımını kullanabilmek için ARFF, CVS, xls formatlarında veri dosyaları kullanmalıyız.

Çalışmada kullanılan derin öğrenme mimarileri için Google Colab isimli platform kullanılmıştır. Google hesabı ile ücretsiz olarak kullanılabilen Google Colab 12 saate kadar çalışma olanağı sağlar. Bilgisayarı yormadan sanal olarak GPU, CPU ve RAM desteği vererek veri seti üzerinde hızlı bir şekilde işlem yapılmasını sağlar. Sınıflandırma işlemlerinde Python 3.7 programlama dili kullanılmıştır. Programda en çok kullanılan kütüphaneler aşağıda açıklanmıştır.

Keras Kütüphanesi: Keras, 2017 yılında Google’da çalışan mühendis François Chollet tarafından geliştirilmiştir (Narin ve İşler, 2021). Keras Python ile yazılmış Tensorflow tabanlı bir Derin Öğrenme API’sidir. Hızlı denemeler yapmayı amaçlamaktadır. Çalışmamızda keras kütüphanesi içerisinde aktivasyon (activation) katmanı, düzleştirme (flatten) katmanı, seyreltme (dropout) katmanları kullanılmıştır. Aktivasyon fonksiyonu olarak Relu, drop out değeri olarak 0.1 ve optimizer fonksiyonu olarak ise ‘Adam’ belirlenmiştir.

4. BULGULAR

Bu çalışmada Bilişim Teknolojileri öğretmenlerinin Scratch'e yönelik özyeterliklerini tespit etmek için TSECT anketi aracılığıyla elde edilen sonuçlara 10 farklı klasik makine öğrenmesi yöntemi ve derin öğrenme yöntemlerinden evrişimli sinir ağı kullanılmıştır. 39 maddeli ölçek 192 kişiyi uygulanmıştır. Pandemiden dolayı sınırlı sayıda veri toplanmıştır. Sınıflar arası küçük dengesizliklerden dolayı Smote methoduyla veri sayı artırılmıştır. 262 kişiye kadar artırılmış ve sınıflar arası dengesizlikler ortadan kaldırılmıştır. Derin öğrenme yöntemlerinden evrişimli sinir ağlarında eğitim tur sayısı (epoch) 100 olarak belirlenmiştir. Modelin değerlendirmesinde klasik makine ve derin öğrenmesi yöntemleri için öncelikli olarak doğruluk, duyarlılık ve özgüllük değerlerine bakılmıştır.

Çalışmamızda oluşturduğumuz veri setine 10 farklı klasik makine öğrenmesi yöntemi çapraz geçerlik olarak 10-fold uygulanmış ve sonuçları Tablo 3'te görülmektedir.

Tablo 3. Klasik Makine Öğrenmesi Yöntemlerinin DeneySEL Sonuçları

Sınıflandırma	Çapraz Geçerlilik (Fold)	Duyarlılık	Özgüllük	Doğruluk
NAIVEBAYES	10	0.911	0.854	%87.40
SMO	10	0.986	0.993	%98.85
MLP	10	0.978	0.944	%95.80
RF	10	0.953	0.968	%96.05
J48	10	0.959	0.968	%96.42
K-STAR	10	0.978	0.910	%93.89
IBK	10	0.966	0.993	%97.70
LOGİSTİC	10	0.924	0.908	%91.60
RANDOM TREE	10	0.959	0.951	%95.48
ZERO-R	10	0.550	0.936	%54.96

Veri seti üzerine uygulanan yöntemlerden en başarılı sınıflandırma performansı gösteren beş yöntem belirlenmiştir. Bu beş yönteme ait karmaşıklık hata matrislerinin sonuçları ve parametreleri şu şekildedir:

4.1 SMO Parametreleri ve Deneysel Sonuçları

SMO sınıflandırma algoritmasında kullanılan parametreler açıklamaları ve değerleri ile verilmiştir:

- BuildCalibrationModels: Kalibrasyon modellerinin SVM'nin çıktılarına uygun olup olmayacağını belirtir. Uygun olasılık tahminleri için kullanılır.
- NumFolds: Kalibrasyon modelleri için eğitim verileri oluşturmak için kullanılan çapraz doğrulama katsayısıdır. -1 kullanılırsa eğitim verileri anlamına gelir.
- RandomSeed: Çapraz doğrulama için rastgele sayı çekirdeği.
- C: Karmaşıklık parametresi.
- BatchSize: Toplu tahmin gerçekleştiriliyorsa işlenecek tercih edilen örnek sayısını belirtir. Daha fazla veya daha az örnek sağlanabilir, ancak bu, uygulamalara tercih edilen bir toplu iş boyutunu belirleme şansı verir.
- Kernel (Çekirdek): Kullanılacak çekirdek.
- FilterType: Verilerin nasıl dönüştürüleceğini/dönüştürülmeyeceğini belirler.
- Calibrator: Kullanılacak kalibrasyon yöntemi.
- DoNotCheckCapabilities: Ayarlanırsa, sınıflandırıcı oluşturulmadan önce sınıflandırıcı yetenekleri kontrol edilmez (Çalışma zamanını azaltmak için dikkatli kullanın). Parametrelerin değeri Tablo 4'te verilmiştir.

Tablo 4. SMO Parametreleri Değerleri

Parametre adı	Değeri
BuildCalibrationModels	False
NumFolds	-1
RandomSeed	1
C	1.0
BatchSize	100
Kernel (Çekirdek)	Poly Kernel
FilterType	Normalize Training Data
Calibrator	Logistic
DoNotCheckCapabilities	False

Veri seti üzerinde yapılan deneysel çalışmalarda WEKA programında sınıflandırma için SMO yöntemi kullanılmıştır. Normal şartlarda veri setinde 144 yeterli değil ve 118 yeterli sonuçları vardır. SMO algoritmasını uyguladığımızda Tablo 5’de görüldüğü üzere 143 yeterli değil ve 116 yeterli sonuçlarıyla veriyi sınıflandırmıştır. Bu çalışmada doğruluk oranı %98.85 olarak bulunmuştur. Çapraz geçerlilik (fold) sayısı 10 olarak belirlenmiştir.

Tablo 5. SMO Karmaşıklık Matrisi

		Gerçek Sınıf	
		Pozitif	Negatif
Tahmin Edilen Sınıf	Pozitif	143 (DP)	2 (YN)
	Negatif	1 (YP)	116 (DN)

4.2 J48 Parametreleri ve Deneysel Sonuçları

J48 sınıflandırma algoritmasında kullanılan parametreler açıklamaları ile verilmiştir:

- Seed (tohum): Hatalı budama kullanıldığında verileri rastgele sıralamak için kullanılan tohum.
- Unpruned: Budama yapılıp yapılmadığı.
- ConfidenceFactor: Budama için kullanılan güven faktörü (daha küçük değerler daha fazla budamaya neden olur).
- NumFolds: Hataları azaltılmış budama için kullanılan veri miktarını belirler. Bir kat budama için, geri kalanı ağacı büyütmek için kullanılır.
- NumDecimalPlaces: Modeldeki sayıların çıktısı için kullanılacak ondalık basamak sayısı.
- BatchSize: Toplu tahmin gerçekleştiriliyorsa işlenecek tercih edilen örnek sayısı. Daha fazla veya daha az örnek sağlanabilir, ancak bu, uygulamalara tercih edilen bir toplu iş boyutunu belirleme şansı verir.

Parametrelerin değeri Tablo 6’da verilmiştir.

Tablo 6. J48 Parametre Değerleri

Parametre adı	Değeri
Seed (tohum)	1
Unpruned	True
ConfidenceFactor	0.7
NumFolds	3
NumDecimalPlaces	10
BatchSize	200

Veri seti üzerinde yapılan deneysel çalışmalarda WEKA programında sınıflandırma için J48 yöntemi kullanılmıştır. Normal şartlarda veri setinde 144 yeterli değil ve 118 yeterli sonuçları vardır. J48 algoritmasını uyguladığımızda Tablo 7’de görüldüğü üzere 141 yeterli değil ve 112 yeterli sonuçlarıyla veriyi sınıflandırmıştır. Bu çalışmada doğruluk oranı %96.42 olarak bulunmuştur.

