



# Konut Satış Fiyatlarının Regresyon Tabanlı Makine Öğrenmesi Yöntemleri Kullanılarak Belirlenmesi

Yazılım Mühendisliği Ana Bilim Dalı

Dönem Projesi

Uğur Kılıç

ORCID 0000-0001-9493-8953

Proje Danışmanı: Prof. Dr. Aytuğ Onan

Ocak 2024

# Konut Satış Fiyatlarının Regresyon Tabanlı Makine Öğrenmesi Yöntemleri Kullanılarak Belirlenmesi

## ÖZ

Bu proje, yapay zeka ve sinir ağlarına vurgu yaparak regresyon tabanlı makine öğrenimi yöntemleriyle konut satış fiyatlarının tahmin edilmesine ilişkin derinlemesine bir analiz sunmaktadır. Ev satış fiyatlarını etkili bir şekilde tahmin eden kapsamlı bir model geliştirmek için yapay zekada uzman sistemlerin uygulanmasını araştırmaktadır.

Çalışma, verileri analiz etmek ve yorumlamak için gelişmiş makine öğrenimi tekniklerini uygulayarak Ames, Iowa'daki ev özelliklerinden oluşan sağlam bir veri kümesi kullanmaktadır. Bu yaklaşım, konut fiyatlandırma dinamiklerine ilişkin önemli içgörüler sunarak gayrimenkul değerlemesinin anlaşılmasına katkıda bulunuyor. Bulgular, yapay sinir ağlarının gayrimenkul fiyatlarını doğru bir şekilde tahmin etme potansiyelini ortaya koyarak sektör profesyonelleri ve araştırmacılar için değerli araçlar sağlamaktadır.

**Anahtar Sözcükler:** Yapay Zeka, Yapay Sinir Ağları, Konut Satış Fiyatları, Regresyon, Uzman Sistemler

# Determination of House Sale Prices Using Regression-Based Machine Learning Methods

## Abstract

This project presents an in-depth analysis of predicting house sale prices through regression-based machine learning methods, emphasizing artificial intelligence and neural networks. It explores the implementation of expert systems in artificial intelligence to develop a comprehensive model that effectively predicts house sale prices.

The study utilizes a robust dataset of house attributes from Ames, Iowa, applying advanced machine learning techniques to analyze and interpret the data. This approach offers significant insights into the dynamics of house pricing, contributing to the understanding of real estate valuation. The findings demonstrate the potential of artificial neural networks in accurately forecasting real estate prices, providing valuable tools for industry professionals and researchers.

**Keywords:** Artificial Intelligence, Artificial Neural Networks, House Sale Prices, Regression, Expert Systems

*Sevgili Ođlum Bulut,*

*Bu proje azmin, merakın ve durmaksızın bilgi peşinde koşmanın gücünün bir kanıtıdır. Kendi eğitim yolculuđuna çıkarken, bu çalışmanın her zorluđun öğrenmek ve büyümek için bir fırsat olduđunu hatırlatmasına izin ver. Her zaman sorgulama kıvılcımını ve yeni ufuklar keşfetme cesaretini yanında taşı. Unutma, bilgiye giden yol sonsuzdur ve attıđın her adım kendi inanılmaz potansiyelini gerçekleştirmeye yönelik bir adımdır.*

*Cesur ol, meraklı ol ve eğitim hayatının sana olasılıklarla dolu bir dünyanın kapılarını açmasına izin ver.*

# Teşekkür

Prof. Dr. Aytuğ Onan'a bu proje boyunca gösterdiği değerli rehberlik ve içgörülerini için en derin şükranlarımı sunarım. Uzmanlığı ve mentorluğu bu araştırmanın şekillenmesinde çok önemli olmuştur. Ayrıca, bu çalışmayı önemli ölçüde geliştirmeme paha biçilmez katkıları ve işbirlikçi ruhu için çalışma arkadaşım Dr. Öğr. Üyesi Orhan Öztürk'e teşekkür ederim. Son olarak, İzmir Kâtip Çelebi Üniversitesi Yazılım Mühendisliği Bölümü öğretim üyelerine destekleri ve teşvikleri için minnettarım. Onların kolektif bilgeliği ve uzmanlığı bu projenin başarıyla tamamlanmasında etkili olmuştur.

# İçindekiler

Öz .....	i
Abstract .....	ii
Teşekkür .....	iv
Şekiller Listesi.....	vii
Tablolar Listesi.....	viii
Kısaltmalar Listesi .....	ix
Semboller Listesi.....	x
<b>1 Giriş .....</b>	<b>1</b>
1.1 Literatür Özeti.....	2
1.2 Projenin Ana Hatları .....	4
<b>2 Materyal ve Metotlar .....</b>	<b>6</b>
2.1 Yapay Zeka .....	6
2.2 Uzman Sistemler.....	7
2.3 Yapay Sinir Ağları .....	8
2.4 Makine Öğrenmesi.....	11
2.5 Kullanılan Araç (Python-Jupyter Notebook) .....	12
2.6 Veri Seti .....	13
2.7 Yöntem (Derin Öğrenme) .....	20
2.7.1 Yapay Sinir Ağı Modelinin Oluşturulması.....	22
2.7.2 Modelin Derlenmesi .....	27
2.7.2.1 Hata Fonksiyonu.....	27
2.7.2.2 Optimizasyon Algoritması.....	29

2.7.2.3 Metrik .....	30
2.7.3 Modelin Eğitilmesi .....	32
<b>3 Analiz Sonuçları .....</b>	<b>35</b>
<b>4 Tartışma ve Sonuç .....</b>	<b>40</b>
<b>Kaynaklar .....</b>	<b>44</b>

# Şekiller Listesi

Şekil 2.1	Makine Öğrenimi Yaşam Döngüsü .....	9
Şekil 3.1	Sonuçlara ilişkin grafikler .....	39



