

T.C.
İSTANBUL BEYKENT ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ BİLİM DALI

**MAKİNE ÖĞRENMESİ YÖNTEMİYLE MÜŞTERİ
DAVRANIŞININ TAHMİN EDİLMESİ**

Yüksek Lisans Tezi

Tezi Hazırlayan

Merve ANIL

İstanbul, 2023

T.C.
İSTANBUL BEYKENT ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ BİLİM DALI

**MAKİNE ÖĞRENMESİ YÖNTEMİYLE MÜŞTERİ
DAVRANIŞININ TAHMİN EDİLMESİ**

Yüksek Lisans Tezi

Tezi Hazırlayan

Merve ANIL

Öğrenci No:2220003039

ORCID ID: 0000-0002-7539-5006

Danışman: Dr. Öğr. Üyesi Talat FİRLAR

İstanbul, 2023

YEMİN METNİ

Yüksek Lisans Tezi olarak sunduğum “Makine Öğrenmesi Yöntemiyle Müşteri Davranışının Tahmin Edilmesi” başlıklı bu çalışmanın, bilimsel ahlak ve geleneklere uygun şekilde tarafımdan yazıldığını, yararlandığım eserlerin tamamının kaynaklarda gösterildiğini ve çalışmamın içinde kullanıldıkları her yerde bunlara atıf yapıldığını belirtir ve bunu onurumla doğrularım. 02.10.2023

Merve ANIL



T.C.
İSTANBUL BEYKENT ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ MÜDÜRLÜĞÜ
TEZLİ YÜKSEK LİSANS SINAV TUTANAĞI

2../10../2023

Enstitümüz *Bilgisayar Mühendisliği* Anabilim Dalı *Bilgisayar Mühendisliği* Programı yüksek lisans öğrencilerinden **2220003039** numaralı **Merve ANIL**'ın "*İstanbul Beykent Üniversitesi Lisansüstü Eğitim – Öğretim Yönetmeliği*"nin ilgili maddesine göre hazırlayarak, Enstitümüze teslim ettiği "*Makine Öğrenmesi Yöntemiyle Müşteri Davranışının Tahmin Edilmesi*" konulu tezini, Yönetim Kurulumuzun 19/09/2023 tarih ve 2023/37 sayılı toplantısında seçilen On-Line toplanan biz jüri üyeleri huzurunda, İstanbul Beykent Üniversitesi Lisansüstü Eğitim ve Öğretim Yönetmeliğinin 29. maddesinin 3. fıkrası gereğince 45 dakika süre ile Zoom programı aracılığıyla on-line olarak aday tarafından savunulmuş ve sonuçta adayın tezi hakkında "*OYBİRLİĞİ*" ile "*KABUL*" kararı verilmiştir.

İşbu tutanak, 2 nüsha olarak hazırlanmış ve Enstitü Müdürlüğü'ne sunulmak üzere tarafımızdan düzenlenmiştir.

DANIŞMAN

Dr. Öğr. Üyesi Ta*** Fİ***
(İstanbul Beykent Üniversitesi)

ÜYE

Dr. Öğr. Üyesi Ke*** Gö*** NA***
(İstanbul Beykent Üniversitesi)

ÜYE

Dr. Öğr. Üyesi Ke*** ŞA***
(İstanbul Medipol Üniversitesi)

Adı Soyadı : Merve ANIL
Danışman : Dr. Öğr. Üyesi Talat FİRLAR
Türü ve Tarihi : Yüksek Lisans Tezi, 2023
Alanı : Bilgisayar Mühendisliği
Anahtar Kelimeler : Makine Öğrenmesi, Müşteri Davranışı, LSTM, RNN, CNN

ÖZ

MAKİNE ÖĞRENMESİ YÖNTEMİYLE MÜŞTERİ DAVRANIŞININ TAHMİN EDİLMESİ

Çalışmanın temel amacı, müşteri davranışlarının tahmin edilmesi için makine öğrenimi yöntemlerinin kullanılmasıdır. Bu, işletmelerin müşteri davranışlarını anlama ve analiz etme yeteneklerini artırmalarına yardımcı olan son derece önemli bir amacı temsil etmektedir. İşletmeler için müşteri davranışlarının öngörülmesi, pazarlama stratejilerinin belirlenmesi, ürün ve hizmetlerin iyileştirilmesi ve satışların artırılması açısından kritik bir rol oynamaktadır.

Bu tez çalışması, açık kaynak kodlu bir veri seti kullanarak müşteri davranışlarını öngörmeye yönelik bir uygulama üzerinde odaklanmıştır. Veri seti, 10.000'den fazla veri ögesini içeren zengin bir veri kaynağını temsil etmektedir. Bu büyük veri seti üzerinde gerçekleştirilen analizler, makine öğrenimi alanının bir parçası olan derin öğrenme yöntemlerini kullanarak gerçekleştirilmiştir. Uygulamada, uzun kısa süreli hafıza (LSTM), yinelemeli sinir ağı (RNN) ve evrimsel sinir ağı (CNN) gibi önemli makine öğrenimi algoritmaları kullanılmıştır. Bu algoritmalar, veri setinin karmaşıklığını ve yapılarını anlamak ve öngörmek için kullanılan güçlü öğrenme yöntemleridir. Çalışmanın gerçekleştirildiği ortam Python programlama dilidir.

Çalışmanın sonuçlarına göre, uzun kısa süreli hafıza (LSTM) yönteminin, yinelemeli sinir ağı (RNN) ve evrimsel sinir ağı (CNN) yöntemlerine göre daha iyi tahmin sonuçları verdiği görülmüştür. Bu, işletmelere müşteri davranışlarını anlama

ve tahmin etme konusunda daha iyi bir anlayış sağlama konusunda büyük bir potansiyel sunmaktadır.

Sonuç olarak, bu tez çalışması, işletmelerin müşteri davranışlarını öngörme yeteneklerini artırmak amacıyla makine öğrenimi ve derin öğrenme tekniklerinin nasıl kullanılabileceğini göstermektedir. Bu, işletmelerin rekabet avantajı elde etmelerine ve daha iyi hizmet sunmalarına yardımcı olabilir. Ayrıca, makine öğrenimi ve derin öğrenme alanlarında daha fazla araştırma ve geliştirme potansiyelini de vurgulamaktadır. Bu tür çalışmalar, gelecekte müşteri odaklı işletmelerin daha fazla başarı elde etmelerine yardımcı olabilir.



Name Surname : Merve ANIL
Supervisor : Asst. Prof. Talat FİRLAR
Degree and Date : Master's Thesis, 2023
Major : Computer Engineering
Key Words : Machine Learning, Customer Behavior, LSTM, RNN, CNN

ABSTRACT

PREDICTION OF CUSTOMER BEHAVIOR USING MACHINE METHODOLOGY

The main purpose of this study is to use machine learning methods for predicting customer behaviors. This represents an extremely important objective as it helps businesses enhance their ability to understand and analyze customer behaviors. Predicting customer behaviors is crucial for businesses in terms of determining marketing strategies, improving products and services, and increasing sales.

This thesis focuses on an application that aims to predict customer behaviors using an open-source dataset. The dataset consists of more than 10,000 data entries, representing a rich source of data. Analyses on this large dataset were conducted using deep learning methods, which are part of the field of machine learning. In the application, significant machine learning algorithms such as Long Short-Term Memory (LSTM), Recurrent Neural Network (RNN), and Convolutional Neural Network (CNN) were employed. These algorithms are powerful learning methods used to understand and predict the complexity and structures of the dataset. The programming environment used for this study is Python.

According to the results of the study, the Long Short-Term Memory (LSTM) method yielded better prediction results compared to the Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN) methods. This indicates significant potential for providing businesses with a better understanding of and insight into predicting customer behaviors.

In conclusion, this thesis demonstrates how machine learning and deep learning techniques can be employed to enhance businesses' capabilities in predicting customer behaviors. This can potentially help businesses gain a competitive advantage and provide better services. Moreover, it underscores the potential for further research and development in the fields of machine learning and deep learning. Such studies could contribute to the success of customer-centric businesses in the future.



İÇİNDEKİLER

Sayfa No.

ÖZ

ABSTRACT

TABLolar LİSTESİ	iii
ŞEKİLLER LİSTESİ	iv
KISALTMALAR	iv
SÖZLÜK.....	v
GİRİŞ.....	1

BİRİNCİ BÖLÜM

LİTERATÜRDE YER ALAN BENZER ÇALIŞMALAR.....	3
---	---

İKİNCİ BÖLÜM

YÖNTEM VE MATERYALLER	8
2.1. Müşteri Davranışları Tahmini ve Önemi	8
2.1.1. Müşteri Davranışı Tahmininin Avantajları.....	9
2.2. Makine Öğrenmesine Genel Bakış	11
2.2.1. Yapay Zekâ	13
2.2.2. Makine Öğrenmesi ve Teknikleri	15
2.2.2.1. Denetimli Öğrenme.....	16
2.2.2.2. Denetimsiz Öğrenme	21
2.2.2.3. Yarı Denetimli Öğrenme	21
2.2.2.4. Takviyeli Öğrenme	22
2.2.3. Derin Öğrenme ve Teknikleri	22
2.2.3.1. Yapay Sinir Ağları	24
2.2.3.2. Derin Sinir Ağları	27
2.2.3.3. Derin Oto Kodlayıcılar	28
2.2.3.4. Derin İnanç Ağları	29
2.2.3.5. Boltzmann Makinesi	30
2.2.3.6. Yinelemeli Sinir Ağları.....	31

2.2.3.7. Evriřimli Sinir Ađları.....	32
2.2.4. Uzun Kısa Süreli Bellek	33
2.3. Veri Seti Hazırlığı.....	36
2.4. Arařtırma Yöntemleri	37
2.4.1. Uygulama Hazırlık Süreci	37
2.4.1.1. Derin Öğrenme Yöntemlerinin Kullanılması	38
2.4.1.2. LSTM Yönteminin Kullanılması	38
2.4.1.3. RNN ve CNN Yönteminin Kullanılması	43
2.4.2. Arařtırma Sonuçları	48
SONUÇ	49
KAYNAKÇA.....	51
EKLER	
EK 1. LSTM Uygulama Kodları	54
EK 2. RNN ve CNN Uygulama Kodları	57

TABLÖLAR LİSTESİ

	Sayfa No.
Tablo 1. Veri Seti Özellikleri	36
Tablo 2. Veri Seti Dağılımı	38
Tablo 3. LSTM Algoritması Sonuçları	43
Tablo 4. RNN Algoritması Sonuçları.....	47
Tablo 5. CNN Algoritması Sonuçları.....	47



ŞEKİLLER LİSTESİ

	Sayfa No.
Şekil 1. Makine Öğrenmesi Teknikleri.....	16
Şekil 2. Denetimli Makine Öğrenmesi Süreci	17
Şekil 3. Yapay Zekanın Tarihsel Gelişim Süreci	23
Şekil 4. Yapay Sinir Hücresinin Yapısı.....	25
Şekil 5. Yapay Sinir Ağlarının Yapısı	26
Şekil 6. İleri Beslemeli Ağ Yapısı Modeli	27
Şekil 7. Derin Oto Kodlayıcıların Mimarisi	29
Şekil 8. Derin İnanç Ağı Modeli	30
Şekil 9. Yinelemeli Sinir Ağı Modeli	31
Şekil 10. Evrişimli Sinir Ağı Modeli.....	32
Şekil 11. LSTM Modeli.....	35
Şekil 12. LSTM, RNN ve CNN Yöntemlerinin Test Skorları.....	48
Şekil 13. LSTM, RNN ve CNN Yöntemlerinin Eğitim Skorları.....	48

KISALTMALAR

ADASYN	: Adaptive Synthetic Sampling Approach for Imbalanced Learning
CNN	: Convolutional Neural Network
CRISP-DM	: Cross-Industry Standard Proces for Data Mining
DVM	: Destek Vektör Makineleri
K-NN	: En Yakın Komşu
LSTM	: Long Short-Term Memory
MSE	: Mean Squared Error
NB	: Naive Bayes
RNN	: Recurrent Neural Network
SMOTE	: Synthetic Minority Oversampling Technique
YSA	: Yapay Sinir Ağları

SÖZLÜK

Modelleme: Değişkenler arasında ilişki kurmak amacıyla model kurulur. Daha sonra kurulan model üzerinden tahminleme yapılır.

Müşteri Davranışı: Müşterinin ürün veya hizmetleri, araştırma, satın alma, deneyimleme süreçlerini içeren bir kavramdır. İşletmeler müşteri davranışlarını anlayarak etkili bir şekilde pazarlama stratejilerini artırırlar.

Müşteri Kaybı: İşletmenin mevcut müşterilerini kaybetmesi veya onların işletmeyle olan bağlılıklarının azalması durumun ifade eder.

Müşteri Memnuniyeti: Müşterinin bir ürün, hizmet veya deneyimleriyle ilgili beklentilerinin karşılandığını veya aşıldığını hissettiği durumu ifade eder. İşletmeler için müşteri memnuniyeti, müşterilerin sadakatini artırabilir ve olumlu pazarlamaya katkı sağlayabilir.

Müşteri Sadakati: Müşterinin belirli bir işletmeye veya markaya bağlı kalma isteği ve eğilimini ifade eder. Sadık müşteriler, tekrarlayan alışverişler yapma, olumlu yorumlar yapma ve işletmeyi başkalarına tavsiye etme eğilimindedirler.

GİRİŞ

Makine öğrenmesi, son yıllarda işletmelerin müşteri davranışlarını anlamak ve tahmin etmek için kullandığı bir yöntem haline geldi. Bu teknoloji, büyük veri setleri üzerinde çalışarak, müşteri davranışlarını analiz edebilir ve gelecekteki eğilimleri tahmin ederek müşteriye özgü bir deneyim sunmayı sağlayabilir. Örneğin, bir işletme, müşterilerin satın alma geçmişine, demografik özelliklerine ve çevrimiçi davranışlarına göre özel teklifler sunabilir. Bu, müşterilerin işletme ile bağlılıklarını artırabilir ve tekrar satın alma olasılıklarını yükseltebilir. Bu yöntem sayesinde, müşterilerin ihtiyaçları ve talepleri daha iyi anlaşılabilir ve buna göre pazarlama stratejileri belirlenir. Ayrıca, müşterilerin ne zaman ve ne tür ürünleri satın alabilecekleri de tahmin edilebilir, böylece işletmeler ürünlerini daha iyi planlayabilirler.

Makine öğrenmesi, veri madenciliği, yapay zekâ ve istatistik gibi disiplinlerin birleşimiyle, çeşitli uygulamalarda kullanılan bir yöntemdir. Son yıllarda, müşteri davranışlarını tahmin etmek amacıyla makine öğrenmesi kullanımı oldukça artmıştır. Bu yöntem, müşterilerin davranışlarına dair verilerin toplanması, analiz edilmesi ve sonuçların tahmin edilmesi esasına dayanmaktadır. Makine öğrenmesi, müşteri davranışlarını anlamak ve tahmin etmek için farklı teknikler kullanır. Bunlardan biri, müşterilerin satın alma davranışlarını tahmin etmek için kullanılan "tutum" analizidir. Bu yöntem, müşterilerin geçmiş davranışlarını inceleyerek, onların gelecekteki davranışlarını tahmin etmeyi sağlar. Ayrıca, müşteri verilerinin toplanması ve analiz edilmesi, müşterilerin ilgi alanlarını ve tercihlerini belirlemek için de kullanılır.

Bu tez çalışmasında müşteri davranışlarının makine öğrenmesi yöntemiyle tahmin edilmesini amaçlamaktadır. Bu sayede, müşterilerin satın alma eğilimleri, ürün tercihleri, web sitesindeki hareketleri ve sosyal medya etkileşimleri gibi davranışları analiz edilerek, gelecekteki davranışlarının tahmin edilmesi sağlanacaktır. Böylece, işletmeler müşterilerinin ihtiyaçlarını daha iyi anlayarak, müşteri memnuniyetini artırabilir ve satışlarını artırabilirler. Bu yöntem, işletmelerin müşteri memnuniyetini artırmalarına yardımcı olabilir ve böylece daha yüksek bir müşteri sadakati sağlayabilir.

Makine öğrenmesi yöntemiyle müşteri davranışının tahmin edilmesi tez çalışması aşağıdaki kapsamda yapılacaktır:

- Müşteri davranışları analiz edilerek, müşteri profilinin belirlenmesi.
- Veri toplama ve veri ön işleme işlemlerinin gerçekleştirilmesi.
- Makine öğrenmesi tekniklerinin kullanılarak, müşteri davranışlarının tahmin edilmesi.
- Tahminlerin doğruluğunun test edilmesi ve sonuçların raporlanması.

Bu çalışmanın başlangıcında, kullanılan veri setinin kalitesini artırmak için bazı ön işleme adımları gerçekleştirilecektir. İlk olarak, eksik veri noktalarının sayısı belirlenecektir. Daha sonra ilgili sütunların ortalamaları ile doldurulacaktır. Veri setinde Min-Max normalizasyonu kullanılarak, farklı özelliklerin aynı ölçekte ifade edilmesi sağlanacaktır. Çalışmada üç farklı model türünün kullanılmasıyla müşteri davranışının tahmin edilmesi ele alınacaktır.

BİRİNCİ BÖLÜM

LİTERATÜRDE YER ALAN BENZER ÇALIŞMALAR

Tezin bu kısmında, müşteri davranışı tahminlemeler için kullanılan yöntemler ve benzer çalışmalar araştırılıp başarı durumları değerlendirilmiştir.

Kaynar ve ark., (2017) yapmış oldukları çalışmada, iletişim-haberleşme sektöründe müşteri kaybını analiz etmek amacıyla makine öğrenmesi yöntemlerinin nasıl kullanılabileceği ve bu yöntemlerin performansını nasıl test edeceği amaçlamaktadır. Çalışmada müşteri kaybını analiz etmek için, Yapay Sinir Ağları (YSA), Destek Vektör Makineleri (DVM) ve Naive Bayes (NB) sınıflama yöntemleri olmak üzere üç farklı model kullanılarak analiz gerçekleştirilmiştir. İletişim-haberleşme sektörü müşteri kayıtlarından elde edilen 4667 veri üzerinde müşteri kaybı analiz çalışması yapılmıştır. Çalışma sonucunda elde edilen başarı oranı, duyarlılık ve F-ölçütü değerlerinde alınan değerler sırası ile %91.35, %91,34 ve %90.18 olmuştur. YSA modeli kullanılarak model kurulumu oluşturulmuştur. Hassasiyet değerinde en yüksek değer %89,64 olmuştur ve bu değer NB ile elde edilmiştir. YSA ve NB kullanılan modellerde birbirlerine yakın sonuçlar elde edilmiştir. Kullanılan DVM modeli beklenen performans seviyesinin altında bir sonuç göstermektedir.