Tablo 7. J48 Karmaşıklık (Hata) Matrisi

		Gerçek Sınıf	
		Pozitif	Negatif
Tahmin Edilen Sınıf	Pozitif	141 (DP)	6 (YN)
	Negatif	3 (YP)	112 (DN)

4.3 MLP Parametreleri ve Deneysel Sonuçları

MLP sınıflandırma algoritmasında kullanılan parametreler açıklamaları ile verilmiştir:

- Seed (tohum): Rastgele sayı üreticini başlatmak için kullanılır. Rastgele sayılar, düğümler arasındaki bağlantıların başlangıç ağırlıklarını ayarlamak ve ayrıca eğitim verilerini karıştırmak için kullanılır.
- Momentum: Ağırlık güncellemelerine uygulanan ağırlıktır.

- **NominalToBinaryFilter:** Örnekler NominalToBinary filtresiyle önceden işlenir. Verilerde nominal nitelikler varsa performansı artırmaya yardımcı olabilir.
- **HiddenLayers:** Bu, sinir ağının gizli katmanlarını tanımlamak için kullanılır. Bu, pozitif tam sayıların bir listesidir. Her gizli katman için 1 virgülle ayrılmıştır. Gizli katman olmaması için buraya tek bir 0 koyulur. Bu katman tanımı yalnızca otomatik oluşturma ayarlanmışsa kullanılacaktır. Ayrıca joker değerler de vardır. Bunlar şu şekildedir: “a” = (nitelikler + sınıflar) / 2, “i” = nitelikler, “o” = sınıflar , “t” = nitelikler + sınıflardır.
- **BatchSize:** Toplu tahmin gerçekleştiriliyorsa işlenecek tercih edilen örnek sayısı belirlenir. Daha fazla veya daha az örnek sağlanabilir, ancak bu, uygulamalara tercih edilen bir toplu iş boyutunu belirleme şansı verir.
- **LearningRate:** Öğrenme oranı miktarını verir

Parametrelerin değeri Tablo 8’de verilmiştir.

Tablo 8. MLP Parametre Değerleri

Parametre adı	Değeri
Seed (tohum)	0
Momentum	0.2
NominalToBinaryFilter	True
HideLayers	a
BatchSize	100
LearningRate	0.3

Veri seti üzerinde yapılan deneysel çalışmalarda WEKA programında sınıflandırma için MLP yöntemi kullanılmıştır. Normal şartlarda veri setinde 144 yeterli değil ve 118 yeterli sonuçları vardır. MLP algoritmasını uyguladığımızda Tablo 9’da görüldüğü üzere 137 yeterli değil ve 114 yeterli sonuçlarıyla veriyi sınıflandırmıştır. Bu çalışmada doğruluk oranı %95.80 olarak bulunmuştur.

Tablo 9. MLP Karmaşıklık (Hata) Matrisi

		Gerçek Sınıf	
		Pozitif	Negatif
Tahmin Edilen Sınıf	Pozitif	137 (DP)	4 (YN)
	Negatif	7 (YP)	114 (DN)

4.4 Random Forest Parametreleri ve Deneysel Sonuçları

Random forest sınıflandırma algoritmasında kullanılan parametreler açıklamaları ile verilmiştir:

- Seed (tohum): Kullanılacak rasgele sayı çekirdeği belirtir.
- NumIterations: Rastgele ormandaki ağaç sayısını belirtir.
- BagSizePercent: Eğitim seti boyutunun yüzdesi olarak her çantanın boyutu belirtilir.
- BatchSize: Toplu tahmin gerçekleştiriliyorsa işlenecek tercih edilen örnek sayısını belirtir. Daha fazla veya daha az örnek sağlanabilir, ancak bu, uygulamalara tercih edilen bir toplu iş boyutunu belirleme şansı verir.
- MaxDepth: Ağacın maksimum derinliğini belirtir. Sınırsız için 0 kullanılır.
- NumFeatures: Rastgele seçilen niteliklerin sayısını belirtir.

Parametrelerin değeri Tablo 10’da verilmiştir.

Tablo 10. Random Forest Parametre Değerleri

Parametre adı	Değeri
Seed (tohum)	15
NumIterations	65
BagSizePercent	100
BatchSize	100
MaxDepth	1
NumFeatures	0

Veri seti üzerinde yapılan deneysel çalışmalarda WEKA programında sınıflandırma için Random Forest yöntemi kullanılmıştır. Normal şartlarda veri setinde 144 yeterli değil ve 118 yeterli sonuçları vardır. Random Forest algoritması uygulandığında Tablo 11’de görüldüğü üzere 140 yeterli değil ve 112 yeterli sonuçlarıyla veriyi sınıflandırmıştır. Bu çalışmada doğruluk oranı %96.05 olarak bulunmuştur.

Tablo 11. Random Forest Karmaşıklık(Hata) Matrisi

		Gerçek Sınıf	
		Pozitif	Negatif
Tahmin Edilen Sınıf	Pozitif	140 (DP)	6 (YN)
	Negatif	4 (YP)	112 (DN)

4.5 IBK Matrisi

IBK sınıflandırma algoritmasında kullanılan parametreler açıklamaları ile verilmiştir:

- KNN: Kullanılacak komşu sayısını belirtir.
- BatchSize: Toplu tahmin gerçekleştiriliyorsa işlenecek tercih edilen örnek sayısını belirtir. Daha fazla veya daha az örnek sağlanabilir, ancak bu, uygulamalara tercih edilen bir toplu iş boyutunu belirleme şansı verir.
- NearestNeighbourSearchAlgorithm: Kullanılacak en yakın komşu arama algoritmasını belirtir. Varsayılan olarak LinearNNSearch kullanılmıştır.
- WindowSize: Eğitim havuzunda izin verilen maksimum örnek sayısını alır. Bu değerin üzerine yeni örneklerin eklenmesi, eski örneklerin kaldırılmasına neden olur. 0 değeri, eğitim örneklerinin sayısında sınır olmadığını belirtir.
- CrossValidate: 1 ile KNN parametresi olarak belirtilen değer arasındaki en iyi k değerini seçmek için bir-arada tutma çapraz doğrulamasının kullanılıp kullanılmayacağı belirtir.

Parametrelerin değeri Tablo 12’de verilmiştir.

Tablo 12. IBK Parametre Değerleri

Parametre adı	Değeri
KNN	2
BatchSize	100
NearestNeighbourSearchAlgorithm	LinearNNSearch
WindowSize	0
CrossValidate	True

Veri seti üzerinde yapılan deneysel çalışmalarda WEKA programında sınıflandırma için IBK yöntemi kullanılmıştır. Normal şartlarda veri setinde 144 yeterli değil ve 118 yeterli sonuçları vardır. IBK algoritmasını uyguladığımızda Tablo 13’de görüldüğü üzere 143 yeterli değil ve 113 yeterli sonuçlarıyla veriyi sınıflandırmıştır. Bu çalışmada doğruluk oranı %97.70 olarak bulunmuştur.

Tablo 13. IBK Karmaşıklık(Hata) Matrisi

		Gerçek Sınıf	
		Pozitif	Negatif
Tahmin Edilen Sınıf	Pozitif	143 (DP)	5 (YN)
	Negatif	1 (YP)	113 (DN)

4.6. Evrişimli Sinir Ağı Deneysel Sonuçları ve Parametreler

Evrişimli sinir ağları evrişim işlemini kullanan yapay bir sinir ağıdır. Izgaraya benzeyen geometrik yapılardan oluşur (Goodfellow, McDaniel ve Papernot, 2018). Adını matematiksel olarak evrişim işlemi yapması nedeniyle almıştır. Evrişimli sinir ağları genel olarak görüntü işleme üzerinde işlemler yapmakta kullanılsa da metin sınıflandırması için de kullanılabilir. Bunun için Keras Kütüphanesi kullanılmaktadır. Keras ’ta görüntü verilerinde Conv2d katmaları kullanılırken metinler üzerinde Conv1d katmanı kullanılmaktadır. Conv2d iki taraflı hareket edebilen kernel’e sahip 3 boyutlu görüntü

verilerinde kullanılır. Conv1d ise tek taraflı hareket edebilen 2 boyutlu veriler üzerinde kullanılır. Conv1d ayrıca zaman serileri, ses ve metin analizlerinde de kullanılmaktadır.