# Tablolar Listesi

Tablo 2.1	Veri setinde yer alan sözel verilerin çıkarıldığı koda ilişkin tablo.....	15
Tablo 2.2	Veri setinde yer alan eksik (NaN) değerlere ait satırların çıkarıldığı koda ilişkin tablo.....	17
Tablo 2.3	Veri setinde yer alan verilerin normalizasyonunun gerçekleştirildiği koda ilişkin tablo.....	18
Tablo 2.4	Eğitim veri setinin xtrain bölümünden hedef değişkenin çıkarıldığı koda ilişkin tablo.....	19
Tablo 2.5	Eğitim veri setinin ytrain bölümüne hedef değişkenin eklendiği koda ilişkin tablo.....	20
Tablo 2.6	Veri setinin üç alt küme (eğitim, doğrulama ve test) ayrılması koduna ilişkin tablo.....	21
Tablo 2.7	Toplu normalleştirme (batch normalization) koduna ilişkin tablo.....	26
Tablo 2.8	Hata fonksiyonu, optimizasyon algoritması ve metriklerin belirlendiği koda ilişkin tablo.....	31
Tablo 2.9	Modelin eğitilmesi koduna ilişkin tablo.....	33
Tablo 3.1	Model sonuçlarının karşılaştırılmasına ilişkin tablo.....	36

# Kısaltmalar Listesi

YZ	Yapay Zeka
YSA	Yapay Sinir Ağları
RELU	Rectified Linear Activation Function
MSE	Mean Squared Error
MAE	Mean Absolute Error
SGD	Stochastic Gradient Descent
NAG	Nesterov Accelerated Gradient
RMSE	Root Mean Squared Error
$R^2$	R-Squared

# Semboller Listesi

W	Ağırlık
b	Bias katsayısı
x	Girdi

# Bölüm 1

## Giriş

İnsanların temel gereksinimi olan barınma, medeniyetin temel taşlarından biri ve günlük yaşamımızın kritik bir yönüdür. Bu temel ihtiyaç sadece hayatta kalmanın ötesine geçerek insan davranışlarının ve toplumsal yapıların çeşitli yönlerini, özellikle de konut kavramının oluşumunu etkilemektedir. İnsanlar bu temel ihtiyacı karşılamak için çabalarken, kaçınılmaz olarak konut piyasasının, öncelikle evlerin satış fiyatlarına yansıyan karmaşıklığıyla karşılaşır. Görünüşte basit olan bu fiyatlar, temelinde barınma ihtiyacı bulunan sayısız sosyo-ekonomik faktörün bir sonucudur [1].

İnsanoğlunun barınma ihtiyacını karşılamada temel olan konut kavramı çok yönlüdür ve zengin bir faktörler dokusu tarafından şekillendirilir. Bu evler yalnızca yapılardan ibaret değildir; doğal çevre ve onun sayısız etkisiyle derinlemesine şekillenir ve her bölgenin kendine özgü karakterini yansıtır. Dahası, toplumsal yapılar ve ilişkiler de oluşumlarında önemli bir rol oynar. Geçim kaynakları, gelir dağılımı kalıpları, kentleşme türleri ve oranları, aile yapıları ve sosyal yaşamın talepleri gibi faktörler karmaşık bir şekilde bir araya gelerek konutların tasarımını, erişilebilirliğini ve gelişimini etkilemektedir. Bu çeşitli unsurlar bir araya gelerek konut kavramını şekillendirmektedir [2].

Konut, temelde sadece basit bir mimari yapıdan daha fazlasıdır; insanın barınma ihtiyacına temel bir yanıttır. Bir sığınak görevi görerek, sakinlerini çevresel etkenlerden korur ve güvenli bir yaşam alanı sağlar. Fiziksel formunun ötesinde, konut; kültürel, çevresel ve teknolojik etkilerin bir karışımını yansıtarak, sakinlerinin gelişen ihtiyaçlarına ve tercihlerine uyum sağlar. Basit barınaklardan karmaşık yapılarına kadar çeşitli tasarımları kapsar, her biri yerel iklim, kültürel normlar ve

mevcut kaynaklara göre şekillendirilmiştir. Konut, insan hayatının dokusunda kilit bir unsur olup toplumda refah ve topluluk duygusunun ayrılmaz bir parçasıdır [3].

Konutlar, sadece yapılar değildir; sığınak, yerleşim ve koruma sunan dinamik ortamlardır. Aile hayatının çekirdeği olarak hizmet ederler, burada bireyler, aynı çevresel koşullar altında yaşamın anlarını kolektif olarak paylaşır ve deneyimler. Bu mekanlar, ailelerin duygusal, fiziksel ve sosyal ihtiyaçlarına uyum sağlamak için evrilir, aile büyüklüğündeki, yaşam tarzındaki ve kültürel uygulamalardaki değişikliklere uyum sağlarlar. Bunlar sadece binalar değil, anıların yaşandığı, geleneklerin aktarıldığı ve aidiyet ve topluluk duygusunun beslendiği sığınaklardır [4].

Barınma ihtiyacına çözüm olarak hizmet eden konutlar, gayrimenkul sektörünün piyasa koşullarını temelden şekillendirir. Bu evlerin değeri ve cazibesi, buldukları konum, çevresel faktörler ve bölgenin sosyo-ekonomik özellikleri tarafından derinden etkilenir. Bu piyasa koşulları, literatürde konut satış fiyatları olarak bilinen konutların alım ve satım fiyatlarını belirler. Bu fiyatları etkileyen faktörler çeşitli ve çok yönlüdür, konut ihtiyacının yerel talebine bağlı olarak önemli ölçüde değişiklik gösterir. Konut ihtiyacının yoğun olduğu bölgelerde, fiyatlar kaynakların kıtlığı veya bolluğunu, yaşam kalitesini ve olanaklara erişilebilirliği yansıtabilir. Konut satış fiyatlarının bu dinamik yapısı, birçok dışsal ve içsel faktörün oynadığı, konut ekonomik manzarasını şekillendiren gayrimenkul piyasasının karmaşıklığını vurgular [5].