Akyiğit ve ark., (2022) yapmış oldukları çalışmada, sigortacılık sektöründe bir müşteriye ait olan yaş, cinsiyet, doğum yeri gibi demografik bilgileri ve araç marka model bilgileri kullanılarak özniteliklerini oluşturulmuşlardır. Bu öznitelikler makine öğrenmesi algoritmalarından Karar Ağacı (Desicion Tree) Algoritması, Rastgele Orman (Random Forest) Algoritması ve K-En Yakın Komşu (K Nearest Neighborhood) Algoritmaları ile terk eden müşterilerin profilleri analiz edilip, terk etme ihtimali olan müşteriler tahmin edilmiştir. Kullanılan 3 farklı sınıflandırma sonucunda en yüksek başarı oranı sırasıyla %87 oranla veren Rastgele Orman (Random Forest) Algoritması, %84 başarı oranıyla Karar Ağacı (Decision Tree) Algoritması ve %80 başarı oranıyla K-En Yakın Komşu (K Nearest Neighborhood) Algoritması olduğu gözlemlenmiştir.

Yaman, (2021) yapmış olduđu çalışmada, müşterileri tanımlayan özniteliklerden müşteri oluşumunun belirlenmesi ve belirlenen müşteri tiplerine göre bir sonraki müşteri veya mevcut müşterilerin davranışlarını tahmin etmek amacıyla makine öğrenmesi modeli geliştirmiştir. Çalışmada sentetik bir veri kümesi kullanılarak Cross-Industry Standard Proces for Data Mining (CRISP-DM) metodolojisi ile ilerletilmiştir. Veri kümesi çeşitli ön işlemlerinden sonra makine öğrenmesi için hazırlanmıştır. Modeller birbirleriyle bağlantılı olacak şekilde kurularak tahminler yapılmıştır. Knime Analytics Platform kullanılarak veri setleri üzerinde Karar Ağacı (Desicion Tree) Algoritması, Destek Vektör Makinesi (Support Vector Machine), Lojistik Regresyon (Logistic Regression) Algoritması, Rastgele Orman (Random Forest) Algoritması ve K-En Yakın Komşu (K Nearest Neighborhood) Algoritması kullanılmıştır. Uygulanan algıritmalar sonucunda başarısı en yüksek olan Rastgele Orman (Random Forest) Algoritması olduđu saptanmıştır. Bu sonuçlara istinaden Python’da SKLearn yazılımında, aktiviteler arası ilişkilerden yola çıkarak birbiri arkası gelmesi olasılığı hesaplanmıştır. Hesaplamalar sonucunda aktiviteler ve olasılık tablosu oluşturulmuştur. Elde edilen bilgiler ile her bir aktivite için Rastgele Orman (Random Forest) Algoritması kullanılarak her bir aktivite için ayrı ayrı modeller oluşturulup birbiriyle ilişkilendirilmiştir. Oluşturulan modeller hem veri setinde barındırdığı parametreler hem de yapay bir veri seti olması nedeniyle tahminlerin başarı oranı %100’e yakın, çok sınıflı sınıflandırma modelinde başarı oranı %75 olmuştur. Elde edilen sonuçlar ile müşteri davranışını değerlendirmek isteyen firmalar için müşteri yolculukları analizi (Customer Journey Analytics) zahmetli, değerli ve dikkate alınması gereken bir süreçtir.

Arı ve Akbıyık (2022) yapmış oldukları çalışmada, müşteri yorumlarının işletmeler için önemli olduđu ve müşteri memnuniyetini arttırmak için doğru bir şekilde ele alınması gerektiği gerçeği göz alındığında son derece yararlıdır. Çalışma, online bir alışveriş platformunda gerçekleştirilen bir anket çalışmasından elde edilen veriler kullanılarak yapılmıştır.

Bu veriler, müşterilerin ürünlerle ilgili yorumlarını içermekte ve bu yorumların faydalı veya faydasız olarak sınıflandırılması amaçlanmaktadır. Lojistik regresyon yöntemi, verilerin analizinde kullanılmış ve sonuçlar oldukça tatmin edici olmuştur. Çalışmanın en önemli sonuçlarından biri, lojistik regresyon modelinin, müşteri yorumlarının faydalı veya faydasız olarak sınıflandırmasında oldukça yüksek bir başarı oranı sağlanmasıdır. Ayrıca, müşteri yorumlarına etki eden faktörlerinde belirlenmesine yardımcı olmuştur. Bu faktörler arasında ürün özellikleri, fiyat ve teslimat süresi gibi unsurlar yer almaktadır. Çalışmanın bir diğer önemli yönü, lojistik regresyon modelinin işletmelerin müşteri yorumlarına daha hızlı ve doğru bir şekilde yanıt vermesine yardımcı olmasıdır. Bu müşteri memnuniyetini artırmak ve işletmelerin ürün veya hizmetlerini geliştirmek için son derece önemlidir. Çalışmanın başarı oranları, farklı ölçütler kullanılarak hesaplanmıştır. Örneğin çalışmada kullanılan lojistik regresyon oranı %88,4 olarak hesaplanmıştır. Ancak modelin hassasiyeti %86,2 özgüllüğü ise %90,1 olarak belirlenmiştir. Bu sonuçlar modelin müşteri yorumlarını faydalı veya faydasız olarak sınıflandırmada oldukça başarılı olduğunu göstermektedir. Ayrıca çalışmada farklı değişkenlerin (ürün özellikleri, ürün fiyatı, teslimat süresi vb.) incelenmiştir. Bu analizler sonucunda, müşteri yorumlarına etki eden faktörlerin belirlendiği ve işletmelere yol gösterdiği bilgiler sunulduğu görülmüştür. Sonuç olarak çalışma yüksek başarı oranları elde ederek müşteri yorumlarının analizinde etkili bir araç olarak kullanılabileceğini göstermektedir.

Göy ve ark., (2020) yaptıkları çalışmada, internet hizmet sağlayıcısı olarak hizmet veren kurumdan alınan veri seti kullanılmıştır. Ön işleme adımları yapıldıktan sonra benzer özellikleri içeren bilgiler çıkarılarak çeşitli veri madenciliği yöntemleri kullanılmıştır. Kullanılan bu yöntemler ile riskli müşterileri tespit eden bir model geliştirilmiştir. Geliştirilen model kullanılarak sonuçlar çıkarılmış ve karşılaştırılmıştır. Karşılaştırma yaparken dengesiz veri setleri kullanılması uygun olan eğri altında kalan alan metriği kullanılmıştır. Özellik seçim tekniği olarak Ki-Kare ve sınıflandırıcı algoritması olarak Rastgele Orman (Random Forest) Algoritması kullanıldığında başarı oranı 0.936 elde edilmiştir.

Sonuç olarak gelecek çalışmalarda daha büyük veri seti kullanılacak ve kullanılmayan diğer özellik seçim yöntemleri de kullanılarak geliştirecek olan modeller kıyaslanacaktır.

Koruyan ve ark., (2022) yapmış olduğu çalışmada, sikayetvar.com sitesinden alınan şikayetlerin iade, değişim, geri ödeme, teslimat, kargo, müşteri ilişkileri ve hizmetleri, garanti ve servis olmak üzere dört kategoriye alınmış ve alınan bu kategoriler üzerinde analizler yapılmıştır. Dört kategori üzerinde Lojistik Regresyon, Doğrusal Destek Vektör Makineleri ve Olasılıksal Dereceli Azalma kullanılarak eğitim ve test süreci gerçekleştirilmiştir. Lojistik Regresyon ile %80 başarı sağlanmıştır.

Ercan (2022) yapmış olduğu çalışmada, havayolu seyahatinden memnun olmayan müşterilerin tahmin edilmesi yapılmıştır. Çalışmada Birleşik Devletleri'ndeki havayolu şirketinden alınan veriler kullanılarak uçuş seyahatinden nötr veya memnun olmayan müşterilerin topluluk öğrenmesi yöntemleriyle tahmin edilmesi amaçlanmıştır. Çalışma kapsamında Gradient Boosting, XGBoost ve Rastgele Orman yöntemleri kullanılmış ve modellenmesi yapılmıştır. Oluşturulan modellerin sonuçlarına göre hesaplanan performans ölçütleri ile modellerin karşılaştırmalı olarak tahmin performansları kıyaslaması yapılmıştır. En iyi doğruluk oranı %96,4 iken en iyi Özgüllük ve Negatif Oranı değerleri %97.7 ve %96 olmuştur.

Uyanık ve ark., (2021) yapmış oldukları çalışmada, aboneliğinden ayrılan müşterilerin tahmin edilmesi yapılmıştır. Çalışma Lojistik Regresyon, Karar Ağacı, Yapay Sinir Ağları, Torbalama ve Artırma sınıflandırma modelleri kullanılmıştır. Veri setinde sınıf dengesizliği olduğu için SMOTE ve ADASYN tekniği ile örnekleme yapılmıştır. Çalışmada, veri ön işleme sırasında, kullanılmayan ve önemsiz özelliklerin veri setinden çıkartılması, normalizasyon, şifreleme ve aşırı örnekleme gibi birçok yöntem kullanılmıştır. Modellerin karşılaştırılması yapılırken belirlenmiş olan Karmaşıklık Matrisi göz önüne alınmıştır. Performans ölçütü olarak Doğruluk oranı (%80.3), Geri Çağırma (%75.0), Hassiyet(%93.9), Özgüllük(%95.7), Dengelenmiş Doğruluk Oranı (%80.4), F1 Skoru (%76.8) ve ROC-AUC Oranı (%87.1), Çalışma Zamanı (15.34 Saniye) kullanılarak iyi sonuçlar elde edilmiştir.

Yapılan iki tahmin modeli de incelendiğinde benzer sonuçlar verdiği tespit edilmiştir. İki tahmin modelinde en iyi sonucu veren Light Gradient Boosting Machine sınıflandırıcısı olmuştur. Geri Çağırma performansı ölçütünün %75.3 ile en iyi sonuç verdiği yöntem YSA olmuştur.

Şeker (2020) yapmış olduğu çalışmada, telekomünikasyon, bankacılık, sigortacılık gibi sektörlerde kullanılan müşterilerin kaybını önceden tahmin etmek amacıyla müşteri kaybı analizi yöntemini kullanmıştır. Yapılan bu tahminleme çalışması sayesinde müşterilerin kaybını engellemek için müşteri ilişkileri yönetimi sayesinde çözümler üretebilmektedir. Telekomünikasyon, bankacılık ve sigortacılık gibi müşteri devamlılığı olan sektörlerde müşteri kaybı çok önemli bir öneme sahiptir. Çalışmada müşteri davranışını tahminleme yapılırken veri madenciliğinin başarılı sonuç vereceğini belirtilmiştir.

İKİNCİ BÖLÜM

2. YÖNTEM VE MATERYALLER

Müşteri davranışı tahmini, işletmelerin müşterileri hakkında önemli bilgiler edinmesine ve pazarlama stratejilerini buna göre şekillendirmesine yardımcı olan bir analiz yöntemidir. Bu yöntem, müşterilerin geçmiş davranışlarından, demografik özelliklerinden ve diğer faktörlerden elde edilen verileri kullanılarak gelecekteki davranışları hakkında tahminler yapar ve işletmelerin ürün ve hizmetlerini müşterilerinin ihtiyaçlarına göre şekillenmesini sağlar. Örneğin, bir müşterinin belirli bir ürüne ne zaman ihtiyaç duyacağı veya hangi promosyon kampanyalarından etkileneceği hakkında tahminler yaparak, işletmeler o ürünü müşterinin ihtiyacı olduğu zamanda stokta tutabilir. Ayrıca müşterilerin hangi kanallardan satın alma işlemlerini gerçekleştirdikleri hakkında da bilgi edinmek mümkündür. Bu sayede, işletmeler, müşterilerinin tercih ettiği kanallar üzerinden daha fazla pazarlama çabası gösterebilir.

2.1. Müşteri Davranışları Tahmini ve Önemi

Müşteri davranışlarının tahmin edilmesi, veri analitiği ve yapay zeka gibi teknolojilerin kullanılmasıyla giderek daha keskin hale gelmektedir. Büyük veri analizi sayesinde işletmeler, müşterilerin geçmiş alışkanlıklarını, demografik bilgilerini ve çevrimiçi etkileşimlerini inceleyerek gelecekteki davranışları daha doğru bir şekilde tahmin edebilirler. Bu, özellikle e-ticaret, finans, sağlık ve seyahat gibi birçok sektörde daha iyi hizmet sunma potansiyelini artırırken, müşteri odaklı bir yaklaşımın benimsenmesine de yardımcı olur. Sonuç olarak, müşteri davranışlarını tahmin etme, işletmeler için sadece bir rekabet avantajı sağlamakla kalmaz, aynı zamanda müşteri memnuniyetini artırarak uzun vadeli başarıyı destekler.

Müşteri davranışlarını tahmin etmek aynı zamanda maliyet tasarrufu sağlayabilir. İşletmeler, tedarik zinciri yönetimi, envanter kontrolü ve personel planlaması gibi alanlarda daha verimli çalışabilirler. Örneğin, bir perakende mağazası, müşteri taleplerini daha iyi tahmin ederek stok maliyetlerini azaltabilir ve atıl stokları önleyebilir. Aynı zamanda, yoğun dönemlerde ek personel istihdam etmek yerine

doğru zamanda doğru sayıda personel tutma konusunda daha iyi bir denge sağlayabilirler.

Müşteri davranışları tahmini yapıldığında aynı zamanda işletmelerin müşteri sadakatini artırmalarına yardımcı olur. İşletmeler, müşterilerinin ihtiyaçlarını belirler ve onların isteklerine yanıt verebildiklerini göstermek için özelleştirilmiş kampanyalar sunabilirler. Bu müşterilerin işletmelere olan bağlılıklarını artırmaya ve daha sık satın almaya teşvik eder.

Ancak, müşteri davranışları tahmini ile ilgili bazı zorluklar vardır. Özellikle, yanıltıcı veriler veya veri eksikliği nedeniyle doğru tahminler yapmak zor olabilir. Müşterilerin davranışlarının karmaşıklığı ve değişkenliği nedeniyle, tahminler her zaman doğru olmayabilir. Bu nedenle işletmeler doğru ve güvenilir veri kaynaklarına sahip olmalı ve verileri doğru bir şekilde analiz etmelidirler.

Sonuç olarak, müşteri davranışları tahmini, işletmelerin müşterileri hakkında önemli bilgiler edinmelerine ve pazarlama stratejilerini buna göre şekillendirmelerine yardımcı olan bir analiz yöntemidir. Doğru bir şekilde kullanıldığında, işletmelerin müşteri sadakatini artırarak, müşteri ihtiyaçlarına daha iyi yanıt vermelerine ve rekabet avantajı sağlar. Ancak, bu yöntemin doğru bir şekilde uygulanması için işletmelerin doğru veri kaynaklarına ve teknolojik altyapıya sahip olmaları gerekmektedir.

2.1.1. Müşteri Davranışı Tahmininin Avantajları

Müşteri davranışı tahmini, günümüzde iş sektörlerinde giderek daha fazla önem kazanmaktadır. İşletmeler için, müşteri davranışlarını tahmin etmek ve analiz etmek, karar alma süreçlerinde kritik bir yol oynamaktadır. İşletmeler, müşterilerinin ihtiyaç ve isteklerini daha iyi anlayarak daha etkili pazarlama stratejileri oluşturabilirler. Bu nedenle, müşteri davranışı tahmini, işletmeler için oldukça faydalı bir araçtır. İşletmelerin en önemli avantajları aşağıda belirtilen şekildedir.

- Müşteri memnuniyeti: Müşteri davranışı tahmini, işletmelerin müşterilerinin ihtiyaçlarını ve beklentilerini daha iyi anlamalarını sağlar. Bu da müşterilerin memnuniyetlerini artırır ve işletmenin müşteri sadakatini sağlamasına yardımcı olur.

- Pazarlama stratejilerini optimize etme: Müşteri davranışı tahmini, işletmelerin müşterilerine daha etkili pazarlama stratejileri geliştirmelerine yardımcı olur. İşletmeler, müşterilerin tercihleri ve ihtiyaçları hakkında daha fazla bilgi sahibi olduklarından, daha iyi hedeflemeli kampanyalar oluşturabilirler.
- Stok yönetimi: Müşteri davranışı tahmini, işletmelerin stok yönetimini daha iyi yapmalarına yardımcı olur. İşletmeler, müşterilerin taleplerini önceden tahmin ederek stoklarını daha doğru bir şekilde yönetebilirler.
- Karlılık: Müşteri davranışı tahmini, işletmelerin karlılığını artırır. İşletmeler, müşterilerin hangi ürünleri daha fazla satın aldıklarını ve hangi kanallar üzerinden satın aldıklarını öğrenerek, ürünlerin fiyatlarını ve dağıtım kanallarını optimize edebilirler.
- Rekabet avantajı: Müşteri davranışı tahmini, işletmelere rekabet üstünlüğü kazanmalarına katkı sağlar. İşletmeler, rakiplerinin müşteri davranışlarını takip ederek, kendilerine avantaj sağlayacak pazarlama stratejileri geliştirebilirler.

Müşteri davranışı tahminleri, şirketlerin kişiselleştirilmiş hizmetler sunma yeteneği ile müşteri memnuniyetini artırır. İşletmeler müşterilerinin geçmiş alışkanlıklarını ve tercihlerini analiz ederek, müşterilerine yönelik özel öneriler, teklifler ve fırsatlar sunabilirler. Böylece müşterilerin işletmeleri ile daha güçlü bir bağ kurmalarını ve sadakatlerini artırmalarını sağlar. Bu, müşterilerin işletme ile daha güçlü bir bağ kurmalarını ve sadakatlerini artırmalarını sağlar.

Müşteri davranışı tahmini yaparak stok yönetimi ve talep tahmininde büyük bir yardımcıdır. Özellikle perakende sektörü için önemli olan bu avantaj, işletmelerin envanter maliyetlerini minimum düzeye indirmeye ve aynı zamanda müşteri talebini karşılamalarına yardımcı olur.

Birden çok avantajı bulunan bu tahminleme yöntemi sayesinde ürün geliştirme süreçlerine oldukça yararı olur. İşletmeler, müşteri geri bildirimlerini ve davranışlarını analiz ederek yeni ürünlerin geliştirilmesi veya mevcut stoklarında bulunan ürünlerin iyileştirilmesi hususunda karar alabilirler.