Bu çalışmada evrişimli sinir ağı için ilk aşamada Python programında yer alan Keras Kütüphanesinden yararlanılmıştır (Ülgen, 2020). Model oluşturmak için Sequential yöntemi kullanılmıştır. Sequential, model için sıralı katmanlar oluşturulmasını sağlar. Veri setimiz metin olduğu için Conv1d katmanı tercih edilmiştir. Modelimiz 3 evrişim katmanından oluşmuştur. Evrişim katmanlarında 32, 64, 96 ve 128 filtreleri kullanılmıştır. Kernel sayısı olarak ise 2 kernel değeri verilmiştir. Kernel nitelik haritası üzerinde hareket eden pencerelerdir. Conv1d 2 kernel 2 niteliği temsil eder. Aktivasyon fonksiyonu olarak ise “relu” fonksiyonu kullanılmıştır. Relu fonksiyonu kullanılmasının sebebi işlem yükü diğer fonksiyonlara göre daha az olmasıdır. Sonuç değeri $[0, +\infty)$ arasındadır. Bir sonraki aşama “fully connected (tam bağlantı)” olarak tanımlanan sınıflandırma işlemidir. Yapay sinir ağlarını beslemek için Flatten işlemi yapılmıştır. Tam bağlantı bölümü iki kısımdan oluşmaktadır. İlk kısımda 128 düğüm mevcuttur ve aktivasyon fonksiyonu olarak “relu” kullanılmıştır. İkinci kısımda yani çıktı katmanımızda birden fazla sonuç üzerinde sınıflandırma yaptığımız için çıkış katmanında aktivasyon fonksiyonu olarak softmax kullanılmıştır. Çıktı katmanında 2 düğüm yer almaktadır. Çünkü 2 farklı sınıf bulunmaktadır. Sonuç olarak her girdi için 2 boyutlu bir vektör oluşacak ve elde edilen değerlerin hangi sınıfa ait olup olmadığının olasılığı bulunacaktır (Ülgen, 2020).

Sonuç olarak 1 tam olasılık 2 ayrı sınıfa paylaştırılmıştır.

“model.fit(xtrain, ytrain, batch_size=16, epochs=100, verbose=1, validation_data=(xtest, ytest))”

Yukarıda görüldüğü gibi modelin eğitimi sırasında verilerin hepsi aynı anda eğitime dahil edilmez (Ülgen, 2020). “Batch_size” olarak adlandırılmış parçalar şeklinde eğitimde dahil edilir. Bu örnekte modelimiz 100 eğitim tur sayısı, 16’lık küçük bölümler kullanılarak eğitilmiştir.

Bu çalışma sonucunda elde edilen başarı oranı %99.6 olarak belirlenmiştir. Farklı eniyileme algoritmaları veya katman sayılarıyla bu değerler artabilir ve azalabilir. Çalışmanın sonucu oluşun doğruluk, duyarlılık ve özgüllük değerleri Tablo 14’te görülmektedir.

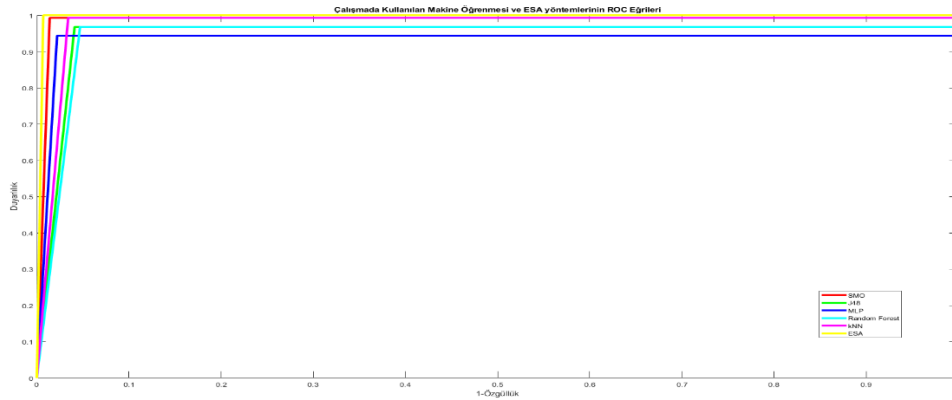
Tablo 14. Evrişimli Sinir Ağı Deneysel Sonuçları

Sınıflandırma	Çapraz Geçerlilik(fold)	Duyarlılık	Özgüllük	Doğruluk
Evrişimli Sinir Ağı	10	0.993	1.00	%99.6

ESA modeli ile elde edilen sonuçların tüm klasik makine öğrenmesi modellerinden elde edilen sonuçlara göre daha başarılı olduğu Tablo 15'te görülmektedir. ESA yöntemi ile %99.6 doğruluk, %99.3 duyarlılık ve %100 özgüllük değerlerine ulaşılmıştır. ESA yönteminin klasik makine öğrenmesi modellerine göre daha iyi sonuçlar vermesinin temel nedeni çok katmanlı yapıdan oluşmasıdır. Evrişim katmanında kullanılan veri kümesindeki düşük ve yüksek seviyeli öznelilikler ön plana çıkarılarak sınıflandırma başarısının artırılması sağlanmıştır.

Tablo 15. Klasik makine öğrenmesi yöntemleri ve Evrişimli Sinir Ağı Sonuçları

Sınıflandırma	Çapraz Geçerlilik (Fold)	Duyarlılık	Özgüllük	Doğruluk
SMO	10	0.986	0.993	%98.85
J48	10	0.959	0.968	%96.42
MLP	10	0.978	0.944	%95.80
RF	10	0.953	0.968	%96.05
IBK	10	0.966	0.993	%97.70
ESA	10	0.993	1.00	%99.60



Şekil 21. Çalışmada Kullanılan Klasik Makine Öğrenmesi ve ESA Yöntemlerinin ROC Eğrisi

Receiver Operating Characteristic (ROC eğrileri), sınıflandırma performansının değerlendirilmesinde kullanılan analitik yöntemlerden biridir. Bu eğriler incelendiğinde x ekseninde 1-Özgüllük, y ekseninde ise duyarlılık yer almaktadır. ROC eğrileri değişkenler için karmaşıklık matrisinde bulunan doğru pozitif, doğru negatif, yanlış pozitif, yanlış negatif hakkında kestirimler yapılmasına sağlar (Hayran, 2011). Test için yapılan en iyi kesim noktasının saptanmasında, doğru ve yanlış kararların hesaplanmasında kullanılır. Bunun yanı sıra farklı testlerin doğru sınıflandırma başarılarının karşılaştırılmasını sağlar (Pagano, Gauvreau ve Mattie 1993). Şekil 21’de yer alan ROC eğrilerinde görüldüğü gibi, Evrişimli sinir ağı ve klasik makine öğrenmesi algoritmaları karşılaştırıldığında evrişimli sinir ağı daha yüksek eğriler çizerek diğer algoritmalar arasından daha başarılı performans göstermiştir.

5. SONUÇ

Teknolojide meydana gelen hızlı gelişmeler bireylerin kendilerini birçok alanda geliştirmelerini ve yeni beceriler kazanmalarını mecbur kılmıştır. Bu becerilerden bilişimsel düşünme becerisi oldukça önemlidir. Bu becerinin gelecek iş gücünü sağlayacak yeni nesillere kazandırılmasında kodlama eğitiminin önemli bir rolü vardır. Gelişmiş ve gelişmekte olan ülkeler kodlamayı eğitim sistemlerine entegre etmişlerdir. Kodlama eğitiminin erken yaşlarda verilmesiyle gelecekte bilişimsel düşünme becerisine sahip bireyler yetiştirmek ülkelerin temel hedeflerinden biri haline gelmiştir. Bu bakımdan kodlama eğitimi veren eğitimcilerin bu konuda gerekli olan yetkinliğe ileri düzeyde sahip olmaları beklenmektedir.