## 1.1 Literatür Özeti

Coakley ve Brown (2000), makine öğrenimi tahmin çalışmalarında yapay sinir ağları modellerinin kapsamlı bir analizini sunarlar. Bu modellerin, özellikle veri setindeki değişken sayısı arttıkça hem sınıflandırma hem de regresyon problemlerinde diğer modellere göre üstün performans gösterdiklerini belirtirler. Araştırmaları, çok sayıda değişken içeren karmaşık veri setlerini yönetmede sinir ağlarının uyumluluğunu ve verimliliğini vurgular. Coakley ve Brown ayrıca makine öğrenimi alanındaki araştırmacılara değerli önerilerde bulunurlar. Veri işlemeye sistematik bir yaklaşım benimsemelerini savunarak, özellik verilerinin eğitim, test ve doğrulama olmak üzere üç farklı gruba ayrılmasının önemini vurgularlar. Bu metodolojik öneri, gerçek dünya

senaryolarını sağlam ve kapsamlı bir eğitim süreciyle ele alabilen doğru ve güvenilir tahmin modellerinin geliştirilmesine yöneliktir [6].

Nghiep ve arkadaşlarının 2001 yılındaki çalışmalarında, konut değerlerini tahmin etmek için yapay sinir ağları yöntemleri ve çoklu regresyon analizini geniş bir şekilde kullandılar. Araştırmaları, satış fiyatı, daire alanı, oda sayısı, banyo sayısı ve binanın yaşı gibi çeşitli parametreleri içeriyordu. Bu çalışma, bu yöntemlerin farklı veri koşulları altında etkinliklerini karşılaştırmak için titizlikle yürütüldü. Sonuçlar, optimal veriler kullanıldığında yapay sinir ağları yönteminin gerçek hayata en yakın sonuçları ürettiğini gösterdi. Ancak, çalışma, optimal olmayan veya daha az uygun veriler kullanıldığında, sinir ağından elde edilen çıktının gerçek hayattaki verilerden önemli ölçüde sapma gösterdiğini de ortaya çıkardı. Bu bulgu, makine öğreniminde veri seçiminin önemini ve gayrimenkul değerlemesindeki tahmin modellerinin doğruluğuna olan etkisini derinlemesine vurgulamaktadır [7].

Khalafallah, 2008 yılında gayrimenkul satışlarını tahmin etmek için yapay sinir ağları modelini kullanan yenilikçi bir çalışma yürüttü. Bu araştırma, zaman, faiz oranları, yıllık satış değişimleri, bir önceki yıla göre satış birim değerindeki değişiklikler, ortalama satış süresi ve toplam işlem hacmi gibi çeşitli parametreleri kapsamlı bir şekilde kullandı. Khalafallah'ın metodolojisi, karmaşık piyasa dinamiklerini analiz etme ve yorumlama yeteneğiyle dikkat çekti ve sadece %2'lik bir tolerans oranı ile satış değeri tahminlerine ulaştı. Bu çalışma, yapay sinir ağlarının gayrimenkul piyasa analizindeki potansiyelini vurgulayarak, çoklu etkenlere dayanan piyasa trendlerini anlama ve tahmin etme konusunda sofistike bir araç sunuyor. [8].

Goh, 1998'deki çalışmasında, 1975'ten 1994'e kadar olan dönemi kapsayan Singapur'daki konut inşaatı talebini geniş ve ayrıntılı bir şekilde analiz etti. Box-Jenkins, çoklu doğrusal regresyon ve yapay sinir ağları olmak üzere üç farklı tahmin yöntemi kullandı, her biri verileri farklı bir açıdan incelemesini sağladı. Goh'un araştırması, bu dönemde konut piyasasının evrimini anlamak için hayati olan tarihsel veri ve eğilimlerin derinlemesine keşfini içeriyordu. Bulguları, özellikle yapay sinir ağlarının, konut talebini tahmin etmede doğruluk ve güvenilirlik açısından öne çıktığını gösterdi. Bu çalışma, çeşitli tahmin modellerinin etkinliği hakkında önemli içgörüler sunmanın yanı sıra, özellikle gayrimenkul sektöründe karmaşık, zaman serisi

verilerini işleme konusunda yapay sinir ağlarının gelişmiş yeteneklerini vurguladı, bu da sektör profesyonelleri ve politika yapıcıları için önemli bir yön teşkil etti [9].

2009'da Selim, ev fiyatlarını tahmin etmek için yapay sinir ağları ile hedonik regresyon modellerini karşılaştıran detaylı bir analiz gerçekleştirdi. Aile üyeleri tarafından doldurulan 2004 gelir-gider anketi verilerini kullanarak, 5741 veri noktası ve 46 değişken içeren geniş çaplı bir çalışma yürüttü. Bu zengin veri seti, ev fiyatlandırmasının çeşitli yönlerini incelemek için zemin sağladı. Selim'in araştırması, bu modellerin tahmin gücüne odaklanmakla kalmadı, aynı zamanda karmaşık, çok yönleri gayrimenkul verilerini ele alırken güvenilirlik ve doğruluk açısından da derinlemesine inceledi. Çalışmanın sonuçları, özellikle gayrimenkul değerlemesinde karmaşık, çok değişkenli veri setleriyle uğraşırken, yapay sinir ağlarının hedonik regresyon modellerine göre güvenilirlik açısından üstünlüğünü önemli ölçüde vurguladı. Çalışmanın bulgularının, gayrimenkul sektöründe makine öğrenimi tekniklerinin gelecekteki uygulamaları için, özellikle mülk değerlendirme için daha doğru tahmin modelleri geliştirmede etkileri bulunmaktadır [10].