Sonuç olarak, müşteri davranışı tahminleri işletmelere rekabet üstünlüğü kazandırır. Müşteri ihtiyaçlarını anlama, pazarlama stratejisi belirleme, maliyetleri düşürme, müşteri memnuniyeti artırma ve uzun vadeli başarıyı sağlama konularında önemli avantajlar sunar. Bu nedenle, işletmelerin müşteri davranışı tahminleri etkin bir şekilde kullanmaları, rekabetçi bir pazarda ayakta kalmalarına ve büyümelerine yardımcı olur.

2.2. Makine Öğrenmesine Genel Bakış

Makine öğrenmesi, bilgisayarların verileri analiz ederek öğrenmelerine ve sonuçlarını tahmin etmelerine olanak sağlayan bir yapay zekâ alt dalıdır. Bu teknoloji oldukça eski bir tarihe sahiptir. İlk örneklerinden biri, 1950'lerde matematikçi ve bilgisayar bilimcisi Alan Turing tarafından önerilen Turing testidir. Turing Testi, yapay zekânın veya bilgisayar programlarının insan benzeri düşünme yeteneğine sahip olup olmadığını değerlendirmek için kullanılan bir kavramdır. İngiliz matematikçi ve bilgisayar bilimcisi Alan Turing tarafından 1950 yılında ortaya atılmıştır. Bu test, yapay zekânın zekâ seviyesini ölçmek ve insan benzeri zekâyâ sahip olup olmadığını belirlemek amacıyla geliştirilmiştir.

Turing Testi, temel olarak bir insanın bir bilgisayar programı veya bir yapay zekâ ile yazılı metin tabanlı bir iletişim kanalı aracılığıyla sohbet etmesini içerir. İnsan, bu iletişim sırasında iki farklı taraf ile etkileşimde bulunur: bir insan ve bir bilgisayar programı. İnsan, bu etkileşim sırasında hangi tarafın insan, hangi tarafın bilgisayar programı olduğunu belirlemeye çalışır. Eğer insan, hangi tarafın insan olduğunu doğru bir şekilde ayırt edemiyorsa, yani bilgisayar programı insan gibi davranıyorsa, bu durumda bilgisayar programı Turing Testi'ni geçmiş olarak kabul edilir. Turing Testi'nin temel fikri, bir makinenin düşünme yeteneğini değerlendirmek için zekâ, dil anlama, mantık yürütme ve insan gibi davranma yeteneklerini içeren bir dizi beceriye sahip olması gerektiğini savunur. Ancak, Turing Testi eleştirilere de maruz kalmıştır. Eleştirmenler, bu testin sadece yüzeydeki dil becerilerini ölçtüğünü ve gerçek zekâyı yeterince değerlendiremediğini ileri sürmüşlerdir. Ayrıca, bazı bilgisayar programları zekâlarını gizlemek veya sınırlamak için tasarlanmış olabilirler. Turing Testi, yapay zekânın gelişimini ve insan benzeri davranışları simüle etme çabalarını teşvik etmiş ve bu alandaki önemli bir ilham kaynağı olmuştur. Ancak, hâlâ yapay zekânın gerçek bir

insan gibi düşünme yeteneğine sahip olup olmadığını kesin bir şekilde belirlemenin zorluğu devam etmektedir.

Makine öğrenmesi teknolojisi, günümüzdeki gelişmiş haline 1980’lerde ve 1990’larda ulaşmıştır. Bu dönemde, yapay sinir ağları, genetik algoritmalar ve karar ağaçları gibi makine öğrenmesi teknikleri ortaya çıkmıştır. Bu teknikler sayesinde, makine öğrenmesi daha da gelişmiş ve birçok farklı uygulama alanında kullanılmaya başlanılmıştır. Özellikle son yıllarda, büyük veri kümelerinin analizi ve yapay zekâ uygulamalarının artmasıyla birlikte, makine öğrenmesi daha da önemli hale gelmiştir.

Günümüzde makine öğrenmesi, birçok farklı algoritma ve tekniklerle çalışır ve bu algoritmalar verileri işler, model oluşturur ve sonuçları tahmin eder. Örneğin, bir makine öğrenmesi modeli, bir resimdeki nesneleri tanımak veya bir müşterinin alışveriş alışkanlıklarına dayalı olarak tavsiyelerde bulunmak gibi birçok farklı uygulama için kullanılabilir.

Bir makine öğrenmesi görevinin gerçekleştirilmesi için beş temel adım aşağıdaki gibidir.

Veri Toplama: Çeşitli veri kaynakları, metin belgesi biçimleri, web siteleri ve benzeri yerlerden veri toplama sürecini içerir. Toplanan bu geçmiş veriler, gelecekteki öğrenmelerin temelini atmaktadır ve ilgili verilerin çeşitliliği, yoğunluğu ve hacmi, makine öğrenmesi sürecinin etkinliğini artırmak için ne kadar yüksek nitelikli olursa, öğrenme süreci o ölçüde üstün olacaktır.

Verilerin Hazırlanması: Verilerin eğitim modelini oluşturmak için sisteme veri girişi yapılmadan önceki aşama, verilerin toplanması, temizlenmesi, dönüştürülmesi ve hazırlanması gibi önemli adımları içerir. Bu süreç, verilerin modelin gereksinimlerini karşılayacak şekilde uygun hale getirilmesini sağlar ve başarılı bir makine öğrenimi modelinin temelini oluşturur.

Modeli Eğitmek: Modeli oluşturmak amacıyla uygun algoritmanın seçilmesini ve verilerin temsili için uygun bir model oluşturma sürecini kapsar. Eğer denetimli eğitim yapılacaksa bu aşamada veriler eğitim ve test kümelerine ayrılır. Eğitim kümesi, sistem tarafından işlenerek, test aşamasında kullanılmak üzere verilere dayalı bir modelin oluşturulmasını sağlar. Bu süreç, makine öğrenimi modelinin

verilerle etkileşime girmesini, öğrenmesini ve gelecekteki tahminlerde kullanmak üzere bilgi kazanmasını sağlar.

Modelin Değerlendirilmesi: Sistemin doğruluğunu sınamak amacıyla önceki aşamada oluşturulan modelin test kümesine uygulanmasıyla gerçekleştirilir. Ardından elde edilen sonuçlar kaydedilir. Bu aşama, modelin performansını en iyi şekilde değerlendirmeyi hedefler ve modelin daha önce görmediği veriler üzerindeki başarısını objektif şekilde ölçmeyi amaçlar. Bu modelin, gerçek dünya koşullarında ne kadar iyi çalıştığını anlamak için kritik bir adımdır. Bu adım sayesinde modelin iyileştirilmesi veya ayarlanması gerekip gerekmediğini belirlemeye yardımcı olur.

Performans Artırmak: Bu adım, modelin doğruluğunu veya etkinliğini artırmak için çeşitli stratejilerin ve yöntemlerin değerlendirilmesini içerir. Bu adım, farklı bir modelin seçilmesini gerektirebileceği gibi, verimliliği artırmak için daha fazla değişkenin dikkate alınmasını da gerektirebilir. Bu nedenle, genellikle daha fazla veri toplanması ve model oluşturmak için önceki aşamalardan geçilmesi gerekebilir. Bu aşama, modelin performansını optimize etmek ve daha iyi sonuçlar elde etmek için çeşitli iyileştirme yöntemlerinin uygulamasını içerir.

2.2.1. Yapay Zekâ

Yapay zekâ, insan düşüncesinin ve davranışlarının benzerlerini yapay olarak üretmeyi amaçlayan bir bilgisayar teknolojisidir. Modern Yapay Zekâ fikri ilk kez dijital makineleri tasarlayan Alan Turing tarafından ortaya atılmıştır. Turing, 1950’li yıllarda düşüncelerini ilk kez yazılı olarak yayınladığı “Bilgi İşlem Makineleri ve Zekâ” makalesinde “Makineler düşünebilir mi?” sorusunu tartışarak yapay zekanın temellerini oluşturmuştur (Turing, 1950).

Çalışmasında makinelerin düşünebilme durumlarını karşıt görüşlere de cevap vererek ayrıntılı olarak açıklamıştır. Turing yaptığı bu çalışma ile Yapay Zekânın kurucusu olarak isimlendirilebilir.

Yapay zekâ ve insan zekâsını karşılaştırmak gerekirse aşağıdaki şekilde listelenebilir.

- Hız ve Kapasite: Yapay zekâ sistemleri, büyük miktarda veriyi çok hızlı bir şekilde işleyebilir ve büyük ölçekli hesaplamaları yapabilirken, insan zekâsı bu kadar hızlı ve yüksek kapasiteli değildir.
- Öğrenme Hızı: Yapay zekâ sistemleri, çok hızlı bir şekilde belirli görevleri öğrenme yeteneğine sahiptir. İnsanlar ise daha uzun süreler boyunca deneyimler ve eğitim alarak öğrenirler.
- Unutma: İnsan zekâsı, öğrendiği bilgileri zamanla unutabilirken, yapay zekâ sistemleri unutmaz. Ancak yapay zekâ sistemlerinin unutmaması bazen istenmeyen sonuçlara yol açabilir.
- Bağımsızlık: Yapay zekâ sistemleri, insan müdahalesi olmadan otomatik olarak çalışabilirler. İnsanlar ise genellikle yönlendirme ve denetim gerektirir.
- Duygusal Zekâ: İnsanlar duygusal zekâ sahibidirler ve duygusal tepkiler verirler. Yapay zekâ ise duygusal zekâ eksikliği ile çalışır.
- Yaratıcılık: İnsanlar yaratıcı düşünme yeteneğine sahiptirler ve yeni fikirler üretebilirler. Yapay zekâ ise mevcut verilere dayalı olarak işlem yapar.
- İnsan İletişimi: İnsanlar çok karmaşık ve çeşitli iletişim yöntemleri kullanabilirler. Yapay zekâ sistemleri ise genellikle önceden programlanmış iletişim yöntemlerini kullanır.
- Hata Yapma Olasılığı: İnsanlar, hata yapma olasılığına sahiptirler ve hatalar yapabilirler. Yapay zekâ ise doğru programlandığında hata yapma olasılığı daha düşüktür.
- Esneklik: İnsanlar farklı türde görevleri öğrenebilir ve uyum sağlayabilirler. Yapay zekâ sistemleri ise genellikle belirli bir görev veya alan için tasarlanmıştır.
- Bilinç: İnsanlar bilinç ve öz farkındalığa sahiptirler. Yapay zekâ sistemleri bilinç veya öz farkındalığa sahip değildirler ve sadece programlandıkları görevleri yerine getirirler.

Bu maddeler, yapay zekâ ve insan zekâsının farklılıklarını ve benzerliklerini vurgulamaktadır. İnsan zekâsı duygusal ve yaratıcı yeteneklere sahipken, yapay zekâ daha hızlı hesaplamalar yapabilir ve belirli görevleri çok verimli bir şekilde gerçekleştirebilir.

Bilgisayar bilimleri, istatistik, matematik, nörobilim, psikoloji ve felsefe gibi birçok alanlardan yararlanan yapay zekânın amacı, insanlar gibi düşünebilen, hareket edebilen, öğrenebilen ve karar verebilen bilgisayar oluşturmaktır. Günümüzde yapay zekâ hızla gelişmekte ve çeşitli sektörlerde kullanılmaktadır. Özellikle sağlık sektöründe hastalık teşhisi, endüstriyel alan için üretim ve kontrol, sanayi, savunma, sanat ve müzik gibi birçok sektör örnek verilebilir. Birçok alanda kullanılan yapay zekâ, ayrıca müşteri davranışları, verinin işlenip yorumlanması gibi alanlarda kullanılmaktadır. (Yılmaz, 2022)

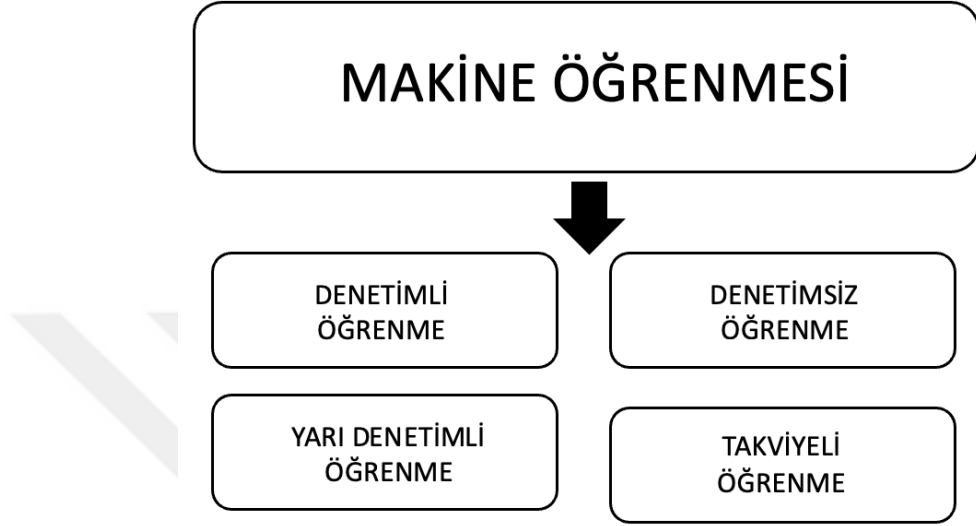
Yapay zekâ, tahminlemelerin yanı sıra kümeleme ve sınıflandırma gibi alanlarda da başarılı bir şekilde kullanılmaktadır. (Atalay ve Çelik, 2017)

2.2.2. Makine Öğrenmesi ve Teknikleri

Makine öğrenmesi kavramını derinlemesine anlamak için, öncelikle öğrenme kavramının ne anlama geldiğini anlamak önemlidir. Öğrenme, yeni bilgileri ve deneyimleri kazanmak için gözlemleri ve etkileşimler kullanarak zihinsel olarak işleyip anlamlandırılan bir evredir. Bu, insanların bilgiyi edinme ve uygulama şekliyle benzerlik gösterirken, makine öğrenmesi bu süreci makinelerin otomatik olarak yapabilmesini sağlama amacını taşır. Bu sayede makineler, büyük veri setleri üzerinde karmaşık desenleri tanıyabilir, tahminlerde bulunabilir ve kararlar alabilir.

Makine öğrenmesi, bilgisayar bilimlerinde yapay zekâ uygulamaları için geliştirilmiş bir tekniktir ve bilgisayar bilimlerinde önemli bir rol oynamaktadır. Temel amacı, algoritmaların öğrenme yeteneği kazanması sağlamak ve ardından tahminlerde bulunabilmesini mümkün kılmaktır. Makine öğrenmesi, geleneksel optimizasyon algoritmalarının aksine, sıralı talimatları takip etmek yerine eğitim veri setlerinden öğrenme ve tahmin yapma sürecini gerçekleştirmeyi hedef alır. Bu sayede, bilgisayarlar, veri üzerinden öğrenme işlemini gerçekleştirir ve giriş verileri aracılığıyla tahminlerde bulunmayı amaçlamaktadır. Makine öğrenimi, karmaşık problemleri çözmek, desenleri tanımak ve veri analizi yapmak gibi birçok farklı alanlarda kullanılır ve bu sayede otomatik kararlar alabilir ve insan benzeri yetenekleri geliştirebilir.

Makine öğrenmesi metodunun tahmin ve analizlerini başarılı kılmak için öğrenme modelleri kullanılmaktadır. Bu öğrenme modelleri; denetimli öğrenme, denetimsiz öğrenme, yarı denetimli öğrenme ve pekiştirmeli öğrenme modelleridir. Şekil 1 de makine öğrenmesi teknikleri gösterilmiştir.



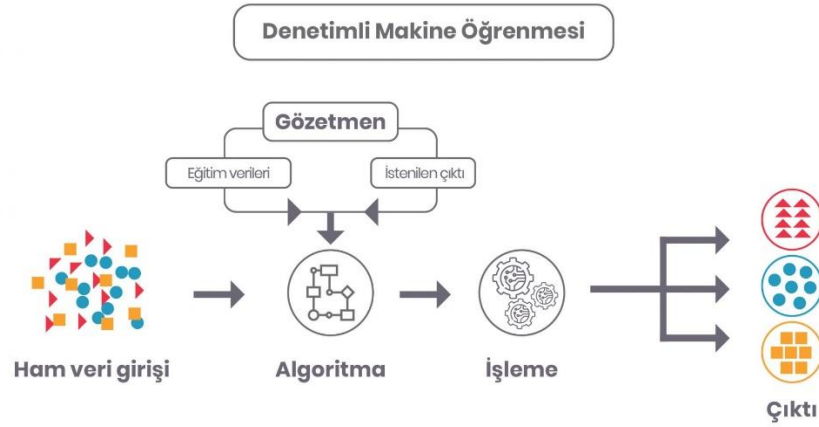
Şekil 1. Makine Öğrenmesi Teknikleri

2.2.2.1. Denetimli Öğrenme

Denetimli öğrenme yöntemiyle, sisteme giriş verileriyle birlikte hedef çıktı değerleri sağlanır. Giriş verilerinin özellikleri incelenir ve çıkış değerlerini tahmin etmek veya öğrenmek için algoritma kullanılır. Bu süreçte, mevcut veriler üzerinden veri analizi yapılır. Bilinen veriler üzerinde bir sınıflandırma yapılarak, sonuçları bilinmeyen veri kümesi için tahminlerde bulunmaya çalışır (Aydın ve Özkul, 2015).

Denetimli makine öğrenmesi, regresyon ve sınıflandırma iki alt grubu bulunmaktadır. Regresyon, sayısal değere dayalı tahminler yaparken sınıflandırma, verileri belirli kategorilere atamak için kullanılır (Çağlayan Akay, 2018).

Şekil 2 de denetimli makine öğrenmesi süreci belirtilmiştir.



Şekil 2. Denetimli Makine Öğrenmesi Süreci

Kaynak: *Makine Öğrenmesi(Machine Learning)Nedir?* (2021, Kasım 26). turhost.com: <https://www.turhost.com/blog/makine-ogrenmesi-machine-learning-nedir/#serp> adresinden alındı

Denetimli öğrenmede, en çok kullanılan yöntemler aşağıdaki gibidir:

- Doğrusal Regresyon: Sayısal veriler arasındaki ilişkileri analiz etmek ve bu ilişkiyi ifade eden bir doğrusal model oluşturmak için kullanılan bir istatistiksel yöntemdir. Bu yöntem, bağımlı bir değişken ile bir veya birden fazla bağımsız değişken arasındaki ilişkiyi matematiksel olarak modellemek amacıyla kullanılır. Doğrusal regresyon, bu ilişkiyi bir doğru çizerek ifade eder, bu doğru da verilere en iyi uyan doğru olur. Bu yöntem, özellikle tahmin ve analiz gibi veri analitiği uygulamalarında yaygın olarak kullanılır. Verilerin gelecekteki davranışını tahmin etmek için önemli bir araçtır.
- Lojistik Regresyon: Sınıflandırma problemleri çözmek için kullanılan istatistiksel bir yöntemdir. Bu yöntem, iki veya daha fazla sınıf arasında bir karar sınıfı oluşturur. Özellikle, bir olayın veya nesnenin belirli bir sınıfa ait olup olmadığını tahmin etmek için kullanılır. Lojistik regresyon, giriş verilerini analiz ederek, bu verilerin hangi sınıfa ait olduğunu belirlemeye yardımcı olan bir olasılık modeli oluşturur. Bu nedenle, sınıflandırma problemlerinin çözümünde oldukça yaygın

olarak kullanılır. Makine öğrenmesi ve veri madenciliği alanlarında önemli bir rol oynar.