Ülkemizde kodlama eğitimi ortaokul ve lise düzeyinde programa dâhil edilmiştir. Ülkemizde kodlama eğitimi blok tabanlı kodlama araçlarından Scratch ile verilmektedir. Bu çalışma ile bilişim teknolojileri öğretmenlerinin kodlama araçlarından biri olan Scratch'e yönelik özyeterlik düzeylerinin sınıflandırılması amaçlanmıştır. Bu doğrultuda derin öğrenme yöntemlerinden Evrişimli Sinir Ağları (ESA) ve klasik makine öğrenmesi yöntemleri ile özyeterlik düzeyleri TSECT anketiyle tahmin edilmeye ve sınıflandırılmaya çalışılmıştır.

TSECT anketi ile toplanan veriler Weka ve Google Colab platformlarına aktararak çözümlenmiştir. Deneysel sonuçlar incelendiğinde en iyi sınıflandırma performansı elde edilen klasik makine öğrenmesi yöntemleri sırasıyla %98,6 doğrulukla SMO, %97,8 doğrulukla MLP, %96,6 doğrulukla IBK, %95,9 doğrulukla J48 ve %95,6 doğrulukla RF'dır. Ayrıca ESA modeli ile erişilen sınıflandırma başarı oranı %99,3'dır. ESA ile klasik makine öğrenmesi yöntemleri karşılaştırıldığında; ESA ile daha başarılı sınıflandırma performansına ulaşılmıştır.

Gelecekteki çalışmalarda farklı klasik makine öğrenmesi ve derin öğrenme yöntemleri kullanılarak işlemler tekrarlanabilir. ESA modelindeki sınıflandırma performansını artırmak için farklı parametreler ya da optimizasyon yöntemleri uygulanabilir. Örnek sayısını arttırarak aynı yöntemlerle daha başarılı sonuçlara erişilebileceği düşünülmektedir.

Çalışmanın veri toplama sürecinin pandemi dönemine denk gelmesi ulaşılan örneklem büyüklüğünün sınırlanmasına neden olmuştur. Benzer bir çalışma örneklem sayısı arttırılarak tekrarlanabilir.



6. KAYNAKÇA

- Akar, Ö., ve Güngör, O. (2012). Classification of multispectral images using Random Forest algorithm. *Journal of Geodesy and Geoinformation*, 1(2), 105-112.
- Akçay, A., ve Çoklar, A. N. (2016). Bilişsel becerilerin gelişimine yönelik bir öneri: Programlama eğitimi. *Eğitim teknolojileri okumaları*, 121-139.
- Aksu, N. (2019). *Farklı ülkelerden Pisa sınavına katılan öğrencilerin matematik okuryazarlığını etkileyen faktörlerin tahmin edilmesi* (Yayınlanmamış Yüksek Lisans Tezi, Adnan Menderes Üniversitesi Sosyal Bilimler Enstitüsü, Aydın).
- Albannai, B. (2019). *Comparative Study of HVAC and HVDC Transmission Systems With Proposed Machine Learning Algorithms for Fault Location Detection* (Doctoral dissertation, Arizona State University).
- Alp, G. (2019). *Scratch programı ile web destekli işbirlikli öğrenme yönteminin ilköğretim 5. sınıf öğrencilerinin kavramsal anlama düzeylerine ve eleştirel düşünme becerilerine etkisi* (Yayınlanmamış Yüksek Lisans Tezi, Uludağ Üniversitesi, Bursa).
- Alpaydin, E. (2013). *Introduction to Machine Learning*, Massachusetts Institute of Technology.
- Arslan, K. (2020). Eğitimde yapay zekâ ve uygulamaları. *Batı Anadolu Eğitim Bilimleri Dergisi*, 11(1), 71-88.
- Atalay, M., ve Çelik, E. (2017). Büyük veri analizinde yapay zekâ ve makine öğrenmesi uygulamaları-artificial intelligence and machine learning applications in big data analysis. *Mehmet Akif Ersoy Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, 9(22), 155-172.

- Aylak, B.L, Oral O., ve Yazıcı, K. (2021). Yapay zekâ ve makine öğrenmesi tekniklerinin lojistik sektöründe kullanımı. *El-Cezrî Fen ve Mühendislik Dergisi*, 8(1), S.74-93. DOI:10.31202/ecjse.776314
- Aytekin, A., Çakır, F. S., Yücel, Y. B., ve Kulaöz, İ. (2018). Geleceğe yön veren kodlama bilimi ve kodlama öğrenmede kullanılabilecek bazı yöntemler. *Avrasya Sosyal ve Ekonomi Araştırmaları Dergisi*, 5(5), 24-41.
- Bandura, A. (1977). Self-efficacy: Toward a unifying theory of behavioural change, *Psychological Review*, 84(2), 191–215.
- Basheer, I. A., & Hajmeer, M. (2000). Artificial neural networks: fundamentals, computing, design, and application. *Journal of microbiological methods*, 43(1), 3-31.
- Bayman, P., & Mayer, R. E. (1983). A diagnosis of beginning programmers' misconceptions of BASIC programming statements. *Communications of the ACM*, 26(9), 677-679.
- Bean, N., Weese, J., Feldhausen, R., & Bell, R. S. (2015, October). Starting from scratch: Developing a pre-service teacher training program in computational thinking. In *2015 IEEE Frontiers in Education Conference (FIE)* (pp. 1-8). IEEE.
- Bengio, Y. (2009). Learning deep architectures for AI. *Foundations and trends in Machine Learning*, 2(1), 1-127.
- Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education*, 72, 145-157.
- Bocconi, S., Chiocciariello, A., Dettori, G., Ferrari, A., Engelhardt, K., Kampylis, P., & Punie, Y. (2016). Developing computational thinking in compulsory education. *European Commission, JRC Science for Policy Report*, 68.
- Breiman, L. (2001). Random forests. *Machine learning*, 45(1), 5-32.
- Burns, J. M. (2012). Leadership. Open Road Media.
- Cankıran, C. (2019). “Makine Öğrenimi”, <https://www.cancankiran.com/makine-ogrenimi/> ,(Erişim tarihi: 5 Mart 2022).

- Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16, 321-357.
- Cleary, J.G., Trigg, L.E. (1995). K*: An instance-based learner using an entropic distance measure. *Proceedings of the 12th International Conference on Machine Learning*, Tahoe City, California, USA, pp.108-114.
- Çatlak, Ş., Tekdal, M., & Baz, F. (2015). Scratch yazılımı ile programlama öğretiminin durumu: Bir doküman inceleme çalışması. *Journal of Instructional Technologies and Teacher Education*, 4(3).
- Çeşmeli, M.Ş., Bozkurt, Ö. Ç., Kalkan, A. ve Pençe, İ, (2015). Yönetim Bilişim Sistemleri Bölümü Öğrencilerinin Yönetim Ve Bilişim Derslerindeki Başarılarının Veri Madenciliği Yöntemleri İle İncelenmesi. *Yönetim Bilişim Sistemleri Dergisi*, 1(2), 36-47
- Çetin, E. (2012). Bilgisayar programlama eğitiminin çocukların problem çözme becerileri üzerine etkisi. (Yayımlanmamış yüksek lisans tezi), Gazi Üniversitesi, Eğitim Bilimleri Enstitüsü, Ankara
- Çiftçi, F., Kaleli, C., & Ünal, S. (2018). Öznitelik seçme ve makine öğrenmesi yöntemleriyle öğretmen performansının tahmin edilmesi. *Anadolu Journal of Educational Sciences International*, 8(2), 419-440.
- Çölkesen, İ., & Kavzoğlu, T. (2010). Farklı boyutta eğitim örnekleri için destek vektör makinelerinin sınıflandırma performansının analizi. *Proceedings of III. Uzaktan Algılama ve Coğrafi Bilgi Sistemleri Sempozyumu*, 161-170.
- Çölkesen. (2010). *Uzaktan algılamada ileri sınıflandırma tekniklerinin karşılaştırılması ve analizi*. (Yayımlanmamış Yüksek Lisans Tezi) , Gebze Yüksek Teknoloji Enstitüsü Mühendislik ve Fen bilimleri Enstitüsü, Kocaeli.
- Demirer, V., & Sak, N. (2016). Programming education and new approaches around the world and in Turkey. *Eğitimde Kuram ve Uygulama* , 12(3), 521-546.
- Deng, L., & Yu, D. (2014). Deep learning: methods and applications. *Foundations and trends in signal processing*, 7(3-4), 197-387.