## 1.2 Projenin Ana Hatları

Ev satış fiyatlarının tahmin edilmesine yönelik geniş çaplı araştırmalar yapılmasına rağmen, net tahminler elde etmek için evrensel olarak etkili bir yöntem olmadığı açıktır.

Çeşitli çalışmalarda keşfedilen farklı metodolojiler ve yaklaşımlar, gayrimenkul piyasasının doğasında bulunan karmaşıklığı ve değişkenliği vurgulamaktadır. Her bir yöntem benzersiz içgörüler sunarken, aynı zamanda kendi sınırlamalarına da sahiptir. Sözü edilen çalışmalar topluca, ev satış fiyatlarının tahmin edilmesinin zorlu ve gelişmekte olan bir ilgi alanı olduğunu ortaya koymaktadır. Bu nedenle, gelecekte daha rafine ve doğru tahmin modellerine yol açabilecek yenilikçi yaklaşımlar ve tartışmaları teşvik ederek, bu alanda sürekli araştırma ve keşfetme ihtiyacını vurgulamaktadır.

Yapay sinir ağları modelleri, özellikle gayrimenkul sektöründe ev satış fiyatlarını belirleme gibi çeşitli tahmin problemlerinde, diğer modellere göre daha üstün bir güvenilirlik seviyesi sergileyerek özellikle etkili olduğunu kanıtlamıştır.

Bu alıřma, ev satıř fiyatlarını doęru bir řekilde tahmin etmek iin bu modellerin gcnden yararlanarak regresyon tabanlı makine ęrenimi yntemlerini kullanmaktadır. alıřmanın kullandığı geniř kapsamlı veri seti, Ames, Iowa'daki evlerin zelliklerini detaylı bir řekilde aıklayan 81 eřitli deęiřkenden oluřmaktadır. Bu zengin veri seti, ev fiyatlarını etkileyen eřitli faktrlerin karmařık etkileřimini yansıtan, doęru tahminler saęlayan kapsamlı bir analize olanak tanır.

Bu arařtırmanın amacı, gayrimenkul analitięi alanına nemli bir katkı sunarak, ev satıř fiyatlarını doęru bir řekilde belirlemek iin ileri regresyon tabanlı makine ęrenimi yntemlerini kullanmaktır. alıřma, Ames, Iowa'daki konutların eřitli ynlerini ayrıntılı bir řekilde yakalayan 81 deęiřkeni kapsayan kapsamlı bir veri setini kullanmaktadır. Bu veri setinin derinlięi ve geniřlięi, analiz iin zengin bir temel saęlayarak, bu blgedeki ev fiyatlarını ynlendiren faktrlere iliřkin nanslı bir anlayıřı kolaylařtırır. Bu yaklařımla arařtırma, sadece mevcut piyasa deęerlerinin doęru tahminlerini saęlamakla kalmayıp, aynı zamanda gayrimenkul piyasasının altında yatan dinamikler hakkında igrler sunmayı hedefleyen tahmin modelleri geliřtirmeyi amalamaktadır.



## Bölüm 2

# Materyal ve Metotlar

### 2.1 Yapay Zeka

Yapay Zeka (YZ), genellikle görevleri yerine getirmek için insan zekasını taklit eden ve topladığı bilgilere göre kendini sürekli olarak geliştirebilen bir sistem olarak tanımlanır. Bu alan, öğrenme, akıl yürütme, problem çözme, algılama, dil anlama ve mantıksal düşünme gibi ileri düzey bilişsel işlevleri kapsar. YZ'nin amacı, insan yeteneklerini ve katkılarını önemli ölçüde artırmak ve genişletmektir. Merkezi hedefi, çıkarım yapma, bağlamı anlama, bilgiyi genelleştirme ve tarihsel verilerden ve deneyimlerden öğrenme gibi insana özgü bilişsel süreçleri taklit eden sistemler geliştirmektir. YZ, bu yetenekleri karmaşık görevleri gerçekleştirmek, büyük veri setlerini analiz etmek, tahminler yapmak ve insanların hızlı veya doğru bir şekilde başaramayacağı içgörüler sağlamak için entegre eder. Bu teknolojinin evrimi, insan ve makine zekası arasındaki boşluğu kapatmayı amaçlayan bilgisayar bilimi, bilişsel psikoloji ve nörobilim karışımını yansıtır [11].

Yapay zekanın (YZ) temel prensibi, insan aklının dünyayı algılama, analiz etme ve buna tepki verme yeteneğini taklit etmektir. YZ, insanların çevreleriyle nasıl etkileşimde bulunduğunu ve onu nasıl yorumladığını taklit etmeyi, büyük miktarda veriyi işleyerek gerçeğe daha yakın sonuçlar elde etmeyi ve genellikle insan zekasının hızını ve analiz genişliğini aşmayı hedefler. YZ'nin insan bilişsel işlevlerini taklit etme ve geliştirme yeteneği, önemli ilerlemelere ve yeniliklere temel oluşturur. YZ sistemleri, deneyimlerden öğrenmek, yeni girdilere uyum sağlamak ve insan benzeri görevleri yerine getirmek üzere tasarlanmıştır. Bunu yaparak YZ, çeşitli alanlarda dönüştürücü bir araç haline gelir, çığır açan gelişmelere yol açar ve sürekli iyileştirme ve yenilik ortamını teşvik eder [11].