- **Karar Ağaçları:** Karar verme süreçlerini ağaç yapısı şeklinde temsil eden bir veri madenciliği ve sınıflandırma yöntemidir. Bu yöntem, veri kümesini bölerek ve ver özelliklerine dayalı kararlar alarak tahminler yapar. Başlangıçta bir kök düğümden başlayarak, veriyi belirli kriterlere göre sınıflara ayırır ve her bir dalda spesifik kararlar alır. Sonunda, bir yaprak düğümünde sonuç elde eder. Karar ağacı, karmaşık karar verme süreçlerini basitleştirerek anlamamıza yardımcı olur. Veri madenciliği, yapay zekâ ve sınıflandırma uygulamaları gibi birçok alanda kullanılır. Ayrıca, sonuçları yorumlama kolaylığı ve görsel olarak temsil edebilir olmaları nedeniyle bu yöntem, veri analizi ve karar verme süreçlerinde etkili bir araçtır.
- **Destek Vektör Makineleri (DVM):** Sınıflandırma veya regresyon problemlerinde kullanılan bir istatistiksel yöntemdir. Bu yöntem, veri analizi ve model oluşturmaya oldukça güçlü ve esnek bir şekilde birleştirir. Temel olarak, DVM, verileri bir uzayda temsil eder ve bu uzayda sınıflandırma veya regresyon için en iyi ayrımı sağlamak için optimize eden karar sınırları oluşturmaya sağlar. Veri noktalarını belirli sınıflara veya değerlere atamak için kullanılır. DVM, özellikle veri madenciliği, yapay zekâ, görüntü işleme ve doğal dil işleme gibi birçok uygulama alanında kullanılan güçlü bir araçtır. Bu nedenle, veri analizi ve karar verme süreçlerinde başarılı sonuçlar elde etmek için sıklıkla tercih edilen bir yöntemdir.
- **K-En Yakın Komşu (K-NN):** Sınıflandırma veya tahminleme yaparken veri analizinde kullanılan bir yöntemdir. Bu yöntem, veri noktaları arasındaki benzerliği değerlendirmek ve bir veri noktasını sınıflandırmak veya tahmin etmek için en yakın komşularının sınıf veya değerlerinin dikkate almak için kullanılır. K-NN yöntemi, basit ve anlaşılması kolay bir yaklaşım olmasının yanı sıra, özellikle sınıflandırma ve tahmin problemlerinde etkilidir. Bu yöntem, benzerlik ölçüleri ve K değeri seçimi gibi faktörleri dikkate alarak veri analizi ve

model oluřturma srelerinde kullanılır. K-NN zellikle desen tanıma, neri sistemleri ve medikal teřhis gibi birok uygulama alanında bařarıyla kullanılan bir algoritmadır.

- Yapay Sinir Aęları: Biyolojik sinir aęlarını taklit eden matematiksel yapılar olan yapay sinir aęları, eřitli derin ęrenme problemlerinde kullanılır. Bu aęlar, insan beyninin iřleyiřini taklit ederek, byk veri setleri zerinde karmařık desenleri tanıma ve ęrenme yeteneklerine sahiptir. Yapay sinir aęları, birok baęlantılı ve iřlevsel birimden oluřur ve bu birimler veriyi iřlerken birbirleriyle iletiřim kurar. Bu iletiřim, veri zerinde derinlemesine ęrenme ve karmařık grevlerin gerekleřtirilmesini mmkn kılar. Yapay sinir aęları, zellikle grnt tanıma, doęal dil iřleme, oyun, ses tanıma ve tahminleme gibi birok alanda byk bařarı elde etmiřtir. Yapay sinir aęları, birok endstriyel ve bilimsel uygulama alanında devrim nitelięinde sonular saęlamıř ve yapay zekanın geliřimine byk katkıda bulunmuřtur.
- Naive Bayes: Baęımsızlık varsayımını kullanarak sınıflandırma problemlerinde kullanılan istatistiksel bir makine ęrenmesi yntemidir. Bu yntem, verilerdeki zelliklerin birbirinden baęımsız olduęunu varsayar. Naive bayes, birok uygulama alanında kullanılır. zellikle metin sınıflandırma, spam filtresi, duygu-durum analizi ve tıbbi teřhis gibi alanlarda bařarıyla kullanılır. Bu yntem, veri analizi srelerinde hızlı ve etkili bir zm sunar ve veri noktalarını farklı sınıflara atamak iin olasılıkları hesaplar. Baęımsızlık varsayımı sayesinde, ok byk veri setlerinde bile etkili ve hızlı sonular elde edebilir. Naive bayes, zellikle metin verileri zerinde alıřırken dikkate alınması gereken bir algoritmadır. Bu tr veriler zerinde sınıflandırma veya tahmin problemlerini bařarıyla zebilir. Bu nedenle, veri madencilięi ve doęal dil iřleme alanlarında yaygın olarak kullanılan bir makine ęrenme yntemidir.
- Rastgele Ormanlar: Birok karar aęacının bir araya gelerek tahminlerde bulunduęu ve sonuların bir topluluk yntemi olarak birleřtirildięi etkili bir makine ęrenme yntemidir. Bu yntem, veri analizi ve

sınıflandırma problemleri gibi birçok uygulama alanında başarılı bir şekilde kullanılır. Rastgele ormanlar, her bir karar ağacının verileri farklı şekillerde incelediği ve ardından bu ağaçların tahminlerini birleştirerek daha güvenilir sonuçlar elde etmeyi amaçlar. Her bir ağaç, veri örneklerinin bir alt kümesini kullanarak bir karar sınırı oluşturur. Bu şekilde farklı görüşlerle sonuçlara ulaşır. Bu çeşitlilik, modelin genel performansını artırırken, aynı zamanda overfitting (aşırı uyum) riskini azaltır. Rastgele ormanlar, özellikle büyük ve karmaşık veri setleri üzerinde çalışırken tercih edilen bir yöntemdir. Bu yöntem, özellik seçimi, sınıflandırma ve regresyon problemleri, görüntü tanıma, ses analizi ve daha birçok uygulama alanında başarıyla kullanılır. Ayrıca, modelin tahminlerini yaparken verilerdeki gürültüyü azaltma ve güvenilir sonuçlar elde etme yeteneği ile bilinir.

- **Derin Öğrenme:** Derin Öğrenme, çok katmanlı yapay sinir ağlarının kullanıldığı ve büyük veri setleri içindeki yüksek düzeyli özellikleri algılayabilen, karmaşık ilişkileri öğrenebilen ve çeşitli görevleri gerçekleştirebilen bir makine öğrenme yaklaşımıdır. Bu yaklaşım, verilerin içerdiği karmaşıklığı ele almak için birçok katman ve karmaşıklık düzeyi kullanarak bilgiyi özümsemeyi hedefler. Derin öğrenme, özellikle büyük veri setleri ve görsel, işitsel veya doğal dil işleme gibi alanlarda başarıyla kullanılır. Bu yöntem, özellikle karmaşık desenlerin tanınması, nesne tespiti, otomatik çeviri, yüz tanıma, ses tanıma ve daha birçok uygulama alanında büyük başarılar elde etmiştir. Derin öğrenme, çok sayıda katmanın birbirine bağlanması ve bu katmanların verileri işlemesi sayesinde, daha önce elde edilmesi zor veya imkansız olan sonuçlara ulaşmayı mümkün kılar. Bu nedenle, yapay zekâ alanında önemli bir rol oynar ve veri analizi, örüntü tanıma ve karar verme süreçlerinde büyük bir etkiye sahiptir.

Bu yöntemler, çeşitli problemlere uygun olarak kullanılarak tahminler yapılmasına yardımcı olur. Tercih edilecek yöntem, veri setinin özelliklerine ve sorunun doğasına bağlı olarak değişebilir.

2.2.2.2. Denetimsiz Öğrenme

Denetimsiz öğrenme yöntemiyle, kullanıcının sisteme müdahale etmeden sadece girdi verilerini sağladığı bir makine öğrenme yöntemidir. İşaretlemeye ihtiyaç duyulmadan, sistem otomatik olarak keşifler yapar ve veri içerisindeki ilişki ağlarını ortaya koymaya çalışır. Eğer elde sınıflandırılmış verilerden çok daha fazla etiketlenmemiş veri varsa, denetimli öğrenme ve denetimsiz öğrenme yöntemleri yetersiz kalabilir ve daha fazla veri etiketlemek için çaba harcamak gerekebilir.

Denetimsiz öğrenmede, en çok kullanılan yöntemler aşağıdaki gibidir:

- Kümeleme: Benzer özelliklere sahip veri noktalarını gruplara ayırarak veri setinin yapısını anlama amacı taşır. Bu yöntem, veri noktalarının benzerliklerine dayalı olarak gruplandığı küme veya kümeler oluşturur. Bu, verileri daha iyi anlaşılmasına ve görselleştirilmesine yardımcı olur.
- Birliktelik Kuralları: Veri setindeki öğelerin birlikte nasıl görüldüğünü inceleyerek bu öğeler arasındaki ilişkileri belirlemek için kullanılır. Özellikle market sepet analizi gibi uygulamalarda kullanılır. Bu yöntem, belirli ürünlerin bir arada satın alınma eğilimini veya kullanıcıların tercihlerini analiz etmek için faydalıdır.

Denetimsiz öğrenme yöntemleri, veri madenciliği, pazar analizi, öneri sistemleri ve daha birçok uygulama alanında kullanılır. Bu yöntemler, veri setlerindeki farklı veri analizi ve keşif amaçları için uygulanabilen temel yöntemlerdir.

2.2.2.3. Yarı Denetimli Öğrenme

Denetimli öğrenme ve denetimsiz öğrenme de elde az sayıda etiketlenmiş veri ve buna karşın daha çok etiketlenmemiş veri varsa yetersiz kalabilir. Böyle bir durumda en doğru yöntem sınırlı sayıda bulunan etiketlenmiş veri kullanarak, etiketlenmemiş veriler hakkında bilgi edinmektir. Etiketlenmemiş verileri sınıflandırmak için en uygun yöntem, etiketlenmiş verilerden bir modele dayalı olarak tahminler yapmaktır. Bu yönteme yarı denetimli öğrenme denir (Kızılkaya ve Oğuzlar, 2018).

Denetimli öğrenme ve yarı denetimli öğrenme arasındaki temel fark, etiketlenmiş veri setinin boyutudur. Denetimli öğrenmede genellikle büyük miktarda etiketlenmiş veri bulunurken, sınıflandırılacak veya tahmin edilecek veri sayısı daha azdır. Yarı denetimli öğrenme ise daha az etiketlenmiş veriyle çalışırken, tahmin edilecek veri sayısı daha fazladır (Kızılkaya ve Oğuzlar, 2018).

2.2.2.4. Takviyeli Öğrenme

Takviyeli öğrenme, hedefine ulaşmak için kendi başına karar alabilen yapay zekâ ve makine öğrenmesi alanında son derece önemli bir yöntemidir.

Bu öğrenme yönteminin temel amacı, öğrenen sistemin denediği farklı durumların hedefe ulaşma potansiyelini kontrol etmek ve tüm deneyimlerden doğru ve yanlışları hatırlamaktır.

Takviyeli öğrenme, birçok uygulama alanında büyük başarılar elde etmiştir. Örnek olarak, oyun oynama, robot kontrolü, otomatik sürüş sistemleri, finansal ticaret ve sağlık bakımı gibi alanlarda kullanılır. Özellikle derin takviyeli öğrenme, derin sinir ağları kullanarak karmaşık görevlerde yüksek performans elde etmek için geliştirilmiştir. Ancak takviyeli öğrenme de bazı zorluklarla karşı karşıyadır. Özellikle büyük veri gereksinimleri, eğitim süreçlerinin yavaşlığı gibi zorlukları bulunmaktadır. Bu nedenle, araştırmacılar ve mühendisler, daha güvenilir ve verimli takviyeli öğrenme algoritmaları geliştirmek için çalışmalar sürdürmektedirler.

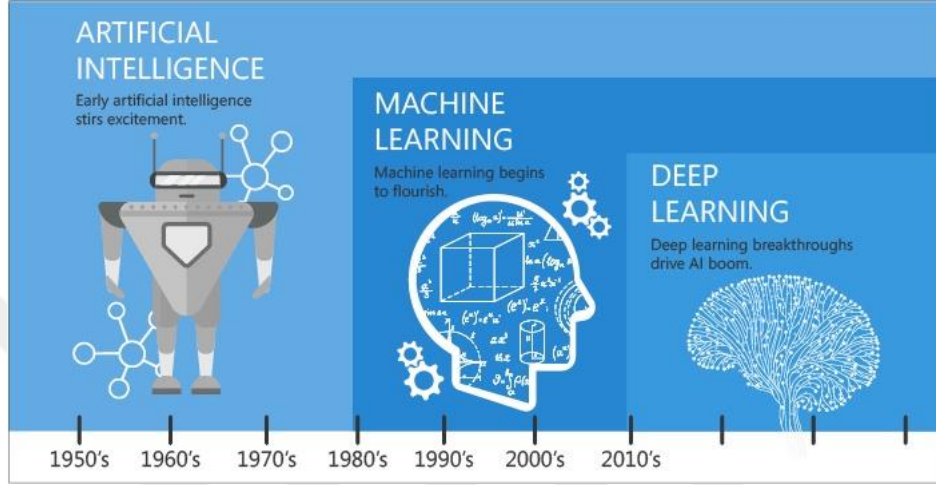
2.2.3. Derin Öğrenme ve Teknikleri

Derin Öğrenme, geleneksel makine öğrenmesinden ayrılarak, kodlanmış kurallar yerine resimler, videolar, sesler ve metinler gibi veri simgeleri üzerinden otomatik olarak öğrenme yeteneğine sahip çok katmanlı yapay sinir ağları kullanan bir yapay zekâ yöntemidir. Bu yöntem, nesne tanıma, konuşma tanıma ve doğal dil işleme gibi alanlarda kullanılmaktadır (Yılmaz ve Kaya 2021).

Derin Öğrenme, beynin yapısından esinlenerek geliştirilmiş bir makine öğrenmesi yöntemidir. Yapay sinir ağları, biyolojik nöronların işleyişini taklit ederek giriş verilerini alır. Alınan bu veriler, birbiriyle etkileşimde bulunarak analiz eder.

Sonuç çıktıları üreterek karmaşık problemleri çözmeye yönelik güçlü bir bilgi işleme yeteneği sunar (Şişmanoğlu ve ark., 2020).

Derin Öğrenme, yapay zekânın alt dalı olan makine öğrenmesinin bir bileşenidir. Şekil 3 de yapay zekadan derin öğrenmeye geline sürecin tarihsel kronolojisi belirtilmiştir.



Şekil 3. Yapay Zekanın Tarihsel Gelişim Süreci

Kaynak: *Yapay Zekâ'nın Rönesansı, Geleceği ve Etik Sorunlar*. (2017, Ekim). 2023 tarihinde Açık Radyo: <https://acikradyo.com.tr/acik-bilinc/yapay-zekanin-ronesansi-gelecegi-ve-etik-sorunlar> adresinden alındı

Derin Öğrenmede, en çok kullanılan yöntemler aşağıdaki gibidir:

- Derin Sinir Ağları
- Derin İnanç Ağları
- Derin Oto-Kodlayıcılar
- Derin Boltzmann Makinesi
- Yinelemeli Sinir Ağları
- Erişimli Sinir Ağları

Bu yöntemler, derin öğrenme alanında önemli bir rol oynamakta ve çeşitli uygulama alanlarında kullanılmaktadır.

2.2.3.1. Yapay Sinir Ağları

Yapay sinir ağları, insan beyninde bulunan nöronları taklit ederek öğrenme yeteneği kazanmayı amaçlayan bir yapay zekâ modelidir. Bu yöntem, biyolojik sinir ağlarının işleyişini matematiksel bir modelleme ile taklit eder ve büyük veri setlerinde karmaşık desenleri tanımak, öğrenmek ve tahminlerde bulunmak için kullanılır. Yapay sinir ağları, insan benzeri öğrenme ve karar verme yeteneklerini bilgisayar sistemlerine kazandırmak amacıyla geliştirilmiştir ve birçok uygulama alanında kullanılmaktadır.

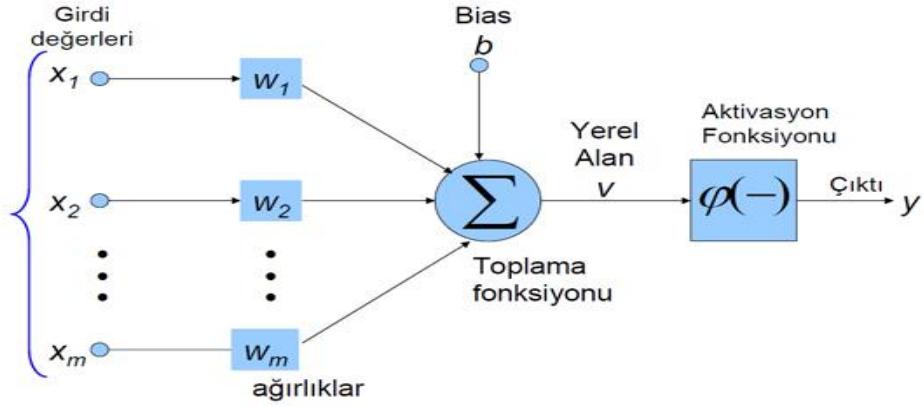
Yapay sinir ağlarında, bilgi kendi hafızasında saklanan nöronlar aracılığıyla iletişim kurar. Katmanlar arasında bulunan nöronlar ise birbirleriyle ağırlıklarıyla bağlanmaktadır (Yılmaz, 2022).