- Durna, M.B. (2020). “Cross-Validation nedir? Nasıl çalışır?”, <https://medium.com/bili%C5%9Fim-hareketi/cross-validation-nedir-nas%C4%B1l-%C3%A7al%C4%B1%C5%9F%C4%B1r-4ec4736e5142> , (Erişim tarihi: 10 Mayıs 2022).
- Elmas, Ç. 2007. Yapay Zeka Uygulamaları. Ankara: Seçkin Yayıncılık.
- Eraldemir, S. G., Arslan, M. T., & Yıldırım, E. (2017, November). The Effect of Feature Selection Algorithms in EEG Signal Analysis. In *International Advanced Researches & Engineering Congress*.
- Ergin, T. (2018). “Convolutional Neural Network (ConvNet yada CNN) nedir, nasıl çalışır?”, <https://medium.com/@tuncerergin/convolutional-neural-network-convnet-yada-cnn-nedir-nasil-calisir-97a0f5d34cad>, (Erişim tarihi: 10 Mayıs 2022).
- Ersoy, A., & Özden, M. (2011). Öğretmen adaylarının ödevlerinde internetten intihal yapmalarında öğretim elemanının rolüne ilişkin görüşleri. *İlköğretim Online*, 10(2), 608-619.
- Ersöz, G. (2017). The Role of University Students' General Self-Efficacy, Depression and Psychological Well-Being in Predicting Their Exercise Behavior. *Journal of Education and Training Studies*, 5(3), 110-117.
- Fawcett, T., & Flach, P. A. (2005). A response to Webb and Ting’s on the application of ROC analysis to predict classification performance under varying class distributions. *Machine Learning*, 58(1), 33-38.
- Feng, D., Haase-Schütz, C., Rosenbaum, L., Hertlein, H., Glaeser, C., Timm, F., ... & Dietmayer, K. (2020). Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Transactions on Intelligent Transportation Systems*, 22(3), 1341-1360.
- Ferhan, M. (2018). *PISA 2012 matematik ilgi ölçeğinin klasik test kuramı ve madde tepki kuramına göre psikometrik özellikleri* (Yayınlanmamış Yüksek Lisans Tezi) , Hasan Kalyoncu Üniversitesi, İstanbul.
- Gholap, J. (2012). Performance tuning of J48 Algorithm for prediction of soil fertility. *arXiv preprint arXiv:1208.3943*.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. London: MIT press.

- Goodfellow, I., McDaniel, P., & Papernot, N. (2018). Making machine learning robust against adversarial inputs. *Communications of the ACM*, 61(7), 56-66.
- Göncü, A., Çetin, İ., & Şendurur, P. (2020). Bilişim Teknolojileri ve Yazılım Dersi Öğretmenlerinin Kodlama Eğitimine Yönelik Görüşleri. *Mersin Üniversitesi Eğitim Fakültesi Dergisi*, 16(2), 301-321.
- Göncü, A., Çetin, İ., & Top, E. (2018). Öğretmen adaylarının kodlama eğitimine yönelik görüşleri: bir durum çalışması. *Mehmet Akif Ersoy Üniversitesi Eğitim Fakültesi Dergisi*, (48), 85-110.
- Gülbahar, Y., Kert, S. B., & Kalelioğlu, F. (2019). Bilgi işlemsel düşünme becerisine yönelik öz yeterlik algısı ölçeği: Geçerlik ve güvenirlik çalışması. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, 10(1), 1-29.
- Güler, İ., & Übeyli, E. D. (2005). An expert system for detection of electrocardiographic changes in patients with partial epilepsy using wavelet-based neural networks. *Expert Systems*, 22(2), 62-71.
- Hayran, M. (2011). *Sağlık Araştırmaları İçin Temel İstatistik* (1. Basım). Ankara :Art Ofset Matbaacılık Yayıncılık Organizasyon.
- He, H., & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9), 1263-1284.
- Jamjoom, A. A., Rhodes, J., Andrews, P. J., & Grant, S. G. (2021). The synapse in traumatic brain injury. *Brain*, 144(1), 18-31.
- Kalelioğlu, F., Gülbahar, Y., Akçay, S., & Doğan, D. (2014). Curriculum integration ideas for improving the computational thinking skills of learners through programming via scratch. In *Local proceedings of the 7th international conference on informatics in schools: Situation, evolution and perspectives* (pp. 101-112).
- Karaduman, T. (2019). Yapay zekâ uygulama alanları. *Gazi Üniversitesi, Bilisim Enstitüsü, Adli Bilisim. ABD*, 1-11.
- Karakuş, B. A. (2018), “Derin Sinir Ağları için Aktivasyon Fonksiyonları”, <http://buyukveri.firat.edu.tr/2018/04/17/derin-sinir-aglari-icin-aktivasyon-fonksiyonlari/> , (Erişim Tarihi: 25 Nisan 2022)

- Kartal, E., & Özen, Z. (2017). Dengesiz veri setlerinde sınıflandırma. O. Torkul, S. Gülseçen, Y. Uyaroğlu, G. Çağıl, and MK Uçar (Eds.), *Mühendislikte Yapay Zekâ ve Uygulamaları içinde, 1st ed.*, (ss. 109-131), Sakarya: Sakarya Üniversitesi Kütüphanesi Yayınevi .
- Kaur, G., & Chhabra, A. (2014). Improved J48 classification algorithm for the prediction of diabetes. *International journal of computer applications*, 98(22).
- Kaynar, O., & Taştan, S. (2009). Zaman serisi analizinde MLP yapay sinir ağları ve arıma modelinin karşılaştırılması. *Erciyes Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, 33, 161-172.
- Kaynar, O., Arslan, H., Görmez, Y., & IŞIK, Y. E. (2018). Makine öğrenmesi ve öznelik seçim yöntemleriyle saldırı tespiti. *Bilişim Teknolojileri Dergisi*, 11(2), 175-185.
- Kızrak, A. (2018). “DERİNE DAHA DERİNE: Evrişimli Sinir Ağları”, <https://ayyucekizrak.medium.com/deri%CC%87ne-daha-deri%CC%87ne-evri%C5%9Fimli-sinir-a%C4%9Flar%C4%B1-2813a2c8b2a9>, (Erişim tarihi: 10 Nisan 2022).
- Kirasich, K., Smith, T., & Sadler, B. (2018). Random forest vs logistic regression: binary classification for heterogeneous datasets. *SMU Data Science Review*, 1(3), 9.
- Korkem, E. (2013). *Mikroarray Gen Ekspresyon Veri Setlerinde Random Forest ve Naive Bayes Sınıflama Yöntemleri Yaklaşım* (Yayınlanmamış Yüksek Lisans Tezi) Hacettepe Üniversitesi, Ankara.
- Korting, T. S. (2006). C4. 5 algorithm and multivariate decision trees, image processing division. *National Institute for Space Research–INPE, SP, Brazil*.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.
- Kuan, T. W., Wang, J. F., Wang, J. C., Lin, P. C., & Gu, G. H. (2011). VLSI design of an SVM learning core on sequential minimal optimization algorithm. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 20(4), 673-683.