Makine öğreniminin kökeni, 1950'lerde Alan Turing'in "Makineler Düşünebilir mi?" sorusuna dayanmaktadır. Bu soru, alan için temel oluşturmuştur. 1956'da John McCarthy, dünyada düzenlenen ilk yapay zeka konferansı olan Dartmouth Konferansı'nda "Yapay Zeka" terimini kullanmıştır. Bu konferans, YZ alanında araştırmacıları ve öncülerini bir araya getirerek, bilgisayar bilimi içinde YZ'nin ayrı bir alan olarak gelişimi ve büyümesi için sahneyi hazırlamıştır. Burada, YZ'nin temel fikirleri, teorileri ve potansiyelleri tartışılarak, bugün gördüğümüz geniş ve hızla gelişen alanın temeli atılmıştır [12].

Yapay Zeka (YZ) ve makine öğreniminde en önemli unsur olan veri, bu alanların temelini oluşturur. Veri, temelde bir veya daha fazla varlıkla ilgili nitel veya nicel değişkenlerin değerlerini temsil eder. Basit sayısal değerlerden karmaşık, yapılandırılmamış metin ve görüntülere kadar çok çeşitli bilgi türlerini kapsar. Bu veriler, YZ sistemleri tarafından desenleri anlamak, tahminler yapmak ve içgörüler elde etmek için toplanır, işlenir ve analiz edilir. YZ ve makine öğrenimi modellerinin etkinliği ve doğruluğu, işledikleri verinin kalitesine, çeşitliliğine ve hacmine büyük ölçüde bağlıdır. Veri, bu modelleri sadece bilgilendirip eğitmekle kalmaz, aynı zamanda sürekli öğrenme ve adaptasyon yoluyla bunları sürekli olarak geliştirir ve iyileştirir [12].

Yapay Zeka (YZ) kullanımı, gayrimenkul sektöründe ev satış fiyatlarını belirlemede önemli bir ilerleme temsil etmektedir. Büyük veri setlerini işleme ve analiz etme yeteneği sayesinde YZ, ev fiyatlarının daha doğru bir şekilde tahmin edilmesini sağlar. Makine öğrenimi algoritmaları ve sinir ağları gibi çeşitli YZ tekniklerini kullanarak, gayrimenkul profesyonelleri piyasa eğilimleri, tarihsel fiyatlar ve mahalle özellikleri gibi geniş veri miktarlarını değerlendirebilir. Bu yaklaşım, fiyat tahminlerinin hassasiyetini artırmakla kalmaz, aynı zamanda mülk değerlerini etkileyen anahtar faktörleri belirlemede de yardımcı olur. YZ'nin öngörüselsel analitiği, alıcılar, satıcılar ve yatırımcılar için daha bilinçli kararlar alınmasına yol açabilir, potansiyel olarak gayrimenkul piyasasının dinamiklerini dönüştürebilir [13].

## 2.2 Uzman Sistemler

Yapay Zeka (YZ) ve uzman sistemler, bilgisayar tabanlı problem çözme alanının iki ayrı bölümünü temsil eder. Ana fark, kapsam ve yeteneklerinde yatar. YZ, geniş

anlamıyla, normalde insan zekasını gerektiren görevleri yerine getirebilen sistemleri ve programları kapsar. Bu, öğrenme, akıl yürütme, problem çözme, algılama ve dil anlama gibi geniş bir yetenek yelpazesini içerir [14]. Bir YZ programı, genel bir insan gibi çeşitli problemleri ele alacak şekilde tasarlanmıştır. Yeni zorluklara uyum sağlayabilir ve geniş bir deneyim yelpazesinden öğrenebilir, bu da onu herhangi bir ortalama insanın ele alabileceği görevlerde çok yönlü kılar [15].

Öte yandan, uzman sistemler, deneyimli bir insan uzmanının düzeyinde belirli bir alanda problemleri çözmek için tasarlanmış YZ'nin özelleştirilmiş bir dalıdır. Bu sistemler, belirli bir alandaki geniş bilgilerle donatılmıştır ve bu alandaki karmaşık sorunları çözmek için akıl yürütme yeteneklerini uygular. Genel YZ'nin aksine, uzman sistemler dar bir uzmanlık alanına odaklanır ve sonuçlara ulaşmak veya önerilerde bulunmak için ayrıntılı, alan özgü uzmanlığı kullanır [16].

## 2.3 Yapay Sinir Ağları

Yapay sinir ağları (YSA), uyarlanabilir yapıları ve örneklerle öğrenme yetenekleri sayesinde çeşitli uygulama alanlarında verilerin modellenmesi için güçlü bir araçtır. Bu ağlar, insan beyninin yapısını ve işlevselliğini taklit edecek şekilde tasarlanmıştır ve karmaşık görevlerin üstesinden gelmelerini sağlar. Paralel bilgi işlemeye izin veren belirli ağırlık katsayıları ile katmanlar halinde organize edilmiş birbirine bağlı işlem birimlerinden oluşurlar [17]. Yapay zekaya yönelik ağ tabanlı yaklaşımların yeniden ortaya çıkması, YSA'ların hem uygulamalarında hem de teorik olarak anlaşılmasında önemli bir olgunlaşmaya yol açmıştır. Tek gizli katmanlı ileri beslemeli ağların yaklaşım ve öğrenme yetenekleri titizlikle analiz edilmiş ve ağ öğrenmesinin birleşik bir görünümü sağlanmıştır. Doğru YSA modelleri oluşturma stratejileri arasında ağ yapısı, verilerin önceden işlenmesi ve fiziksel-sinirsel ağ modelleri ve model transferi gibi teknikler yer almaktadır [18].

Yapay sinir ağlarının belirleyici özelliklerinden biri, tıpkı insanların deneyimlerinden öğrendikleri gibi, verilerden öğrenme yetenekleridir. Performanslarını artırmak ve yeni durumlara uyum sağlamak için daha önce karşılaştıkları bilgileri analiz edebilir ve kullanabilirler [19]. Bu öğrenme süreci, yapay sinir ağlarının içgörüler elde etmesini ve verilerdeki örüntüleri keşfetmesini sağlayarak onları örüntü tanıma ve veri analizi gibi görevlerde oldukça etkili hale getirir [20].