Yapay sinir ağları ile ilgili ilk model 1943 yılında Warren McCulloch ve Walter Pitts tarafından basit bir yapay sinir ağı oluşturularak ortaya konmuştur (McCulloch ve Pitts, 1943).

Kendi kendine öğrenme becerisi olan yapay sinir ağlarının avantajlarından biri, bilgileri hızlı şekilde tanımlayarak algılayabilmektedir. Ayrıca büyük kapasiteli verileri işleme, sayısal verilerin yanı sıra metin, görüntü ve sese dayalı verileri de işleyip analiz edebilir. Yapay Sinir Ağlarının çeşitli uygulama alanları ve karmaşık problemleri çözme yetenekleri geniş bir potansiyel sunar. Ancak her problem için her zaman en uygun çözüm olmayabilir ve eğitim süreçleri zaman ve kaynak gerektirebilir. Ayrıca başarılı sonuçlar elde edilmek için genellikle büyük veri setleri ve yeterli hesaplama gücü gerekebilir. Bununla birlikte, bu tür zorluklara rağmen yapay sinir ağları, veri madenciliği, görüntü işleme, doğal dil işleme, oyun gibi daha birçok alanda önemli başarılar elde eden güçlü bir araç olarak kabul edilirler.

Yapay sinir ağlarında, biyolojik sinir ağlarındaki gibi sinir hücrelerini içeren bir modeldir. Mühendislik alanında bu yapay sinir hücreleri proses elemanları olarak da adlandırılırlar.

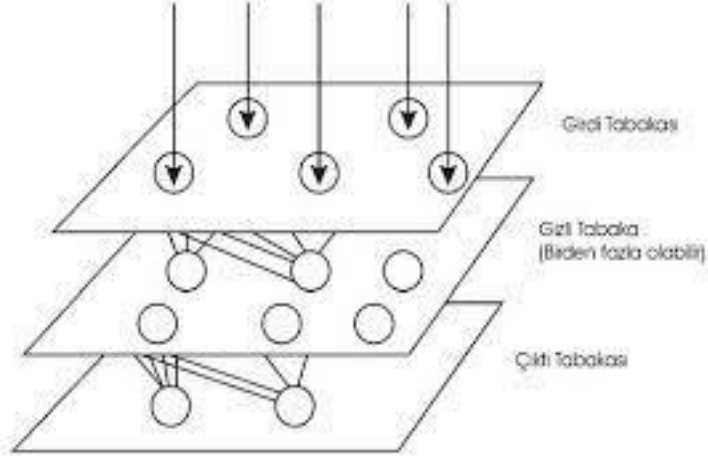
Her proses elemanı girdiler, ağırlıklar, toplama fonksiyonu, aktivasyon fonksiyonu ve çıktılar olmak üzere 5 temel elemandan meydana gelmektedir. Şekil 4 de yapay sinir hücresinin yapısı gösterilmiştir. Bu bileşenler, yapay sinir ağlarının bilgi işleme ve öğrenme yeteneklerini modellemek için kullanılır.



Şekil 4. Yapay Sinir Hücresinin Yapısı

Kaynak: *Yapay Sinir Ağları*. (2018, Ağustos). 2023 tarihinde Veri Bilimi Okulu: <https://www.veribilimiokulu.com/yapay-sinir-aglari/> adresinden alındı

Yapay sinir ağlarının yapıları ve sinir hücrelerinin sayıları farklılık gösterebilir. Bu farklılık yapay sinir ağlarının oluşması için kabul gören bir kural bulunmamaktadır. Yeterli sayıda gizli katman bulunmayan yapay sinir ağları karmaşık fonksiyonların çözümünde yetersiz kalır. Çok fazla gizli katmana sahip olan yapay sinir ağları ise istenmeyen kararsızlıklarla karşılaşabilmektedir. Gizli katmanda en verimli nöron sayısını belirlemek için herhangi bir test bulunmamaktadır (Ataseven, 2013). Şekil 5 de yapay sinir ağlarının yapısı gösterilmiştir.



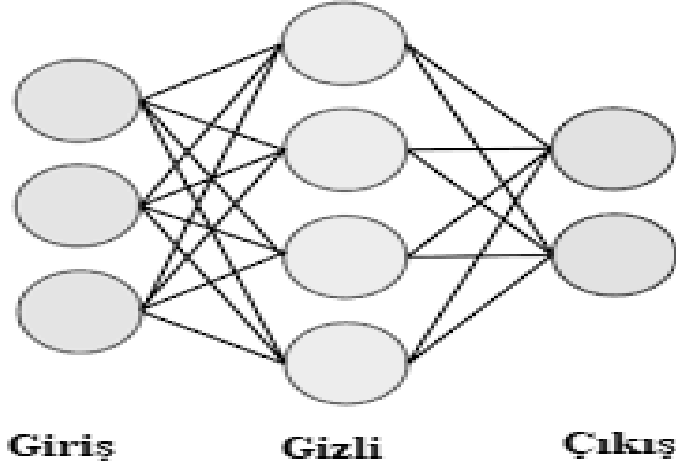
Şekil 5. Yapay Sinir Ağlarının Yapısı

Kaynak: Ataseven, B. (2013). Yapay Sinir ağları ile Öngörü Modellemesi. 10(39), 101-115

Öğrenme kabiliyeti, yapay sinir ağlarının en önemli özelliklerinden biridir. Öğrenme süreçlerini optimize etmek amacıyla birçok öğrenme algoritmaları geliştirilmiştir. Bunlar; denetimli öğrenme, denetimsiz öğrenme ve takviyeli öğrenme algoritmalarıdır. Denetimli öğrenme yöntemiyle, giriş değerleri ve onlara karşılık gelen çıktı değerleri kullanılarak sinir ağı eğitilir. Denetimsiz öğrenmede ise giriş değerleri kullanılarak sinir ağı eğitilir. (Civalek ve Ülker, 2004). Takviyeli öğrenmede ise denetimli öğrenmede gibi sisteme çok fazla detay verilmemektir. İterasyonlar sonucunda ağı elde ettiği değer kalitesi, iyi veya kötü olduğu değerlendirilip geri bildirim sağlanır (Yılmaz, 2022).

Yapay sinir ağ mimarileri, ileri beslemeli ağlar ve geri beslemeli ağlar olmak üzere iki ana başlıkta düşünülebilir. İleri beslemeli ağlarda nöronlar katmanlara ayrılmıştır. Katmanlarda bulunan nöronlar; tek yönlü bağlantılarla iletilir. Giriş katmanında bulunan nöronlar, dış ortamdan alınan bilgileri değişikliğe uğratmadan orta katmana iletir. İletilen bilgi, orta ve çıkış katmanında işlem görerek çıkışı belirlenir. Tek yönlü bağlantılarla iletilen ileri beslemeli ağlara örnek olarak çok katmanlı algılayıcılar örnek gösterilebilir. Geri beslemeli ağların en belirgin özelliği danışmansız öğrenme ile çözümlenen ağlarda kullanılır. Geri beslemeli ağlarda geri besleme işlemi yapılmaktadır. Geri beslemeli ağlar, ara katman ve çıkış katmanındaki çıkışlarını, bir önceki katmana geri beslediği ağ yapısıdır. Dinamik hafızaları bulunan

bu ağılara Hopfield, SOM, Elman ve Jordan ağıları örnek gösterilebilir (Ataseven, 2013). Şekil 6 da ileri beslemeli ağ yapısı gösterilmiştir.



Şekil 6. İleri Beslemeli Ağ Yapısı Modeli

Kaynak: Kaya, E. (2022). İleri Beslemeli Yapay Sinir Ağı'nın Eğitiminde Meta-Sezgisel Yaklaşımlar. *Bilgisayar Bilimleri ve Mühendisliği Dergisi*, 15(1), 38-43.

İleri beslemeli ağ modeli, yapay sinir ağının en yaygın kullanılan türlerinden biridir. Özellikle sınıflandırma, tahmin yapma, yorumlama gibi temel konularda kullanılmaktadır.

2.2.3.2. Derin Sinir Ağları

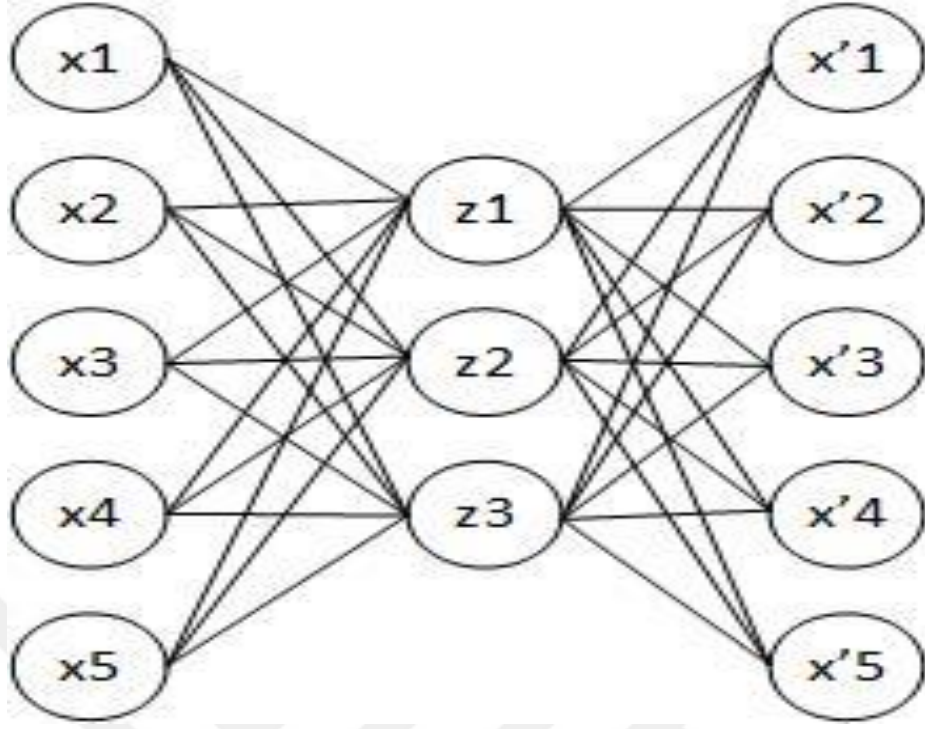
Sınıflandırma ve regresyon analizi için kullanılan derin sinir ağ yapısı, derin öğrenme makine öğrenmesinin alt dalıdır. Derin sinir ağ yapısında ikiden fazla gizli katmana sahiptir. Katmanlarda, girdi verileri işlenir ve bir sonraki katmana aktarımı sağlanır. Derin sinir ağlarının ilk giriş katmanında verileri alırken, son katmanda çıktılar üretilir. Ara katmanlar ise girdi ve çıktı arasındaki bağlantıyı temsil eder. Eğitim süreci büyük miktarda etiketli veri kullanılarak gerçekleştirildiği için birçok alanda kullanımı başarılı olmuştur (Yılmaz, 2022).

2.2.3.3. Derin Oto Kodlayıcılar

Derin oto kodlayıcılar, Geoffrey Hinton ve Ruslan Salakhutdinov tarafından 2006 yılında geliştirilmiş bir derin öğrenme yöntemidir. Bu model, özellik çıkarımı ve boyut azaltma gibi görevler için kullanılır. Derin oto kodlayıcılar, çok katmanlı algılayıcıdan oluşur ve giriş-çıkış katmanları aynı özelliklere sahip olan derin öğrenme yöntemidir. Bu modelin temel amacı, daha az parametre kullanarak daha fazla başarı eldeetmektir. Bununla birlikte, daha az parametre kullanarak veri kümesindeki doğruluk ve hız oranını hedeflenir. Derin oto kodlayıcılar, otomatik özellik çıkarımı ve boyut azaltma işlerinde etkilidir. Özellikle büyük ve karmaşık veri setlerinde kullanıldıklarında önemli avantajlar sunarlar.

Bu model, derin öğrenme alanında önemli bir gelişme olarak kabul edilir. Bilgisayar görüşü, doğal dil işleme, ses işleme ve daha birçok uygulama alanında kullanılır. Derin oto kodlayıcılar, veri analizi ve model oluşturma süreçlerinde parametre sayısını düşürerek daha etkili ve verimli sonuçlar elde etmemize yardımcı olur.

Derin oto kodlayıcılar, şifreleme ve şifreleme çözme olarak iki aşaması bulunmaktadır. Şifreleme kısmında verilen giriş bilgisi alır ve gizli katmanlar aracılığıyla girişi daha küçük boyutlu bileşenlerle temsil etmeye çalışır. Şifre çözme kısmında ise, gizli katmanlar kullanılarak şifrelenen bilgi girişi yeniden oluşturmayı çalışır (Uçar ve Uçar,2019). Şekil 7 de derin oto kodlayıcıların mimarisi gösterilmiştir.

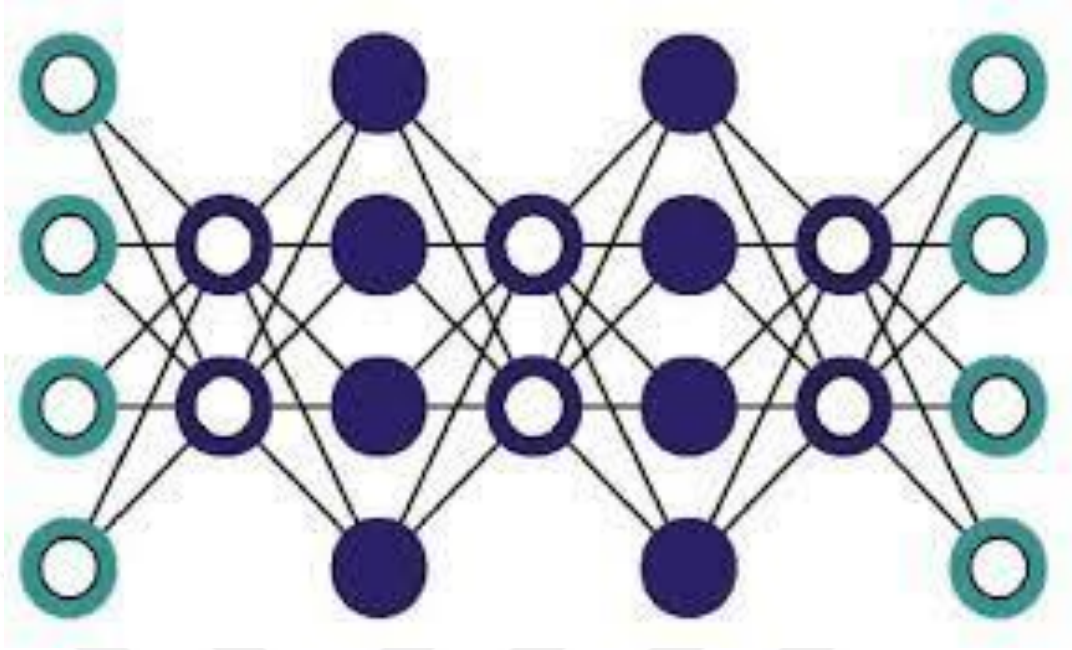


Şekil 7. Derin Oto Kodlayıcıların Mimarisi

Kaynak: Kaynar, O., Tuna, M. F., Görmez, Y., & Deveci, M. A. (2017). *C.Ü. İktisadi ve İdari Bilimler Dergisi*, 18(1), 1-14.

2.2.3.4. Derin İnanç Ağları

Hinton ve arkadaşları tarafından önerilen derin inanç ağları üretken bir yapıdır ve çok katmanlı grafiksel bir model olarak çalışır. Her katman arasında bir bağlantı bulunan derin inanç ağları, kısıtlı Boltzmann makinesinin bileşimini temsil eder. Ancak bu bağlantılar arasında birimler bulunmamaktadır. Derin inanç ağlarında en önemli özelliklerinden biri herhangi bir katmanda bulunan veri yukarıdaki katmanda bulunan verilerle bağlı olduğunu belirleyen yukarıdan aşağıya ağırlıklarını öğrenmek için katman-katman prosedürü uygulanır. Bu öğrenme süreci, verilerin temsili üzerinde hiyerarşik bir yapı oluşturur ve verileri daha yüksek seviyeli özelliklere dönüştürme yeteneği kazandırır. Şekil 8 de derin inanç ağları modeli gösterilmiştir. Bu model, birçok uygulama alanında kullanılan ve özellikle özellik çıkarımı ve desen tanıma gibi görevlerde etkili olan güçlü bir derin öğrenme yaklaşımını temsil eder.



Şekil 8. Derin İnanç Ağı Modeli

Kaynak: Doğan, F., & Türkoğlu, İ. (2019). Derin Öğrenme Modelleri ve Uygulama Alanlarına İlişkin Bir Derleme. *DÜMF Mühendislik Dergisi*, 10(2), 409-445

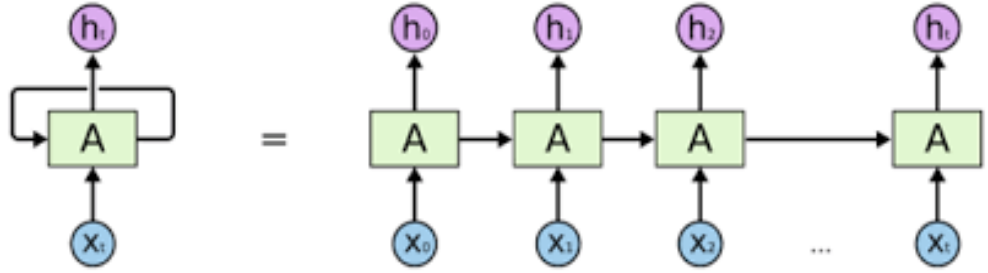
2.2.3.5. Boltzmann Makinesi

Boltzmann makinesi, Salakhutdinov ve Hinton tarafından önerilen yapay zeka ve makine öğrenmesi alanlarında önemli bir yere sahip olan bir modeldir. Bu yöntem veri örneklerini temsil etmek amacıyla ağırlık değerlerini kullanır ve enerji fonksiyonunu minimize etmek için çalışır. 1999 yılında Younes tarafından önerilen Boltzmann makinesinin ağ yapısı, belirsiz girdiler ile daha doğru sonucu elde etmek için yukarıdan aşağıya doğru geri bildirim sağlamaktadır. Boltzmann makinesi, özellikle denetimsiz öğrenme ve olasılıksal modelleme görevlerinde kullanılır. Veri setlerini temsil etmek ve veri içindeki desenleri yakalamak için kullanılır. Enerji fonksiyonunu minimize etmek için veri örneklerinin en uygun temsili bulmaya çalışırken önemlidir. Bu yöntem, büyük ve karmaşık veri setleri üzerinde kullanılabilir ve özellikle veri madenciliği ve doğal dil işleme alanlarında başarılı sonuçlar verir. Boltzmann makinesi, belirsizlik içeren veri analizini görevlerinde etkili bir araç olarak kabul edilir.