- Madhusudana, C. K., Kumar, H., & Narendranath, S. (2016). Condition monitoring of face milling tool using K-star algorithm and histogram features of vibration signal. *Engineering science and technology, an international journal*, 19(3), 1543-1551.
- Marsland, S. (2011). *Machine learning: an algorithmic perspective*. Chapman and Hall/CRC.
- McCarthy, J. (2004). *What is artificial intelligence?*. Erişim adresi (17 Temmuz 2022): <http://www-formal.stanford.edu/jmc/whatisai/>.
- McCarthy, J., Minsky, M. L., Rochester, N., & Shannon, C. E. (2006). A proposal for the dartmouth summer research project on artificial intelligence, August 31, 1955. *AI magazine*, 27(4), 12-12.
- Narin, A., Kaya, C., & Pamuk, Z. (2021). Automatic detection of coronavirus disease (COVID-19) using X-ray images and deep convolutional neural networks. *Pattern Analysis&Applications:PAA*, 1–14.
- Nasution, D. K. (2018). Penerapan Teknik Molina & Albir dalam Penerjemahan Teks Mantera Jamuan Laut dari Bahasa Melayu ke dalam Bahasa Inggris. *Kumpulan Penelitian dan Pengabdian Dosen*, 1(1).
- Nizam, H., & Akın, S. S. (2014). Sosyal medyada makine öğrenmesi ile duygu analizinde dengeli ve dengesiz veri setlerinin performanslarının karşılaştırılması. *XIX. Türkiye'de İnternet Konferansı*, 1(6).
- Nur, I. M., & Ülker, E. (2020). A Novel Hybrid IoT Based IDS Using Binary Grey Wolf Optimizer (BGWO) and Naive Bayes (NB). *Avrupa Bilim ve Teknoloji Dergisi*, 279-286.
- Ozoran, D., Cagiltay, N., & Topalli, D. (2012, January). Using scratch in introduction to programming course for engineering students. In *2nd International Engineering Education Conference (IEEC2012)* (Vol. 2, pp. 125-132).
- Özmen, B., & Altun, A. (2014). Undergraduate students' experiences in programming: difficulties and obstacles. *Turkish Online Journal of Qualitative Inquiry*, 5(3), 1-27.
- Pagano, M., Gauvreau, K., & Mattie, H. (2022). *Principles of biostatistics*. Boston: CRC Press.

- Painuli, S., Elangovan, M., & Sugumaran, V. (2014). Tool condition monitoring using K-star algorithm. *Expert Systems with Applications*, 41(6), 2638-2643.
- Parsania, V., Bhalodiya, N., & Jani, N. N. (2014). Applying Naïve bayes, BayesNet, PART, JRip and OneR algorithms on hypothyroid database for comparative analysis. *International Journal Of Darshan Institute On Engineering Research & Emerging Technologies*, 3(1), 60-64.
- Platt, J. (1998). Sequential minimal optimization: A fast algorithm for training support vector machines. *Microsoft Research Technical Report MSR-TR-98-14*, 1-21.
- Quinlan, J. R. (1992, November). Learning with continuous classes. In *5th Australian joint conference on artificial intelligence* (Vol. 92, pp. 343-348).
- Sanket, G. (2018). “What is Artificial Intelligence (AI) ?”, <https://becominghuman.ai/what-is-artificial-intelligence-ai-4bde325e5462> ,(Erişim tarihi: 15 Mart 2022).
- Sargın, A. & Göçen, A. (2020). *Çocuklar İçin Yapay Zeka*. Şanlıurfa: Urfa STEM ve Bilim Merkezi.
- Sarıkoç, F. (2012). *Meme Kanseri Biyopsi Örneklerinde Östrojen Hormonu Alıcı Varlığının Zeki Yöntemlerle Tespiti*. (Yayınlanmamış Doktora Tezi), Erciyes Üniversitesi, Fen Bilimleri Enstitüsü, Kayseri.
- Saygıner, Ş. (2017). *Blok tabanlı görsel ve metin tabanlı programlama öğretimlerinin erişimi, mantıksal düşünme ve motivasyona etkileri*. (Yayınlanmamış Yüksek Lisans Tezi), Hacettepe Üniversitesi, Ankara
- Sayın, Z., & Seferoğlu, S. S. (2016). Yeni bir 21. yüzyıl becerisi olarak kodlama eğitimi ve kodlamanın eğitim politikalarına etkisi. *Akademik Bilişim Konferansı*, 3(5).
- Sayracı, B. (2021). “Derin Öğrenme ve Evrimsel Sinir Ağları”, <https://medium.com/deep-learning-from-deepest/derin-öğrenme-ve-evrimsel-sinir-ağları-446dc8a8d2f> ,(Erişim tarihi: 5 Mart 2022).
- Sırakaya, M. (2018). Ön lisans öğrencilerinin nomofobi düzeylerinin akıllı telefon kullanım durumlarına göre incelenmesi. *Mersin Üniversitesi Eğitim Fakültesi Dergisi*, 14(2), 714-727.

- Simard P.Y., Steinkraus D. & Platt J.C. Best practices for convolutional neural networks applied to visual document analysis. In: Proceedings of the seventh international conference on document analysis and recognition; Edinburgh, UK; 2003; 2: 958–62.
- Sing, K. S. (1991). Characterization of porous solids: An introductory survey. *In Studies in Surface Science and Catalysis* (Vol. 62, pp. 1-9). Elsevier.
- Solmaz, R., Günay, M., & Alkan, A. (2014). Fonksiyonel Tiroit Hastalığı Tanısında Naive Bayes Sınıflandırıcının Kullanılması. *Akademik Bilişim Konferansı, Mersin*, 891-897.
- Sterling, L. (2015). An education for the 21st century means teaching coding in schools. *The Conversation*.
- Süt, N., & Çelik, Y. (2012). Prediction of mortality in stroke patients using multilayer perceptron neural networks. *Turkish Journal of Medical Sciences*, 42(5), 886-893.
- Şimşek, H. K. (2018). “Makine Öğrenmesi Dersleri 8: Cross Validation”, <https://medium.com/data-science-tr/overfitting-underfitting-cross-validation-b47dfda0cf4e> ,(Erişim tarihi: 5 Şubat 2022).
- Tankuş, S. (2019). *Üniversite sınavına giren öğrencilerin sınav sonuçlarının yapay zeka ile analizi ve değerlendirmesi: 2018 yılı Şanlıurfa ili örneği*. (Yayınlanmamış Doktora Tezi), Harran Üniversitesi, Şanlıurfa.
- Taşkın, Ç. Ş., & Hacıömeroğlu, G. (2010). Öğretmen özyeterlik inanç ölçeğinin Türkçeye uyarlanması ve sınıf öğretmeni adaylarının özyeterlik inançları. *Dokuz Eylül Üniversitesi Buca Eğitim Fakültesi Dergisi*, (27), 63-75.
- Tekerek, M., Ercan, O., Udum, M. S., & Saman, K. (2012). Bilişim teknolojileri öğretmen adaylarının bilgisayar öz-yeterlikleri. *Turkish Journal of Education*, 1(2), 80-91.
- Tezbaşaran, A., 2008. Likert tipi ölçek hazırlama kılavuzu, Türk Psikologlar Derneği Yayınları, Ankara, Türkiye.
- Truba (2021), “Evrişimli sinir ağları (Convolutional neural networks)”, <https://docs.truba.gov.tr/education/pytorch/cnn.html> ,(Erişim tarihi: 15 Mart 2022).
- Tucker, M. R., Araujo, A. C. G., Paech, N. A., Hecht, V., Schmidt, E. D., Rossell, J. B., ... & Koltunow, A. M. (2003). Sexual and apomictic reproduction in Hieracium

subgenus *Pilosella* are closely interrelated developmental pathways. *The Plant Cell*, 15(7), 1524-1537.

Turgut, M. F. ve Baykul, Y. (1992). *Ölçekleme Teknikleri*. Ankara: ÖSYM Yayınları.

Turkhost (2021). “Makine Öğrenmesi (Machine Learning) Nedir?”, <https://blog.turhost.com/makine-ogrenmesi-machine-learning-nedir/>, (Erişim tarihi: 10 Mart 2022).