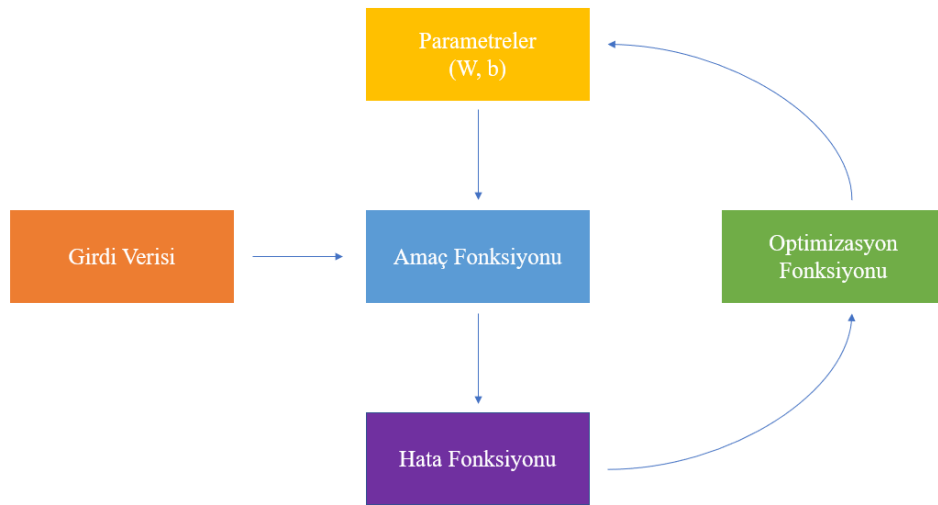
Yapay sinir ağlarını diğerlerinden ayıran şey, bu görevleri açık bir programlamaya ihtiyaç duymadan otonom olarak gerçekleştirme kapasiteleridir [18]. İç parametrelerini otomatik olarak ayarlayabilmeleri ve insan müdahalesi olmadan verilerden öğrenebilmeleri, onları büyük veri kümelerini ve karmaşık sorunları ele almak için inanılmaz derecede verimli kılmaktadır [20].

Pratik anlamda, yapay sinir ağları geniş bir uygulama yelpazesine sahiptir. Tahmine dayalı modellemede mükemmeldirler ve geçmiş verilere dayalı tahminler ve öngörüler yapmalarına olanak tanırırlar. Ayrıca, verileri farklı sınıflara veya gruplara ayırabildikleri sınıflandırma görevlerinde de uzmandırlar. Ek olarak, veriler içindeki benzerlikleri ve örüntüleri tanımlamayı içeren kümelemede kullanılırlar. Yapay sinir ağları, duyuşal girdilere dayalı kararlar almak ve eylemlerde bulunmak için kontrol sistemlerinde bile kullanılmaktadır [21].

Bir yapay sinir ağı yapısı temel olarak girdi verileri, parametreler (ağırlık ve bias değişkeni) ve çıktıdan oluşur.

$$F(x, W, b) = W * x + b$$

Makine öğrenimi yapısı öğrenme yaşam döngüsü olarak adlandırılır. Bu yaşam döngüsünde, amaç fonksiyonuna ek olarak, bir hata fonksiyonu ve bir optimizasyon fonksiyonu da vardır. Yapısı aşağıdaki gibidir [22]:



Şekil 2.1: Makine Öğrenimi Yaşam Döngüsü

Makine öğrenimi alanında, çeşitli modellerin temel işlevselliği, verilerden adaptasyon ve öğrenme yetenekleri etrafında döner. Bu, modelin yapısındaki yapay nöronlar veya karar verme düğümleri gibi katmanlardan oluşan kısımların iç parametrelerini, genellikle ağırlıklar olarak bilinen, dinamik olarak ayarlamaları ile gerçekleştirilir. Bu ayarlamalar, değişiklik yöntemlerini ve kurallarını belirleyen önceden tanımlanmış öğrenme algoritmaları tarafından yönlendirilir[23].

Öğrenme süreci, modele bir dizi girdi verisi besleyerek başlar, burada her girdi istenen bir çıktı veya etiketle ilişkilendirilir. Veri modelin katmanlarından akarken, her katman belirli hesaplamalar yaparak girdiyi adım adım dönüştürür. Son katman modelin çıktısını üretir ve bu çıktı, istenen çıktıyla karşılaştırılır [24].

Modelin tahmini ile gerçek istenen çıktı arasındaki fark, genellikle bir hata veya kayıp olarak nicelendirilir ve kritik bir geri bildirim mekanizması olarak hizmet eder. Öğrenme kuralı, bu geri bildirimini kullanarak hatayı en aza indirecek şekilde ağırlıkları ayarlamak için kullanır. Tahmin yapma, hataları değerlendirme ve ağırlıkları güncelleme süreci birçok döngü boyunca devam eder, bu işleme modelin eğitilmesi denir [25].

Bu eğitim yoluyla model, eğitim verilerinden öğrendiği desenlere dayanarak, gerçekliği yakından yansıtan çıktıları tahmin edebilecek bir duruma ulaşmayı hedefler. Bir makine öğrenimi modelinin verimliliği ve doğruluğu, veri kalitesine, seçilen öğrenme algoritmasının uygunluğuna ve modelin yapısının karmaşıklığına büyük ölçüde bağlıdır [26].