Bu model, veriler arasındaki karmaşık ilişkileri öğrenme yeteneği sayesinde özellikle büyük veri analizi ve öneri sistemleri gibi uygulamalarda kullanılmaya başlandı. Tarihçesi, makine öğrenme ve yapay zekâ alanındaki gelişmelerle birlikte sürekli evrilmeye devam etti. Boltzmann makineleri, günümüzde derin öğrenme ve yapay sinir ağları gibi teknolojilerinin temelini oluşturan kavramlar arasında yer almaktadır.

2.2.3.6. Yinelemeli Sinir Ağları

Yinelemeli sinir ağları (RNN), 1989 yılında Williams ve Zipser tarafından önerilmiş bir yapay sinir ağı yöntemidir. Bu yöntem, veri akışlarını analiz etmek amacıyla tasarlanmış bir yapay sinir ağı yapısına sahiptir. Yinelemeli sinir ağları, aynı ağırlıkları paylaşan bir yapıya sahiptir ve bu özellikle sıralı verilerin işlenmesinde güçlüdür. Bu nedenle, sıralı olayları öğrenme ve modelleme yeteneğine sahiptir. Şekil 9 da yinelemeli sinir ağları modeli gösterilmiştir. B



Şekil 9. Yinelemeli Sinir Ağı Modeli

Kaynak: <https://medium.com/@azercelikten/rnn-recurrent-neural-network-nedir-ve-nasil-calisir-d1246b1a61fb>

Yinelemeli sinir ağları (RNN), makine öğrenmesi ve yapay zekâ alanında önemli bir role sahip olan bir sinir ağı türüdür. Temelde ardışık verilerle çalışmak için tasarlanmıştır. Bu nedenle doğal dil işleme, zaman serisi analizi ve metin üretimi gibi birçok uygulama için kullanılmaktadır.

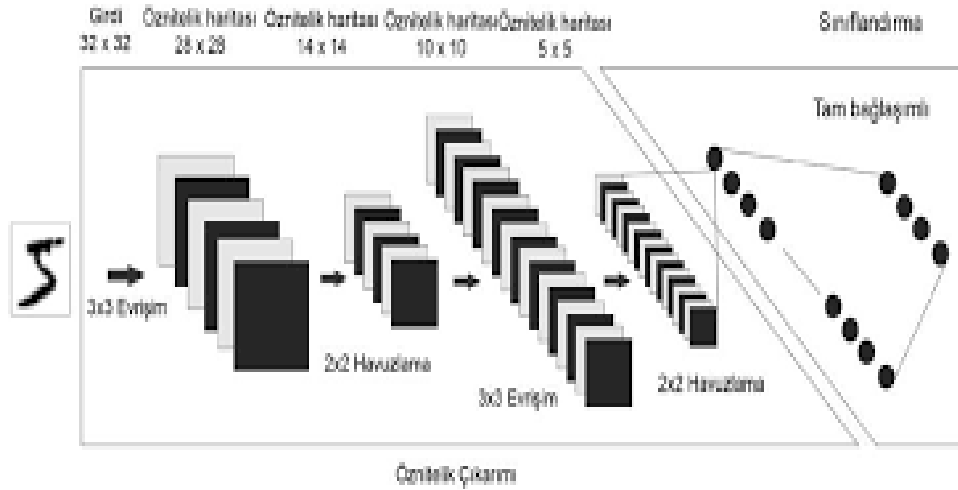
Yinelemeli sinir ağları (RNN), girdi verisinin geçmiş zaman bağlamını dikkate alabilme yeteneği ile öne çıkmaktadır. Her bir adımda önceki adımlardan gelen bilgileri içerirler ve böylece zaman içindeki ilişkileri ve desenleri öğrenebilirler. Bu özellikleri sayesinde, dil modelleri, konuşma tanıma, çeviri sistemleri ve otomatik metin üretimi gibi alanlarda başarıyla kullanılırlar. Ancak, yinelemeli sinir ağlarının

(RNN) bazı zorlukları da bulunmaktadır. Özellikle uzun bağımlılık sürelerinde, bilgi kaybı ve gradyan patlaması gibi problemlerle karşılaşabilirler. Bu nedenle, daha gelişmiş bir yinelemeli sinir ağı (RNN) varyasyonları ve alternatif yaklaşımlar, bu sorunların üstesinden gelmesi için geliştirilmiştir.

Sonuç olarak yinelemeli sinir ağları (RNN), ardışık verilerin işlenmesi gereken birçok görevde hala önemli bir rol oynamaktadır. Bu teknolojinin ilerlemesiyle, doğal dil işleme, zaman serisi analizi ve daha birçok uygulama alanında daha iyi sonuçlar elde etmek mümkün hale gelmektedir.

2.2.3.7. Evrişimli Sinir Ağları

Evrişimli sinir ağları (CNN), 1988 yılında Yenn Lecun ve arkadaşları tarafından önerilmiştir. Geleneksel yapay sinir ağı mimarilerinde bulunan sinir hücreleri diğer katmanda bulunan sinir hücreleriyle bağlantılı olurken, evrişimli sinir ağında veri sayısı ile işlem yükü geleneksel yapay sinir ağlarına göre büyük oranda azaltılmaktadır. Bu özellik ile evrişimli sinir ağları mimarisi nesne algılama, görüntü işleme, video analizi ve dil işleme gibi alanlarda başarılı sonuçlar ortaya çıkmaktadır (Çay ve ark., 2023). Şekil 10 da evrişimli sinir ağı modeli gösterilmiştir.



Şekil 10. Evrişimli Sinir Ağı Modeli

Kaynak: Kurt, F., Evrişimli Sinir Ağlarında Hiper Parametrelerin Etkisinin İncelenmesi

Evrişimli sinir ağlarının temel özelliği, veri işleme katmanları arasında özelleşmiş bir yapı kullanmalarıdır. Bu yapı, özellik çıkarma işlemi için tasarlanmıştır. Veri içindeki özellikleri belirlemek için öğrenme yeteneğine sahiptir. Bu sayede, bir görüntüdeki belirli özelliklerin tanınması veya bir metindeki önemli bilgilerin çıkarılması gibi görevler daha etkili bir şekilde gerçekleştirilebilir.

Şekil 10'da gösterilen evrişimli sinir ağı modeli, bu özel yapıları ve katmanları temsil eder. Veri işleme işlemi katmanlar arasında ilerlerken bu özellikler giderek daha soyut ve karmaşık hale gelir.

Sonuç olarak, evrişimli sinir ağları geleneksel yapay sinir ağlarına göre veri işleme yükünü azaltan ve bu nedenle bir dizi uygulamada büyük başarılar elde edilen güçlü bir mimari sunar. Bu nedenle, yenilikçi yaklaşım ile bilgisayar görüşü, dil işleme ve daha birçok alanda hızla popülerlik kazanmaktadır.

2.2.4. Uzun Kısa Süreli Bellek

Uzun kısa süreli bellek, Hochreiter ve Schmidhuber tarafından RNN mimarisinin eksikliklerini azaltmak amacıyla geliştirilmiştir. 1997'de geliştirilmiş olan bu algoritma aynı zamanda tekrarlayan sinir ağı olarak bilinmektedir. LSTM, uzun vadeli bağımlılık sorunlarının çözümü için önerilmiş bir modeldir. Bu özelliği sayesinde sıralı ve zaman serisi problemlerinde kullanılmaktadır. LSTM mimarisi 4 kapıdan oluşmaktadır; Giriş, çıkış, unutma ve hafıza kapılarıdır (Süzen, 2019).

Kapılar, yapay sinir ağlarının önemli bileşenleridir ve farklı görevleri yerine getirirler:

- **Giriş Kapısı:** Yeni bilgilerin ağa ne kadar eklenmesi gerektiğini kontrol eder. Bu kapı, ağın mevcut durumunu güncellemek ve yeni bilgilere uygun şekilde tepki vermek için kullanılır. Yeni girdilere mevcut bilgilerin birleştirilmesini sağlar.
- **Çıkış Kapısı:** Bilgilerin ne kadarının çıkış olarak kullanılacağını kontrol eder. Ağın sonuçlarını şekillendirir ve genellikle bir çıkış sinyali üretir. Bu kapı, ağın tahminlerini veya sonuçlarını düzenler.

- Unutma Kapısı: Bilginin ne kadarının unutulup silineceğini kontrol eder. Özellikle uzun süreli bağımlılıkları ve önemli bilgiyi saklama görevine sahiptir. Bellekteki gereksiz bilgileri temizler.
- Hafıza Kapısı: Sıralı problemler ve zaman serileri için kullanılan kısımdır. Önceki zaman adımlarından gelen bilgileri saklamak ve gelecekteki hesaplamalar için kullanılacak olan hafızayı oluşturmak için kullanılır.

Bu kapılar, özellikle tekrarlayan sinir ağları (RNN) gibi sıralı verileri işleyen ağlarda önemlidir. Her bir kapı, ağın öğrenme yeteneklerini ve bilgi işleme yeteneklerini şekillendirmeye yardımcı olur ve farklı görevleri başarıyla yerine getirir.

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \quad \text{Denklem 1}$$

Denklem 1’de giriş kapısı hesaplanması ifade edilmiştir. Giriş toplam değeri 0 ile 1 arasında bir değere sınırlanması yapılır. Böylece yeni bilgilerin hafızaya eklenme oranı ortaya çıkmış olur. Eğer çıkış değeri 0 ise, yeni bilgilere hafızaya eklenme işlemi gerçekleşmez ve unutulur.

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad \text{Denklem 2}$$

Denklem 2’te hücrenin önceki hücre durumuyla güncelleme sürecinde nasıl hesaplanacağı gösterilmiştir. Bu denklem, hücrenin bir önceki durumunu unutma işlemi ve yeni girişe dayalı olarak güncellemesini sağlanması belirlenir.

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \quad \text{Denklem 3}$$

Denklem 3’te hücrenin unutma kapısından önceki hücre durumu ağırlıklarıyla nasıl hesaplanacağı gösterilmiştir. Unutma vektörü 0 ile 1 aralığındaki bir değere sınırlandırmasını yapar. Bu sınırlama ile değer 1’e yaklaştıkça önceki hücre durumu daha az unutulurken, değer 0’a yaklaştıkça unutma oranı daha fazla olur.

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \quad \text{Denklem 4}$$

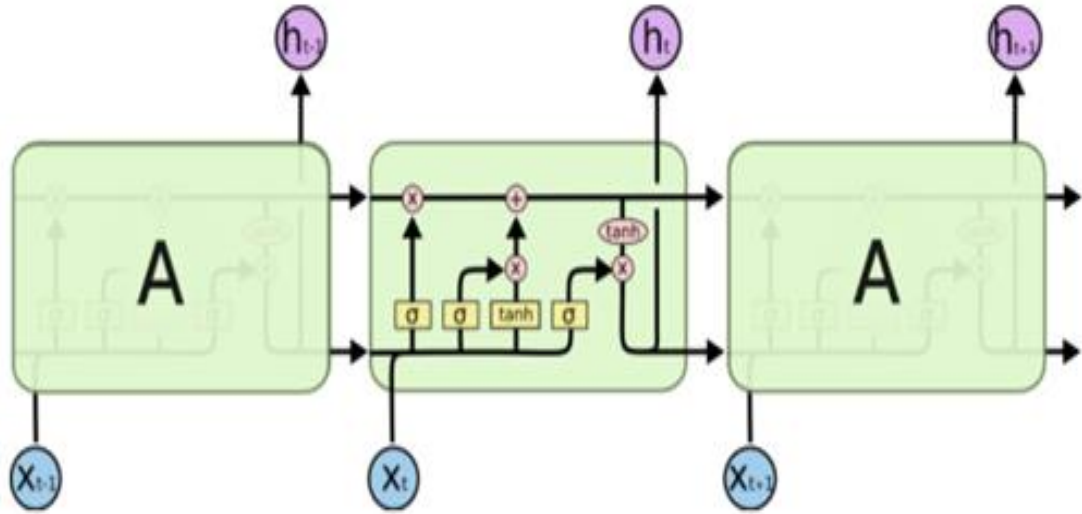
Denklem 4'te hücrenin çıkış kapısı için nasıl hesaplanacağı gösterilmiştir. Bu denklem, hücre çıkışının ne kadarını aktif bir halde kullanılacağı kontrol edilmesi belirlenmiştir. Geçerlilik vektörü 0 ile 1 aralığındaki bir değere sınırlandırmasını yapar. Bu sınırlama ile değer 1'e yaklaştıkça yeni bilgiler hafızaya daha çok eklenirken, değer 0'a yaklaştıkça daha az eklemesi yapılır.

$$h_t = o_t \odot \tanh(c_t) \quad \text{Denklem 5}$$

Denklem 5'de hücrenin çıkışı nasıl hesaplanacağı gösterilmiştir. Bu denklem, hücrenin çıkışını belirli bir şekilde ölçeklendirileceği belirlenir.

Zaman serileri tahmin edilmesi işlemlerinde sıkça kullanılan ve derin sinir ağı modeli olan LSTM, doğal dil işleme, duygu analizi, metin sınıflandırılması gibi uygulamalarında kullanılmıştır (Süzen, 2019).

Şekil 11 de LSTM ağ yapısı modeli gösterilmiştir.



Şekil 11. LSTM Modeli

Kaynak: <https://ishakdolek.medium.com/lstm-d2c281b92aac>

2.3. Veri Seti Hazırlığı

Veri seti hazırlığı, uygulamanın sürecinde en önemli aşamalarından biridir. Kullanılacak olan veriler, uygulamanın başarısını ve modelin etkinliğini büyük ölçüde etkiler. Bu nedenle, başarılı sonuçlar elde etmek için uygun ve kaliteli verilerin seçilmesi önemlidir. İyi bir veri seti, modelin doğru öğrenmesini ve uygulamanın amacına uygun şekilde çalışmasını sağlar.

Veri seti seçimi, verilerin temizlenmesi ve veri işleme gibi aşamalarını içerir. Verilerin kalitesini artırmak ve gürültüyü azaltmak için bu işlemler yapılır. Ayrıca, veri setinin boyutu, çeşitliliği ve temsiliyeti de önemlidir. Doğru ve çeşitli verilere sahip olmak, modelin genelleme yeteneğini artırır. Sonuç olarak, veri seti hazırlığı, bir uygulamanın başarısını önemli ölçüde etkileyen kritik bir adımdır. Doğru ve uygun verilerle çalışmak, başarılı sonuçlar elde etmek için temel bir gerekliliktir.

Bu tez çalışması kapsamında, açık kaynak kodlu platform olan kaggle (<https://www.kaggle.com>) üzerinden müşteri bilgileri verilen veri seti üzerinden alınmıştır. Veri seti içerisinde; fatura numarası, müşteri numarası, cinsiyet, yaş, kategori, adet, ücret, ödeme yöntemi, fatura tarihi, satın alınan yer bilgileri bulunmaktadır. Tablo 1’de veri seti özellikleri belirtilmiştir.

Tablo 1. Veri Seti Özellikleri

Fatura Numarası	Müşteri gelir bilgileri belirtilmiştir
Müşteri Numarası	Ödeme yapan müşterinin 7 haneli birbirinden farklı müşteri numarası belirtilmiştir.
Cinsiyet	K/E şeklinde ödeme yapan müşterilerin cinsiyetleri belirtilmiştir.
Yaş	Ödeme yapan müşterilerin yaş bilgileri GG/AA/YYYY formatında alınarak toplamları hesaplama ile veri setine eklenmiştir.
Kategori	Alınan ürün kategorisi belirtilmiştir.
Adet	Alınan ürün miktarı belirtilmiştir.
Ücret	Alınan ürünün ücreti TL cinsinden belirtilmiştir.
Ödeme Yöntemi	Banka kartı, nakit ve kredi olmak üzere toplam 3 adet ödeme yöntemi belirtilmiştir.
Fatura Tarihi	Ödeme yapılan tarih bilgisi GG/AA/YYYY şeklinde belirtilmiştir.
Satın Alınan Yer	Alınan ürünün nereden alındığının bilgisi verilmiştir.

Tablo 1’de incelendiğinde veri seti içerisinde 10 öznitelik bulunmaktadır. Bu öznitelik için 10000 adet veri bulunmaktadır. Veri seti virgülle ayrılmış format olan UTF-8 ile alınmıştır.

Bu tez çalışması kapsamında, makine öğrenmesin altkütmesi olan derin öğrenme yöntemleri ile müşteri davranışlarının gelecek dönemlerde eğilimlerinin tahminlemesi yapılmıştır. Elde edilen sonuçların başarı oranları karşılaştırılmıştır.

2.4. Araştırma Yöntemleri

2.4.1. Uygulama Hazırlık Süreci

Bu tez çalışması kapsamında, yapılan işlemler sırasıyla gerçekleştirilip ve detaylandırılarak anlatılmıştır.

Çalışmanın ilk aşaması, veri toplama sürecini içerir. Bu aşamada, kullanılacak olan verinin temini ve ayrıntılarının detaylı bir şekilde incelenmesi gerçekleştirilir. Veri kaynaklarının güvenilirliği ve veri kalitesi önemli bir odak noktasıdır. Veri setinin bulunmasının ardından, ikinci aşama veri setinin makine öğrenimi algoritmaları için analiz edilmesini içerir. Bu aşamada veri seti üzerinde istatistiksel analizler, veri görselleştirmeleri gibi işlemler gerçekleştirilir. Veri setinin kullanılabilir hale gelmesi ve makine öğrenimi modelleri için uygun hale getirilmesi bu aşamanın hedefidir. Son aşama ise, bu hazırlanan veri seti üzerinde farklı makine öğrenimi algoritmalarının uygulanması ve bu algoritmaların performansının değerlendirilmesidir. Bu aşamada başarı oranları gözlenir, modelin doğruluğu ve performansı değerlendirilir. Bu süreçler, veri odaklı bir çalışmanın temel adımlarını oluşturur ve sonuçlarının güvenilir ve etkili olmasını sağlar.

Uygulama ve kodlama süreci için Python 3.10 versiyonu seçilmiştir. Kodlama ortamı Spyder 5.4.1 kullanılmıştır. Pandas kütüphanesi kullanılarak veri setinin okuması yapılmıştır ve sisteme aktarılması gerçekleşmiştir. Çalışma macOS Ventura 13.2.1 işletime sahip bilgisayarda yapılmıştır.