Türk, B., & Tali, H. H. (2020). Makine Öğrenmesi Yöntemleri ile Somatizasyon ve Duygusal Öz Farkındalık Üzerinden Şiddetin Tahmini. *Adli Tıp Bülteni*, 25(2), 99-105.

Türker, P. M., & Pala, F. K. (2018). Ortaokul öğrencilerinin, öğretmenlerin ve öğrenci velilerinin kodlamaya yönelik görüşleri. *İlköğretim Online*, 17(4), 2013-2029.

Uzun, Y. (2005). Tıbbi veriler üzerinde makine öğrenme algoritmaları ve bulanık mantık ile kurallar öğrenme. (Yayımlanmamış yüksek lisans tezi), Selçuk Üniversitesi, Konya .

Ülgen, E. K. (2020), “*Evrişimli Sinir Ağlarıyla Haberlerin Sınıflandırılması*”, <https://www.veribilimiokulu.com/evrisimli-sinir-aglariyla-haberlerin-siniflandirilmasi/>, (Erişim tarihi: 15 Nisan 2022)

Ünal, Z. (2019). *Likert tipi verilerde bulanık mantık ve derin öğrenme entegrasyonu*. (Yayımlanmamış doktora tezi). Akdeniz Üniversitesi, Antalya

Ünsal İ, 2020, *Blok Tabanlı Programlama Etkinliklerinin İlkokul 2. Sınıf Öğrencilerinin Bilgi İşlemsel Düşünme Becerisi ve Etkinlik Algısı Üzerine Etkisi*. (Yayımlanmamış Yüksek Lisans Tezi), Sakarya Üniversitesi, Sakarya.

Visa, S., Ramsay, B., Ralescu, A. L., & Van Der Knaap, E. (2011). Confusion matrix-based feature selection. *MAICS*, 710(1), 120-127.

Wirojcharoenwong, W., Luangnaruedom, N., Rattanasiriwongwut, M., & Tiantong, M. (2014). Decision Tree Classifier for Computer Self-Efficacy with Best First Feature Selection. *Int. J. Comput. Internet Manag*, 22, 62-67.

Yavaş, M., Güran, A., & Uysal, M. (2020). Covid-19 Veri kümesinin Smote tabanlı örnekleme yöntemi uygulanarak sınıflandırılması. *Avrupa Bilim ve Teknoloji Dergisi*, 258-264.

- Yavuz, E. (2020). *Konutlarda enerji tüketimi kestirimi için derin öğrenme ve makine öğrenme yöntemlerinin karşılaştırılması* (Yayınlanmamış Yüksek Lisans), Tezi, İstanbul Üniversitesi, İstanbul.
- Yıldız, M., & Börekci, C. (2020). Predicting Academic Achievement with Machine Learning Algorithms. *Journal of Educational Technology and Online Learning*, 3(3), 372-392.
- Yükseltürk, E., & Altıok, S. (2016). Bilişim teknolojileri öğretmen adaylarının programlama öğretiminde Scratch aracının kullanımına ilişkin algıları. *Mersin Üniversitesi Eğitim Fakültesi Dergisi*, 12(1).



EKLER



EK 1. TSECT Ölçeği (Orijinal)

TEACHERS' SELF-EFFICACY IN COMPUTATIONAL THINKING(TSECT)

TABLE I. ITEM WORDING FOR FIRST TSECT INSTRUMENT

Item	Wording
1	I feel confident using computer technology.*
2	I feel confident writing simple programs for the computer.
3	I know how to teach programming concepts effectively.
4	I can promote a positive attitude towards programming in my students.
5	I can guide students in using programming as a tool while we explore other topics.
6	I feel confident using programming as an instructional tool within my classroom.
7	I can adapt lesson plans incorporating programming as an instructional tool.
8	I can create original lesson plans incorporating programming as an instructional tool.
9	I can identify how programming concepts relate to Common Core Standards
10	I can identify how programming concepts relate to Next Generation Science Standards.
Note: All items used a five-point Likert scale with options of: Strongly Agree, Agree, Neither Agree nor Disagree, Disagree, Strongly Disagree.	

* Item excluded from TSECT scale measure

TABLE II. ITEM WORDING FOR THE SECOND TSECT INSTRUMENT

Item	Wording
<i>Items Aligned with CT Concepts</i>	
I can teach students to write programs that ...	
1	execute a step-by-step sequence of commands
2	uses loops to repeat commands
3	respond to events like pressing a key on the keyboard
4	do more than one thing at the same time

EK 1. (devamı)

5	only execute some commands when a specific condition is met
6	perform arithmetic operations like addition and subtraction
7	can store, update, and retrieve values
8	can ask the user a question
<i>Items Adopted from earlier TSECT Instrument</i>	
With my future classes, I can ...	
9	promote a positive attitude towards programming
10	guide students in using programming as a tool as we explore other topics
11	use programming as an instructional tool
12	adapt lesson plans incorporating programming to meet my students' learning level
13	Create original lesson plans using programming as an instructional tool
14	Recognize how programming relates to the Common Core State Standards
15	Recognize how programming relates to the Next Generation Science Standards
Note: All items used a 100-point scale where 0 = cannot do 50 = moderately I can do 100 = highly certain I can do	

TABLE III. ITEM WORDING FOR THE SECT INSTRUMENT

Item	Wording
<i>Items Aligned with Intervention Activities</i>	
I can write a computer program to ...	
1	draw shapes
2	animate characters
3	tell a story
4	play music

EK 1. (devamı)

5	model plant growth
<i>Items Aligned with CT Concepts</i>	
I can create a computer program which ...	
6	executes a step-by-step sequence of commands
7	uses loops to repeat commands
8	responds to events like pressing a key on the keyboard
9	does more than one thing at the same time
10	only executes some commands when a specific condition is met
11	perform arithmetic operations like addition and subtraction
12	can store, update, and retrieve values
13	can ask the user a question
<i>Items Aligned with CT Practices</i>	
When creating a computer program I ...	
14	make improvements one step at a time, and incorporate new ideas as I have them
15	run my program frequently to make sure it does what I want, and fix any problems I find
16	share my programs with others and look at others' programs for ideas
17	break my program into multiple parts to carry out different actions
Note: All items used a 100-point scale where 0 = cannot do 50 = moderately I can do 100 = highly certain I can do	

EK 2. TSECT Ölçeği (Türkçe Çeviri)

Bilişimsel Düşünmede Öğretmenlerin Öz-yeterliği (TSECT)

Madde		Kesinlikle Katılmıyorum	Katılmıyorum	Ne katılıyorum ne katılmıyorum	Katılıyorum	Kesinlikle Katılıyorum
1	Bilgisayar programları yazımında kendime güvenirim.					
2	Programlama kavramlarını etkili bir biçimde öğretmeyi bilirim.					
3	Öğrencilerimin programlamaya karşı olumlu tutumunu destekleyebilirim					
4	Farklı konuları işlerken öğrencilerimin programlamayı bir araç olarak kullanmalarını sağlamak için rehberlik edebilirim.					
5	Sınıfımda programlamayı eğitim aracı olarak kullanma konusunda kendime güvenirim.					
6	Programlamayı bir öğretim aracı olarak kapsayan ders planlarını uyarlayabilirim.					
7	Programlamayı bir öğretim aracı olarak kapsayan orjinal ders planları oluşturabilirim.					
8	Programlama kavramlarının Yeni Nesil Bilim Standartları(?) ile nasıl ilişkili olduğunu açıklayabilirim.					