Gayrimenkul ve mülk değerlendirme alanında, yapay sinir ağları (YSA'lar) ev satış fiyatlarını tahmin ve analiz etmek için değerli bir araç olarak ortaya çıkmıştır. YSA'lar, büyük miktarda veriyi işleme ve içindeki karmaşık kalıpları ayırt etme yetenekleri nedeniyle bu alanda mükemmeldir. Konut satış fiyatlarına ilişkin son derece doğru tahminler üretmek için mülk boyutu, konumu, yatak odası sayısı, banyo ve geçmiş satış verileri gibi çeşitli faktörleri göz önünde bulundurabilirler. Ayrıca, YSA'lar yeni verilerle karşılaştıkça tahminlerini uyarlayabilir ve iyileştirebilir, bu da onları emlak piyasalarının dinamik doğasını ele almak için ideal hale getirir. YSA'lardan yararlanarak, emlak uzmanları ve ev sahipleri mülk değerlemesi hakkında değerli

bilgiler edinebilir ve ev satın alma veya satma söz konusu olduğunda bilinçli karar vermeyi sağlayabilir [27].

## 2.4 Makine Öğrenmesi

Makine öğrenimi, makinelerin her görev için açıkça programlanmadan deneyimlerden öğrenmelerini ve geliştirmelerini sağlayan ileri bir tekniktir. Bu yapay zeka (YZ) formu, makinelerin büyük veri setlerini işlemesine ve analiz etmesine, desenleri tanımlamasına ve kazanılan içgörülere dayanarak kararlar almasına olanak tanır. Bir makine öğrenimi sisteminin çekirdeği, daha fazla veriye maruz kaldıkça operasyonlarını uyarlaması ve geliştirmesidir [28].

Tipik bir makine öğrenimi modelinin mimarisi, birkaç anahtar bileşeni içerir: hedef fonksiyon, hata (veya maliyet) fonksiyonu ve optimizasyon fonksiyonu [29]. Hedef fonksiyon, temelde modelin ulaşmayı hedeflediği hedeftir. Bir dizi parametre ve işlediği girdi verileri tarafından tanımlanır. Bu parametreler, modelin istenen sonuca daha yakın hizalanması için öğrenme sürecinde ayarlayacağı değişkenlerdir [30].

Hata fonksiyonu, aynı zamanda yaygın olarak maliyet fonksiyonu olarak da bilinir, makine öğreniminde merkezi bir rol oynar. Modelin performansını, modelin tahminleri ile gerçek sonuçlar arasındaki farkı ölçerek değerlendiren matematiksel bir fonksiyondur. Bu fonksiyon, tahminlerin 'hatasını' veya 'maliyetini' sayısallaştırır ve yaygın örnekleri arasında ortalama kare hatası, mutlak mesafe kaybı, menteşe kaybı, lojistik kaybı ve çapraz entropi kaybı bulunur. Bu fonksiyonların her biri hatayı kendi yöntemine göre hesaplar ve ele alınan problemin özel gereksinimlerine göre seçilir [31].

Hata fonksiyonunu tanımladıktan sonra, bir sonraki kritik bileşen optimizasyon fonksiyonudur. Bu fonksiyonun rolü, hatayı minimize etmek, ideal olarak sıfıra mümkün olduğunca yakın getirmek için modelin parametrelerini yinelemeli olarak ayarlamaktır. Optimizasyon süreci, genellikle parametreleri değiştirerek hata fonksiyonunun minimum değerini bulan gradyan inişi gibi algoritmaları içerir. Bu süreç, modelin sadece verilen verilerden öğrenmesini sağlamakla kalmaz, aynı zamanda zamanla doğruluğunu ve verimliliğini artırır [32].

Bu hedef belirleme, hata ölçümü ve parametre optimizasyonu etkileşimi aracılığıyla, makine öğrenimi modelleri, basit desen tanımadan karmaşık karar verme işlemlerine kadar geniş bir görev yelpazesini gerçekleştirebilir hale gelir, böylece makinelerin verilerle ve dünya ile etkileşim şeklini dönüştürür [33].

## 2.5 Kullanılan Araç (Python-Jupyter Notebook)

Bu araştırmada, birincil geliştirme ortamı olarak Jupyter Notebook kullanıldı. Jupyter Notebook, birçok programlama diline entegre bir çalışma platformu sunan ve açık kaynak yapısına sahip olan çok yönlü, web tabanlı bir uygulamadır. Başlangıçta IPython Notebook olarak piyasaya sürülen ve yalnızca Python'u destekleyen bu araç, zaman içinde geliştirilerek Julia, Octave, R, Haskell ve Ruby gibi birçok başka programlama dilini desteklemeye başlamıştır [34].

Jupyter Notebook'un işlevselliği sadece kodlama ile sınırlı değildir; canlı kod, denklemler, görselleştirmeler ve anlatı metni kombinasyonlarını içeren defterler oluşturmaya olanak tanır. Bu format, özellikle veri bilimi, makine öğrenimi ve derin öğrenme projeleri için uygundur, çünkü araştırmacıların kodlarını açıklamalar, yorumlar ve görsel veri analizi ile sorunsuz ve etkileşimli bir şekilde birleştirmelerine imkan tanır. Araç, hemen çıktı üretebilme, en son bırakıldığı yerden işe devam edebilme ve programı yönetilebilir bölümlere ayırabilme yetenekleriyle kullanılabilirliğini ve etkinliğini artırır [34].

Jupyter Notebook'un bu çalışma bağlamındaki anahtar özelliklerinden biri, özellikle “.csv” formatındaki veri dosyalarını ele alabilme kapasitesidir. Bu özellik, veri setlerinin kolayca yüklenmesini ve manipüle edilmesini sağlar. Bu veri setlerinde sütun adları, verilerin özelliklerini tanımlamak için kullanılır ve her satır, bireysel bir veri noktasını veya örneği temsil eder. Verilerin bu yapılandırılmış temsili, özellikle makine öğrenimi ve derin öğrenme gibi veriye dayalı alanlarda doğru analiz ve yorumlamayı sağlamada kritik öneme sahiptir. Jupyter Notebook'un çeşitli formatlarda çıktılar gösterebilme yeteneği, veri odaklı araştırmalarda daha kapsamlı bir anlayış ve sonuçların sunulmasına olanak tanır ve bu özellik, aracın veriye dayalı araştırmalardaki kullanım değerini daha da artırır [34].