Veri seti, virgülle ayrılmış format (UTF-8) haliyle başarılı bir şekilde okuma işlemine tabi tutulmuştur. Veri setinin okunmasının ardından eksik veriler titizlikle kontrol edilmiş ve eksik değerler, ortalama değerlerle doldurularak veri setinin eksiksiz hale getirilmesi sağlanmıştır. Makine öğrenimi algoritmalarının daha doğru ve etkili çalışabilmesi ve gürültüyü azaltmak amacıyla dönüşüm süreci uygulanmıştır.

Bu bağlamda, veri setinde bulunan metinsel veriler, özellikle cinsiyet ve müşteri numarası gibi veriler, sayısal verilere dönüştürülmüştür. Yapılan bu dönüşüm işlemi sonucunda, veri seti içerisindeki tüm veriler sayısal değerleri içerecek şekilde hazır hale getirilmiştir. Bu sayede makine öğrenimi modelleri, veri setini daha iyi anlayabilir ve daha etkili sonuçlar üretebilmesi amaçlanmıştır.

Veri setinde bulunan verilerin belirli bir aralığa ve ölçeğe uygun hale getirilmesi amacıyla veri ölçeklendirmesi işlemine ihtiyaç duyulur. Veri ölçeklendirme, modelin daha hızlı ve etkili bir şekilde eğitilmesine katkı sağlar. Ayrıca, verilerin dengeli ve tutarlı bir biçimde kullanılabilmesi için normalizasyon işlemi uygulanır. Bu çalışmada, veri ölçeklendirme işlemi için min-max normalizasyon yöntemi kullanılmıştır. Min-max normalizasyonu, en yaygın kullanılan normalizasyon yöntemlerinden biridir. Bu yöntemde, verilerin minimum değeri 0 ve maksimum değeri 1 aralığına ölçeklenir. Bu sayede verilerin model tarafından daha iyi işlenmesi ve karşılaştırılması sağlanması amaçlanmıştır.

2.4.1.1. Derin Öğrenme Yöntemlerinin Kullanılması

Bu tez çalışması kapsamında, veri setinde bulunan 10000 verinin %40'ı test süreci için kullanılmış olup, %60'lık kısmı eğitim sürecinde kullanılmıştır. Tablo 3'de veri seti dağılımı belirtilmiştir.

Tablo 2. Veri Seti Dağılımı

Test Verisi	Eğitim Verisi
4000	6000

2.4.1.2. LSTM Yönteminin Kullanılması

LSTM algoritmalar uygulanırken uygulanan parametrelerin açıklamaları aşağıda verilmiştir:

- `numeric_features`: Veri setinde kullanılacak olan sayısal sütunların adlarını içerir. Modelde giriş olarak kullanılacak olan sütunlar bu listeden seçilir.
- `test_size` ve `train_size`: `train_test_split` fonksiyonu ile veri seti eğitim ve test kümelerine bölünürken, bu parametreler eğitim ve test veri kümelerinin oranlarını belirler. Örneğin, `test_size= 0.4` ve `train_size =`

0.6 verildiğinde, veri setinin %40'ı test veri kümesi olarak ayrılır. %60'ı eğitim veri kümesi olarak kullanılır.

- `random_state`: Rastgele verilerin bölünmesi veya karıştırılması gibi işlemlerde kullanılan başlangıç durumunu belirten bir parametre. Bu parametre, aynı başlangıç durumu kullanıldığında aynı sonuçların elde edilmesini sağlar.
- `LSTM(64)`: LSTM katmanının 64 hücre içerdiğini belirtir. Bu sayı, LSTM katmanının karmaşıklığını belirler ve modelin kapasitesini etkiler. Daha fazla hücre, daha karmaşık ilişkileri modellemek için daha fazla kapasite sağlayabilir ancak aynı zamanda daha fazla hesaplama gücü gerektirebilir. LSTM hücre sayısının seçimi, modelin performansı ve genelleme yeteneği üzerinde etkili olabilir.
- `Dropout(0.2)`: Overfitting(Aşırı öğrenme)'i önlemek amacıyla eğitim sırasında düşürme uygulanır. Eğitim sırasında 0.2, %20 olasılıkla düşürme yapacağını belirtir. Bu sayede, model, aşırı öğrenmeye karşı daha dirençli hale gelir ve daha iyi genelleme yapabilir.
- `Dense(32, activation= 'relu')`: Tam bağlı bir katmanın eklenmesini temsil eder. Bu katman, 32 adet birime sahiptir ve aktivasyon fonksiyonu olarak ReLU kullanır. Bu katman, girdi verileri 32 özellik veya nörona dönüştürerek işler. ReLU aktivasyon fonksiyonu, lineer olmayanları ekleyerek ve negatif değerleri sıfır yaparak katmanın öğrenme kapasitesini artırır.
- `Dense(1,activation= 'linear')`: Bu ifade, çıkış katmanının eklenmesini ifade eder. Bu çıkış katmanı sadece 1 birime sahiptir ve aktivasyon fonksiyonu olarak doğrusal (linear) fonksiyon kullanır. Bu, regresyon problemleri için yaygın olarak kullanılan bir yapıdır. Çıkış, doğrusal bir ilişki ile hesaplanır ve çıkış değeri tahmin edilmek istenen sürekli bir değeri temsil eder.
- `model.compile(optimizer= 'adam', loss= 'mse')`: Bu ifade, modelin derleme (compile) aşamasında kullanılan optimizasyon algoritmasının "adam" ve kayıp fonksiyonunun "mean squared error" (mse) olarak belirlendiğini ifade eder. Bu konfigürasyon, modelin eğitimi ve kayıp

hesaplamalarını yönlendiren önemli parametrelerden biridir. "Adam" optimizasyon algoritması, genellikle çok kullanılan bir gradyan iniş algoritmasıdır ve kayıp fonksiyonu olarak "mse" kullanıldığında, model, ortalama kare hatayı minimize etmeye çalışır. Bu, regresyon problemleri için yaygın bir tercihtir.

Model eğitimi yapılırken kullanılan parametrelerin açıklamaları aşağıda verilmiştir:

- `model.fit(x_train, y_train, epochs=30, batch_size=32)`: Bu ifade, modelin eğitim sürecini temsil eder. Model, eğitim veri kümesi (`x_train` ve `y_train`) üzerinde 30 epok boyunca eğitilir. Her bir eğitim adımında, 32 örneklilik bir grup (batch) kullanılır (`batch_size=32`). Bu, modelin ağırlıklarını ve parametrelerini eğitmek ve verilerden öğrenmek için kullanılan eğitim sürecini tanımlar.

Model değerlendirilmesi yapılırken kullanılan parametrelerin açıklamaları aşağıda verilmiştir:

- `Model.predict(x_train)` ve `model.predict(x_test)`: Eğitim ve test verileri üzerinde tahmin yapar.
- `np.Sqrt(np.mean((y_train - y_train_pred)**2))` ve `np.sqrt(np.mean((y_test - y_test_pred)**2))`: Eğitim ve test RSME değerlerini hesaplar.

Algoritma uygulaması için gerekli olan parametreler kullanıldıktan sonra tamamlandıktan sonra model eğitime girmiştir. Eğitim sonucunda oluşan epoch çıktıları aşağıdaki gibidir.

Epoch 1/30

1875/1875 [=====] - 2s 694us/step - loss: 260.8298

Epoch 2/30

1875/1875 [=====] - 1s 681us/step - loss: 53.5897

Epoch 3/30

1875/1875 [=====] - 1s 676us/step - loss: 50.9087

Epoch 4/30

1875/1875 [=====] - 1s 673us/step - loss: 49.3865

Epoch 5/30

1875/1875 [=====] - 1s 674us/step - loss: 46.2119

Epoch 6/30

1875/1875 [=====] - 1s 673us/step - loss: 44.0809

Epoch 7/30

1875/1875 [=====] - 1s 701us/step - loss: 41.5288

Epoch 8/30

1875/1875 [=====] - 1s 679us/step - loss: 39.2694

Epoch 9/30

1875/1875 [=====] - 1s 682us/step - loss: 36.7708

Epoch 10/30

1875/1875 [=====] - 1s 681us/step - loss: 35.0906

Epoch 11/30

1875/1875 [=====] - 1s 676us/step - loss: 33.0300

Epoch 12/30

1875/1875 [=====] - 1s 678us/step - loss: 30.4681

Epoch 13/30

1875/1875 [=====] - 1s 676us/step - loss: 29.0607

Epoch 14/30

1875/1875 [=====] - 1s 676us/step - loss: 25.4961

Epoch 15/30

1875/1875 [=====] - 1s 673us/step - loss: 23.7715
Epoch 16/30
1875/1875 [=====] - 1s 674us/step - loss: 22.6887
Epoch 17/30
1875/1875 [=====] - 1s 676us/step - loss: 19.8358
Epoch 18/30
1875/1875 [=====] - 1s 676us/step - loss: 18.7768
Epoch 19/30
1875/1875 [=====] - 1s 676us/step - loss: 17.6012
Epoch 20/30
1875/1875 [=====] - 1s 712us/step - loss: 16.3745
Epoch 21/30
1875/1875 [=====] - 1s 685us/step - loss: 15.5404
Epoch 22/30
1875/1875 [=====] - 1s 680us/step - loss: 14.4485
Epoch 23/30
1875/1875 [=====] - 1s 681us/step - loss: 13.7685
Epoch 24/30
1875/1875 [=====] - 1s 679us/step - loss: 12.9008
Epoch 25/30
1875/1875 [=====] - 1s 683us/step - loss: 11.9953
Epoch 26/30
1875/1875 [=====] - 1s 675us/step - loss: 11.6870
Epoch 27/30
1875/1875 [=====] - 1s 681us/step - loss: 10.7572
Epoch 28/30
1875/1875 [=====] - 1s 682us/step - loss: 10.2578
Epoch 29/30
1875/1875 [=====] - 1s 675us/step - loss: 9.5639
Epoch 30/30
1875/1875 [=====] - 1s 676us/step - loss: 9.1401

Model çalıştığında, eğitim ve test verisinin LSTM algoritması sonuçları Tablo 4'te belirtilmiştir.

Tablo 3. LSTM Algoritması Sonuçları

	RSME
Test	0.3718
Eğitim	0.3707

2.4.1.3. RNN ve CNN Yönteminin Kullanılması

RNN ve CNN algoritmaları uygulanırken kullanılan parametreler ve açıklamaları aşağıdaki gibidir:

- `missing_values = data.isnull().sum()`: Bu ifade, eksik verilerin sayısını bulmak için kullanılır. Her sütundaki eksik değerlerin sayısını hesaplar. Bu sayede veri setinin hangi sütunlarında kaç adet eksik değer olduğu belirlenir. Bu bilgi, eksik verilerin işlenmesi veya doldurulması gerekip gerekmediğini değerlendirmek için kullanılır.
- `Data.fillna(data.mean(numeric_only = True), inplace =True)`: Bu ifade, eksik verilerin sayısal sütunların ortalamasıyla doldurulmasını sağlar. Parametre olarak `'numeric_only=True'` kullanılarak yalnızca sayısal sütunlar dikkate alınır. Eksik değerler, ilgili sütunun ortalaması ile değiştirilir ve bu işlem veri setinde yapılır (`'inplace=True'`). Bu sayede eksik verilerin sayısal sütunların ortalamalarıyla doldurulması sağlanır, veri setinin tutarlılığı korunur.
- `numeric_features = ['price', 'quantity', 'customer_id', 'age']`: Bu ifade, veri setinde kullanılacak sayısal sütunların listesini belirtir. Liste içinde yer alan sütunlar, veri analizi veya modelleme işlemlerinde kullanılacak olan sayısal değişkenleri temsil eder. Bu şekilde belirtilen sayısal sütunlar, veri setinin ilgili analiz veya modelleme süreçlerinde odak noktası haline gelir.
- `scaler = MinMaxScaler()`: Bu ifade, veri normalizasyonu yapmak için kullanılır. Verileri `[0, 1]` aralığına indirger. Min-max normalizasyonu, verilerin orijinal aralığından bağımsız olarak belirli bir aralığa ölçeklemek için kullanılır. Bu işlem, verilerin modelin daha iyi eğitilmesine ve karşılaştırılmasına yardımcı olabilir.

- `x_train_rnn = x_train.reshape((x_train.shape[0],1, x_train.shape[1]))`: Bu ifade, LSTM modeli için veriyi uygun hale getiren bir parametreyi temsil eder. Veriyi yeniden şekillendirerek, her örnek için bir zaman adımı ve sütunlar arasındaki özellikleri içerecek şekilde düzenler. Bu, zaman serisi veya sıralı verilerle çalışırken LSTM'nin giriş formatına uyum sağlar.
- `x_train_cnn = x_train.reshape((x_train.shape[0], x_train.shape[1],1))`: Bu ifade, CNN modeli için veri şeklini uygun hale getiren bir parametreyi temsil eder. Veriyi yeniden şekillendirerek, her bir örnek için sütunlar arasındaki özellikleri içerecek şekilde düzenler. Bu, görüntü işleme veya özellik haritaları kullanıldığında CNN modelinin giriş formatına uyum sağlar.
- `model_rnn.add(LSTM(64,input_shape=(x_train_rnn.shape[1],x_train_rnn.shape[2])))`: Bu parametre, LSTM modeline 64 hücreden oluşan bir LSTM katmanı ekler. Ayrıca, giriş verisinin şeklini belirler. LSTM katmanının hücre sayısı, modelin karmaşıklığını ve öğrenme kapasitesini etkileyen önemli bir parametredir.
- `model_rnn.add(Dropout(0.2))`: Overfitting önlemi için %20 dropout uygulanır.
- `model_rnn.add(Dense(32, activation='relu'))`: 32 nöronlu tam bağlı bir katman eklenir ve aktivasyon fonksiyonu olarak ReLU kullanılır.
- `model_rnn.add(Dense(1, activation='linear'))`: Çıkış katmanı olarak lineer aktivasyon kullanılır.
- `model_rnn.compile(optimizer='adam', loss='mse')`: Bu ifade, modelin derleme aşamasını temsil eder. Model, "adam" olarak adlandırılan bir en iyileştirici (optimizer) kullanır ve kayıp fonksiyonu olarak ortalama karesel hata (MSE) kullanır. Derleme aşamasında, modelin eğitim için nasıl yapılandırılacağı ve nasıl optimize edileceği belirlenir.
- `model_rnn.fit(X_train_rnn, y_train, epochs=30, batch_size=32)`: Bu ifade, modelin eğitim aşamasını temsil eder. Model, `x_train_rnn` verisi ve `y_train` etiketleri kullanılarak eğitilir. Eğitim 30 epoch boyunca devam eder ve her bir parti (batch) 32 örnek içerir. Bu, modelin öğrenme sürecini belirler ve veri üzerinde belirli bir sayıda tekrarlı eğitim dönemini ifade eder. Batch size ise her bir eğitim adımında kullanılacak veri örneklerinin sayısını belirtir.

CNN Modeli:

- `model_cnn.add(Conv1D(filters=64, kernel_size=3, activation='relu', input_shape=(X_train_cnn.shape[1], X_train_cnn.shape[2])))`: Bu ifade, modelin bir Conv1D katmanını temsil eder. Bu katman, 64 filtre içerir, her biri 3 boyutlu bir pencereyi işler ve ReLU aktivasyon fonksiyonunu kullanır. Ayrıca, giriş verisinin şeklini belirler. Conv1D katmanları, zaman serileri veya metin verileri gibi sıralı verilerle çalışırken kullanılır.
- `model_cnn.add(MaxPooling1D(pool_size=2))`: Bu ifade, modelin bir MaxPooling1D katmanını temsil eder. MaxPooling, özellik haritalarını küçültmek ve önemli özellikleri vurgulamak için kullanılır. `'pool_size=2'` parametresi, her iki zaman adımında bir en büyük değeri seçerek boyut küçültme işlemi yapar.
- `model_cnn.add(Flatten())`: Bu ifade, modelin Flatten (Düzleştirme) katmanını temsil eder. Flatten katmanı, özellik haritasını düz bir vektöre dönüştürür. Bu, Convolutional Neural Network (CNN) sonrası gelen tam bağlantılı (fully connected) katmanlara geçiş yaparken kullanılır.

Algoritmalar uygulaması için gerekli olan parametreler kullanıldıktan sonra model eğitime girmiştir. Eğitim sonucunda oluşan epoch çıktıları aşağıdaki gibidir.

Epoch 1/30

1875/1875 [=====] - 2s 679us/step - loss: 198.9387

Epoch 2/30

1875/1875 [=====] - 1s 670us/step - loss: 37.3745

Epoch 3/30

1875/1875 [=====] - 1s 669us/step - loss: 36.8778

Epoch 4/30

1875/1875 [=====] - 1s 668us/step - loss: 34.6686

Epoch 5/30

1875/1875 [=====] - 1s 683us/step - loss: 32.9712

Epoch 6/30

1875/1875 [=====] - 1s 671us/step - loss: 30.9220

Epoch 7/30

1875/1875 [=====] - 1s 677us/step - loss: 29.3534
Epoch 8/30
1875/1875 [=====] - 1s 675us/step - loss: 27.2874
Epoch 9/30
1875/1875 [=====] - 1s 691us/step - loss: 25.7879
Epoch 10/30
1875/1875 [=====] - 1s 670us/step - loss: 24.6391
Epoch 11/30
1875/1875 [=====] - 1s 665us/step - loss: 22.0851
Epoch 12/30
1875/1875 [=====] - 1s 666us/step - loss: 20.6523
Epoch 13/30
1875/1875 [=====] - 1s 679us/step - loss: 19.7407
Epoch 14/30
1875/1875 [=====] - 1s 663us/step - loss: 17.7624
Epoch 15/30
1875/1875 [=====] - 1s 662us/step - loss: 17.0717
Epoch 16/30
1875/1875 [=====] - 1s 654us/step - loss: 16.0940
Epoch 17/30
1875/1875 [=====] - 1s 653us/step - loss: 15.3264
Epoch 18/30
1875/1875 [=====] - 1s 645us/step - loss: 14.6477
Epoch 19/30
1875/1875 [=====] - 1s 653us/step - loss: 13.7571
Epoch 20/30
1875/1875 [=====] - 1s 682us/step - loss: 13.3565
Epoch 21/30
1875/1875 [=====] - 1s 652us/step - loss: 12.7108
Epoch 22/30
1875/1875 [=====] - 1s 651us/step - loss: 12.1975

Epoch 23/30

1875/1875 [=====] - 1s 651us/step - loss: 11.5898

Epoch 24/30

1875/1875 [=====] - 1s 654us/step - loss: 11.0877

Epoch 25/30

1875/1875 [=====] - 1s 658us/step - loss: 10.6179

Epoch 26/30

1875/1875 [=====] - 1s 658us/step - loss: 10.3118

Epoch 27/30

1875/1875 [=====] - 1s 666us/step - loss: 9.8090

Epoch 28/30

1875/1875 [=====] - 1s 670us/step - loss: 9.4424

Epoch 29/30

1875/1875 [=====] - 1s 670us/step - loss: 8.9995

Epoch 30/30

1875/1875 [=====] - 1s 664us/step - loss: 8.6984

Algoritmaların eğitim süreci tamamlandıktan sonra RSME sonuçları Tablo 5 ve Tablo 6 da belirtilmiştir.