Öğrencilere ... için program yazmayı öğretebilirim		Yapamam	Kısmen Yapabilirim	Kesinlikle Yapabilirim
1	komut dizisini adım-adım çalıştırılmak			
2	komutları yinelemek için döngü kullanmak			
3	klavye üzerinde bir tuşa basarak bir olaya müdahale etmek			
4	bir anda birden fazla iş yapmak			
5	özel koşulları sağlayacak komutları çalıştırmak			
6	toplama ve çıkartma benzeri işlemleri yapmak			
7	değerleri yüklemek, düzeltmek ve bulmak			

8	kullanıcıya bir soru sormak			
---	-----------------------------	--	--	--

EK 2. (devamı)

Gelecek dönemlerde sınıflarımda,.... Yapabilirim				
9	programalmaya yönelik olumlu tutum geliştirmeye destek			
10	farklı konularda araştırırken programlamayı bir araç olarak kullanmalarına rehberlik			
11	bir eğitim aracı olarak programlama kullanımı			
12	ders programlarımı öğrencilerimin öğrenme seviyesine göre ayarlama			
13	bir eğitim aracı olarak programlama kullanımı için yeniden ders programı oluşturma			
14	Programlamanın Yeni Nesil Bilim Standartları ilişkisi hakkında farkındalığımı artırma			

... amacıyla bilgisayar programı (kodlama yapabilirim) yazabilirim		Yapamam	Kısmen Yapabilirim	Kesinlikle Yapabilirim
1	şekil çizimi			
2	animasyon (canlandırma filmi yapımı)			
3	bir hikaye anlatmak			
4	müzik yapmak			
5	modelleme yapmak			
... amacıyla bilgisayar programı oluşturabilirim.				
6	adım adım komutları çalıştırmak			
7	komutları yinlemek için döngü kullanmak			
8	klavye üzerinde bir tuşa basarak bir olaya müdahale etmek			
9	bir anda birden fazla iş yapmak			
10	özel koşulları sağlayacak komutları çalıştırmak			
11	toplama ve çıkartma benzeri işlemleri yapmak			
12	değerleri yüklemek, düzeltmek ve bulmak			

13	kullanıcıya bir soru sormak			
----	-----------------------------	--	--	--

EK 2. (devamı)

Bir bilgisayar programı oluştururken ...				
14	geliştirmeleri adım adım yaparak, yeni fikirler geldikçe eklerim			
15	istediğim veya bulduğum sorunları düzeltmeyi yaptığıma emin olmak için sık sık programımı çalıştırırım			
16	programlarımı başkaları ile paylaşarak onların fikirlerini görmek isterim			
17	değişik işlevleri yapmak için programımı birden fazla bölüme ayırırım			

EK 3. Ölçek İzin Maili

**Burak KOCA**
Kime: Nathan Bean

14.05.2021 Cum 20:05

From: Burak KOCA
Sent: Sunday, May 9, 2021 3:51 PM
To: Nathan Bean
Subject: Permission for TSECT scale

This email originated from outside of K-State.

Dear Bean,

I am a master student of Aksaray University on management information systems. I am conducting a thesis study on Computational Thinking. In this study, I want to determine the self-efficacy of I.T. teachers towards SCRATCH. If you have your permission, I would like to use your TSECT scale that you have developed. I want to use it in my thesis study by adapting it to the Turkish language.

Best regards.

Kimden: Nathan Bean
Gönderildi: 14 Mayıs 2021 Cuma 21:19
Kime: Burak KOCA
Konu: RE: Permission for TSECT scale

Burak,

You are more than welcome to adapt the TSECT instrument to Turkish. Shoot me any papers on your results – I'd love to see them.

Sincerely,

Nathan Bean

From: Burak KOCA
Sent: Sunday, May 9, 2021 3:51 PM
To: Nathan Bean
Subject: Permission for TSECT scale

EK 4. Etik Kurul Kararı



T.C.
AKSARAY ÜNİVERSİTESİ REKTÖRLÜĞÜ
İnsan Araştırmaları Etik Kurulu



Konu : Başvurumuz Hk.

Sayın: Burak KOCA

“Bilişim Teknolojileri Öğretmenlerinin Scratch Yazılımına İlişkin Öz Yeterlilik İnançlarının Makine Öğrenmesi ve Derin Öğrenme Yöntemleri İle Tahmin Edilmesi”
başlıklı 2021/05-14 protokol numaralı başvuru 21.06.2021 tarihli toplantıda kurulumuz tarafından incelenmiş, Üniversitemiz İnsan Araştırmaları Etik Kurulu Yönergesi’nde belirtilen etik ilkelere uygun olduğuna toplantıya katılan üyelerin oy birliği ile karar verilmiştir.

Bilgilerinize rica ederim.

Prof. Dr. Necmettin AYGÜN
Aksaray Üniversitesi İnsan Araştırmaları
Etik Kurul Başkanı

Burakın elektronik imzalı suretine <https://e-belge.aksaray.edu.tr> adresinden a103ccc8-ee2e-4c4d-9d09-731bd785e5ff kodu ile erişebilirsiniz.
Bu Belge 5070 sayılı Elektronik İmza Kanunu’nun 5. Maddesi gereğince güvenli elektronik imza ile imzalanmıştır.
Bu belge güvenli elektronik imza ile imzalanmıştır.

Belge Doğrulama Kodu: A103CCC8-EE2E-4C4D-9D09-731BD785E5FF

Belge Doğrulama Adresi: <https://e-belge.aksaray.edu.tr>

Adres: tarih Bölümü

Telefon No: 2882194 Faks No: 2882125

e-Posta:

KEP Adresi:

İnternet Adresi: <https://www.aksaray.edu.tr/>

Ayrıntılı bilgi için: Necmettin AYGÜN

Profesör

Telefon No: 2882194



TEZ DEĞERLENDİRME FORMU

Aksaray Üniversitesi Tez Değerlendirme Formu		
Öğrencinin Adı Soyadı:		EVET
Kapak		X
1	Tez Başlığı tutanaktaki başlıkla aynı mı?	X
2	Kapakdaki ay ve yıl savunmaya girilen tarihle tutarlı mı?	X
3	Kapak format kılavuzdaki kapak formatına uygun mu?	X
4	Kapakta yazılan tüm yazılar doğru olarak verilmiş mi?	X
İçindekiler		
5	Sayfa numaraları tam verilmiş mi?	X
6	Şekil, Çizelge vb. listeleri verilmiş mi? Sıralaması doğru mu?	X
7	Özet, Abstract, Giriş, Sonuçlar vb. bölümler var mı?	X
8	Yazım hataları kontrolü yapıldı mı?	X
Giriş		
9	Hazırlanan tezin önemini anlatıyor mu?	X
10	İkinci ve Üçüncü dereceden başlık içermemeli kuralına uyuldu mu?	X
Özet/Abstract		
11	Kılavuza uygun mu?	X
12	Ay ve yıl savunmaya girilen tarihle tutarlı mı?	X
13	Özet; tek sayfa, tek aralık, tek paragraf kuralına uygun olarak yazıldı mı?	X
14	Bilim kodu, sayfa adedi, anahtar kelimeler ve tez danışmanı yazıldı mı?	X
Kaynakça		
15	Kaynakların tamamına metin içinde atıf yapıldı mı?	X
16	Kaynak formatı Kılavuzdaki kaynak formatına uygun olarak hazırlanmış mı?	X
17	Atıf formatı kılavuzdaki atıf formatına uygun mu?	X
Atıf Yöntemi APA 6 ☒ CMS ☐ İSNAD ☐		
Genel Değerlendirme		
18	Etik Beyan açıklaması okundu, uyuldu ve imzalandı mı?	X
19	Kabul/Onay sayfası kılavuzdaki formata uygun olarak düzenlenmiş mi?	X
20	Kabul /Onay sayfasında belirtilen oy birliği/oy çokluğu seçeneklerinden uygun olanı savunmayla tutarlı olacak şekilde belirlenmiş mi?	X
21	Sayfa kenar boşluklar ve sayfa numaraları kılavuzdaki formatına uygun mu?	X
22	Paragraf boşlukları ve metin satır aralığı kılavuza uygun olacak şekilde düzenlenmiş mi?	X
23	Başlık yazımları kılavuzdaki başlık formatlarına uygun mu?	X
24	Yazı tipi ve boyutu kılavuzdaki yazı tipi ve boyutu formatına uygun mu?	X
25	Şekil, Çizelge vb. açıklama ve numaralandırmaları kılavuzdaki formata uygun olarak yazılmış mı?	X