Tablo 4. RNN Algoritması Sonuçları

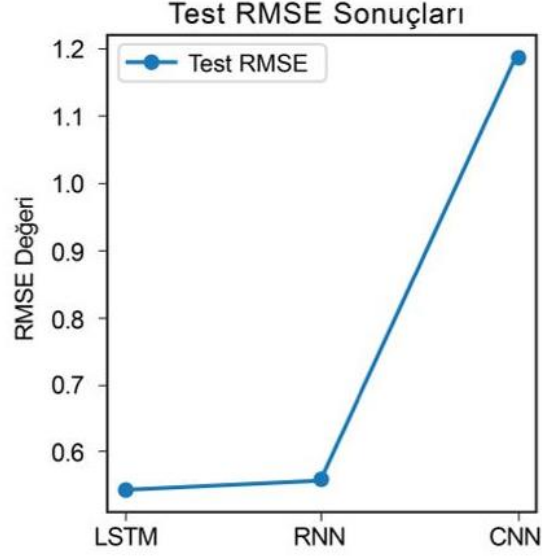
	RSME
Test	0.6981
Eğitim	0.6949

Tablo 5. CNN Algoritması Sonuçları

	RSME
Test	1.041
Eğitim	1.038

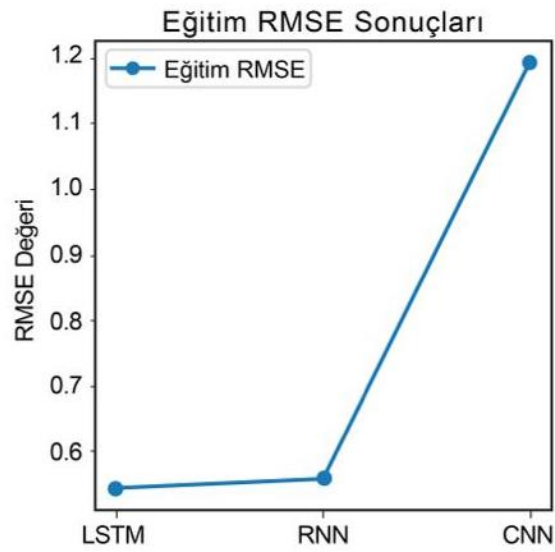
2.4.2. Araştırma Sonuçları

Şekil 12’de LSTM, RNN ve CNN algoritmalarının test RMSE skorları karşılaştırılması belirtilmiştir.



Şekil 12. LSTM, RNN ve CNN Yöntemlerinin Test Skorları Karşılaştırması

Şekil 13’de LSTM, RNN ve CNN algoritmalarının eğitim RMSE skorları karşılaştırılması belirtilmiştir.



Şekil 13. LSTM, RNN ve CNN Yöntemlerinin Eğitim Skorları Karşılaştırması

SONUÇ

Bu tez çalışması müşteri davranışının tahmin edilmesi alanında makine öğrenmesi yöntemleriyle tahmin edilmesi önerilmiştir. Müşteri davranışı, işletmeler için kritik bir öneme sahiptir. İşletmelerin iş stratejilerinin oluşturulması ve pazarlama çabalarının yönlendirilmesi için temel bilgiler sunar. Bu çalışma, makine öğrenimi yöntemlerini kullanarak bu sorunu çözmeye yönelik bir adım olarak görülmektedir. Çalışmada farklı derin öğrenme yöntemleri, LSTM, RNN ve CNN algoritmaları kullanılarak müşteri davranışı tahminlemesi üzerindeki etkilerini incelemeyi amaçlayarak gerçekleştirilmiştir. Önceki bölümlerde sunulan sonuçlar ve analizler temel alınarak, bu bölümde elde edilen sonuçların özetlenmesi ve gelecekteki araştırmalara yönelik öneriler sunulacaktır. Çalışmada kullanılan veriler açık kaynak kodlu bir platformdan alınmış olup, 6000 adedi eğitime, 4000 adedi ise test için ayrılmıştır. RNN modeli için veri şekli, her örneğin bir zaman adımını temsil etmesi amacıyla yeniden düzenlenmiştir. CNN modeli için ise veri şekli, evrişim işlemlerine uygun olarak yeniden yapılandırılmıştır.

Sonuçlar incelendiğinde, LSTM algoritmasının hem eğitim hem de test aşamalarında daha başarılı olduğu gözlemlenmiştir. Eğitim aşamasında elde edilen en düşük değer, algoritmanın modelin verilere daha iyi uyum sağladığını göstermiştir. Ayrıca test aşamasında da düşük değeri elde etmesi, modelin genelleme yeteneğinin yüksek olduğunu göstermektedir. Böylece elde edilen sonuçlar literatürde yer alan çalışmalarda örtüştüğünü göstermiştir.

RNN ve CNN algoritmaları uygulaması sonucunda müşteri tahmininde kabul edilebilir sonuçlar vermiştir. Ancak, elde edilen değerlerin LSTM algoritmasına kıyasla daha yüksek olduğun görülmüştür. Bu durum, RNN ve CNN algoritmalarının verilere daha az uyum sağladığını ve genelleme yeteneklerinin sınırlı olduğunu göstermiştir.

LSTM yönteminin, RNN ve CNN yöntemlerine göre daha başarılı olduğu görülmüştür. LSTM yöntemi veri sayısı artmasıyla daha başarılı olduğu ortaya çıkmıştır. Ayrıca bundan sonraki çalışmalarda veri seti hazırlanırken daha büyük veri setleriyle çalışma değerlendirilebilir. Ayrıca daha yüksek kaliteli veri setleri

kullanarak deneyler gerekleřtirilmesi, algoritmaların performansını daha ayrıntılı bir řekilde deęerlendirilebilir. Eksik veri noktalarının daha etkili bir řekilde ele alınması da nemlidir. Daha geniř ve eřitli veri setlerinin kullanılması, algoritmaların genelleme yeteneklerini daha iyi anlamak iin faydalı olacaktır.

Sonu olarak bu alıřma farklı derin ęrenme algoritmalarının mřteri davranıřlarının yař tahmini problemindeki performansını karřılařtırarak nemli bilgiler sunmuřtur. Bu alıřma sonuları, gelecekte benzer tahmin problemleri zerine daha fazla arařtırma yapılmasına olanak saęlayabilir.



KAYNAKÇA

- Akyiğit, H. E., & Taşçı, T. (2022). Sigortacılık Sektöründe Makine Öğrenmesi İle Müşteri Kaybı Analizi. *Tasarım, Mimarlık ve Mühendislik Dergisi*, 2, 66-79.
- Atalay, M., & Çelik, E. (2017). Büyük Veri Analizinde Yapay Zekâ ve Makine Öğrenmesi Uygulamaları. *Mehmet Akif Ersoy Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, 9(22), 155-172.
- Ataseven, B. (2013). Yapay Sinir ağları ile Öngörü Modellemesi. *10*(39), 101-115.
- Aydın, S., & Özkul, A. (2015, Ağustos). Veri Madenciliği ve Anadolu Üniversitesi Açıköğretim Sisteminde Bir Uygulama. *Eğitim ve Öğretim Araştırmaları Dergisi*, 4(3), 36-44.
- Civalek, Ö., & Ülker, M. (2004). MO Teknik Dergi, 2004 3171-3190, Yazı 213 Dikdörtgen Plakların Doğrusal Olmayan Analizinde Yapay Sinir Ağı Yaklaşımı1. *İMO Teknik Dergi*, 15(72), 3171-3190.
- Çay, T., Ölmez, E., & Er, O. (2023). Bölgesel Tabanlı Evrişimli Sinir Ağı ile Araç Plaka Tanıma. *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, 11(1), 10-20.
- Çağlayan Akay, E. (2018). Ekonometride Yeni Bir Ufuk: Büyük Veri ve Makine Öğrenmesi. *Social Sciences Research Journal*, 7(2), 2147-5237.
- Doğan, F., & Türkoğlu, İ. (2019). Derin Öğrenme Modelleri ve Uygulama Alanlarına İlişkin Bir Derleme. *DÜMF Mühendislik Dergisi*, 10(2), 409-445.
- Ercan, U. (2022). Havayolu Taşımacılığında Müşteri Memnuniyetinin Topluluk Öğrenmesi Yöntemleri ile Belirlenmesi. *Alanya Akademik Bakış Dergisi*, 2763-2774.
- Göy, G., Kolukısa, B., Bahçevan, C., & Güngör, V. (2020). Makine Öğrenmesi Teknikleri ile İnternet Servis Sağlayıcısı için Müşteri Kayıp Tahmini. *5th International Conference on Computer Science and Engineering (UBMK)* (s. 248-253). IEEE.
- Kaya, E. (2022). İleri Beslemeli Yapay Sinir Ağıının Eğitiminde Meta-Sezgisel Yaklaşımlar. *Bilgisayar Bilimleri ve Mühendisliği Dergisi*, 15(1), 38-43.

- Kaynar, O., Tuna, M. F., Görmez, Y., & Deveci, M. A. (2017). *C.Ü. İktisadi ve İdari Bilimler Dergisi*, 18(1), 1-14.
- Koruyan, K., & Ekeryılmaz, A. (tarih yok). Makine Öğrenmesi ile Müşteri Şikayetlerinin Sınıflandırılması. *AJIT-e: Academic Journal of Information Technology*, 13(50), 168-183.
- Makine Öğrenmesi(Machine Learning)Nedir?* (2021, Kasım 26). turhost.com: <https://www.turhost.com/blog/makine-ogrenmesi-machine-learning-nedir/#serp> adresinden alındı
- McCulloch, W. S., & Pitts, W. A. (1943). A logical calculus of the ideas immanent in nervous activity. *Buttetin of Mathematics and Biophysics*. 115-133.
- Süzen, A. A. (2019). LSTM Derin Sinir Ağları ile Üniversite Giriş Sınavındaki Matematik Soru Sayılarının Konulara Göre Tahmini. *Engineering Sciences*, 14(3), 112-118.
- Şeker, E. (2020). Yapay Zeka Tekniklerinin/ Uygulamalarının Siber Savunmada Kullanımı. *Uluslararası Bilgi Güvenliği mühendisliği Dergisi*, 6(2), 108-115.
- Şişmanoğlu, G., Koçer, F., Önde, M. A., & Şahingöz, Ö. K. (2020). Derin Öğrenme Yöntemleri ile Borsada Fiyat Tahmini. *BEÜ Fen Bilimleri Dergisi*, 9(1), 434-445.
- Uçar, M., & Uçar, E. (2019). Derin Otomatik Kodlayıcı Tabanlı Özellik Çıkarımı ile Android Kötücül Yazılım Uygulamalarının Tespiti. *Ybs Dergisi*, 5(2), 21-28.
- Uyanık, F., & Kasapbaşı, M. C. (2021). Telekomünikasyon Sektörü için Veri Madenciliği ve Makine Öğrenmesi Teknikleri ile Ayrılan Müşteri Analizi. *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, 9(3), 172-191.
- Yılmaz, A. (2022). *Yapay Zeka*. İstanbul: Kodlab.
- Yılmaz, A., & Kaya, U. (tarih yok). *Derin Öğrenme*. İstanbul: Kodlab.
- Yaman, Y. (2021). Müşteri Yolculuğu Analitiğini Makine Öğrenmesi Modeliyle Tahminleme. *YBS Ansiklopedisi*, 9(1), 29-40.
- Yapay Sinir Ağları*. (2018, Ağustos). 2023 tarihinde Veri Bilimi Okulu: <https://www.veribilimiokulu.com/yapay-sinir-aglari/> adresinden alındı

Yapay Zekâ'nın Rönesansı, Geleceđi ve Etik Sorunlar. (2017, Ekim). 2023 tarihinde
Açık Radyo: <https://acikradyo.com.tr/acik-bilinc/yapay-zekanin-ronesansi-gelecegi-ve-etik-sorunlar> adresinden alındı



EKLER

EK 1. LSTM Uygulama Kodları

```
import pandas as pd

import numpy as np

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import MinMaxScaler

from keras.models import Sequential

from keras.layers import LSTM, Dense, Dropout

data = pd.read_excel('/Users/merve/Desktop/testveriseti.xlsx')

# Veri Ön İşleme Adımları

# Eksik veri sayısını bulma aşaması

missing_values = data.isnull().sum()

# Eksik verileri doldurma aşaması

data.fillna(data.mean(numeric_only=True), inplace=True)

# Diğer sayısal sütunları seçme aşaması

numeric_features = ['price', 'quantity', 'customer_id', 'age']

# Veri setinde yalnızca sayısal sütunları kullanma aşaması

X = data[numeric_features]

y = data['age']
```

```
# Veri Normalizasyonu aşaması

scaler = MinMaxScaler()

X = scaler.fit_transform(X)


# Eğitim ve test veri kümelerini ayırma aşaması

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4,
train_size=0.6, random_state=42)


# Veri şeklini LSTM modeline uygun hale getirme (özellikle zaman serisi
veriler için gereklidir)

X_train = X_train.reshape((X_train.shape[0], 1, X_train.shape[1]))
X_test = X_test.reshape((X_test.shape[0], 1, X_test.shape[1]))


# Modeli oluşturma aşaması

model = Sequential()

model.add(LSTM(64, input_shape=(X_train.shape[1], X_train.shape[2])))

model.add(Dropout(0.2))

model.add(Dense(32, activation='relu'))

model.add(Dropout(0.2))

model.add(Dense(1, activation='linear'))


# Modeli derleme aşaması

model.compile(optimizer='adam', loss='mse') # 'mean_squared_error' yerine
'mse' kullanın
```

```
# Modeli eğitme aşaması

model.fit(X_train, y_train, epochs=30, batch_size=32)

# Eğitim ve test RMSE hesaplama aşaması

y_train_pred = model.predict(X_train).ravel()
y_test_pred = model.predict(X_test).ravel()

train_rmse = np.sqrt(np.mean((y_train - y_train_pred)**2))
test_rmse = np.sqrt(np.mean((y_test - y_test_pred)**2))

print("Eğitim RMSE:", train_rmse)
print("Test RMSE:", test_rmse)
```

EK 2. RNN ve CNN Uygulama Kodları

```
import pandas as pd

import numpy as np

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import MinMaxScaler

from keras.models import Sequential

from keras.layers import LSTM, Dense, Dropout, Conv1D,
MaxPooling1D, Flatten

data = pd.read_excel('/Users/merve/Desktop/testveriseti.xlsx')

# Veri Ön İşleme Adımları aşaması

# Eksik veri sayısını bulma

missing_values = data.isnull().sum()

# Eksik verileri doldurma aşaması

data.fillna(data.mean(numeric_only=True), inplace=True)

# Diğer sayısal sütunları seçme

numeric_features = ['price', 'quantity', 'customer_id', 'age']

# Veri setinde yalnızca sayısal sütunları kullanma aşaması

X = data[numeric_features]

y = data['age']
```

```

# Veri Normalizasyonu aşaması

scaler = MinMaxScaler()

X = scaler.fit_transform(X)


# Eğitim ve test veri kümelerini ayırma aşaması

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4,
train_size=0.6, random_state=42)


# Veri şeklini RNN modeline uygun hale getirme aşaması

X_train_rnn = X_train.reshape((X_train.shape[0], 1,
X_train.shape[1]))

X_test_rnn = X_test.reshape((X_test.shape[0], 1, X_test.shape[1]))


# Veri şeklini CNN modeline uygun hale getirme aşaması

X_train_cnn = X_train.reshape((X_train.shape[0], X_train.shape[1],
1))

X_test_cnn = X_test.reshape((X_test.shape[0], X_test.shape[1], 1))


# Modeli oluşturma (RNN) aşaması

model_rnn = Sequential()

model_rnn.add(LSTM(64, input_shape=(X_train_rnn.shape[1],
X_train_rnn.shape[2])))

model_rnn.add(Dropout(0.2))

model_rnn.add(Dense(32, activation='relu'))

model_rnn.add(Dropout(0.2))

model_rnn.add(Dense(1, activation='linear'))

```

```
# Modeli derleme aşaması
```

```
model_rnn.compile(optimizer='adam', loss='mse')
```

```
# Modeli eğitme aşaması
```

```
model_rnn.fit(X_train_rnn, y_train, epochs=30, batch_size=32)
```

```
# Modeli oluşturma (CNN) aşaması
```

```
model_cnn = Sequential()
```

```
model_cnn.add(Conv1D(filters=64, kernel_size=3, activation='relu',  
input_shape=(X_train_cnn.shape[1], X_train_cnn.shape[2])))
```

```
model_cnn.add(MaxPooling1D(pool_size=2))
```

```
model_cnn.add(Flatten())
```

```
model_cnn.add(Dense(32, activation='relu'))
```

```
model_cnn.add(Dropout(0.2))
```

```
model_cnn.add(Dense(1, activation='linear'))
```

```
# Modeli derleme aşaması
```

```
model_cnn.compile(optimizer='adam', loss='mse')
```

```
# Modeli eğitme
```

```
model_cnn.fit(X_train_cnn, y_train, epochs=30, batch_size=32)
```

```
# Eğitim ve test RMSE hesaplama
```

```
y_train_pred_rnn = model_rnn.predict(X_train_rnn).ravel()
```

```
y_test_pred_rnn = model_rnn.predict(X_test_rnn).ravel()
```

```
train_rmse_rnn = np.sqrt(np.mean((y_train - y_train_pred_rnn)**2))
test_rmse_rnn = np.sqrt(np.mean((y_test - y_test_pred_rnn)**2))
```

```
print("Eğitim RMSE (RNN):", train_rmse_rnn)
print("Test RMSE (RNN):", test_rmse_rnn)
```

```
y_train_pred_cnn = model_cnn.predict(X_train_cnn).ravel()
y_test_pred_cnn = model_cnn.predict(X_test_cnn).ravel()
```

```
train_rmse_cnn = np.sqrt(np.mean((y_train - y_train_pred_cnn)**2))
test_rmse_cnn = np.sqrt(np.mean((y_test - y_test_pred_cnn)**2))
```

```
print("Eğitim RMSE (CNN):", train_rmse_cnn)
print("Test RMSE (CNN):", test_rmse_cnn)
```