## 2.6 Veri Seti

Çalışmanın veri seti, Ames, Iowa'daki konutların özelliklerini tanımlayan 81 değişkene karşılık gelen bilgilerden oluşmaktadır (Veri seti açık kaynaklı bir veri setidir ve herkesin erişimine açıktır. Ayrıntılı bilgi için bakınız: <https://www.kaggle.com/competitions/house-prices-advanced-regression-techniques/data>).

Bu değişkenler; Id, MSSubClass, MSZoning, LotFrontage, LotArea, Street, Alley, LotShape, LandContour, Utilities, LotConfig, LandSlope, Neighborhood, Condition1, Condition2, BldgType, HouseStyle, OverallQual, OverallCond, YearBuilt, YearRemodAdd, RoofStyle, RoofMatl, Exterior1st, Exterior2nd, MasVnrType, MasVnrArea, ExterQual, ExterCond, Foundation, BsmtQual, BsmtCond, BsmtExposure, BsmtFinType1, BsmtFinSF1, BsmtFinType2, BsmtFinSF2, BsmtUnfSF, TotalBsmtSF, Heating, HeatingQC, CentralAir, Electrical, 1stFlrSF, 2ndFlrSF, LowQualFinSF, GrLivArea, BsmtFullBath, BsmtHalfBath, FullBath, HalfBath, Bedroom, Kitchen, KitchenQual, TotRmsAbvGrd, Functional, Fireplaces, FireplaceQu, GarageType, GarageYrBlt, GarageFinish, GarageCars, GarageArea, GarageQual, GarageCond, PavedDrive, WoodDeckSF, OpenPorchSF, EnclosedPorch, 3SsnPorch, ScreenPorch, PoolArea, PoolQC, Fence, MiscFeature, MiscVal, MoSold, YrSold, SaleType, SaleCondition and SalePrice.

Söz konusu değişkenler sadece veri noktaları değildir; proje için kritik bir öge olan veri setinin içsel özelliklerini temsil ederler. Genellikle gayrimenkul analizi için derlenen bu veri seti, her evi ayrıntılı bir şekilde tanımlayan kapsamlı bir özellikler yelpazesini içerir. Bu özellikler arasında evin büyüklüğü, yaşı, konumu, yatak odası ve banyo sayısı, garaj veya bahçe varlığı, inşaat kalitesi ve malzemeleri, okullar, parklar ve alışveriş merkezleri gibi önemli olanaklara yakınlık gibi özellikler bulunur, ancak bunlarla sınırlı değildir [35].

Bu değişkenler, bir evin satış fiyatını potansiyel olarak etkileyebilecek faktörlere dair değerli içgörüler sağlar. Örneğin, daha büyük bir yaşam alanı veya aranan bir mahalledeki birinci sınıf bir konum, bir evin piyasa değerini önemli ölçüde artırabilir. Buna karşılık, eski mülkler veya tadilat gerektirenler daha düşük fiyatlarla satılabilir. Yüzme havuzu veya yüksek kaliteli bitişler gibi lüks özelliklerin dahil edilmesi de



fiyatı etkileyebilir. Ayrıca, faiz oranları ve gayrimenkul piyasasının genel sağlığı gibi piyasa eğilimleri ve ekonomik faktörler, bu özelliklerin satış fiyatlarına nasıl dönüştüğünü daha da etkileyebilir [36].

Ev satış fiyatlarını tahmin etmek için veri kullanımı bağlamında, 'train.csv' olarak adlandırılan eğitim veri seti ve 'test.csv' olarak adlandırılan test veri seti olmak üzere iki farklı veri seti bulunmaktadır. Her biri, evlerin çeşitli özellikleri hakkında zengin bilgiler içeren bu veri setleri farklı amaçlar için kullanılır. Eğitim veri seti, modelin evlerin satış fiyatları ile farklı özellikler arasındaki ilişkiyi öğrenmesi için kullanılır. Bu set, evin boyutu, konumu, yaşı ve evin satış fiyatını potansiyel olarak etkileyebilecek diğer önemli özellikler gibi konuları kapsayan 81 değişken veya özellik içerir.

Öte yandan, test veri seti, model eğitildikten sonra etkinliğinin değerlendirilmesi için kullanılır. Bu veri seti, eğitim veri setinde bulunan 80 değişkeni içerir. Test veri setinde bulunmayan dikkate değer bir özellik 'SalePrice' (Satış Fiyatı) değişkenidir. Bu, test veri setinin modelin sağlanan özelliklere dayanarak gerçek satış fiyatlarını referans almadan satış fiyatlarını tahmin etmesi amacıyla kasıtlı yapılmış bir eksikliklerdir. Test veri setinde 'SalePrice' değişkeninin olmaması, modelin tahminlerinin sadece eğitim verilerinden öğrendiklerine dayanması ve gerçek satış fiyatlarından etkilenmemesi için makine öğreniminde yaygın bir uygulamadır.

Bu iki veri seti arasındaki ayrım hayati öneme sahiptir. Eğitim veri seti, tahmin modelinin inşa edildiği temeldir. Model, bu veri seti aracılığıyla çeşitli ev özellikleri ile satış fiyatları arasındaki karmaşık ilişkileri öğrenir. Bu arada, test veri seti, modelin ne kadar iyi öğrendiğini ve yeni, görülmemiş verilere uygulayıp doğru tahminler yapabilir yapamayacağını değerlendirmek için bir test alanı olarak hizmet eder [37].

Bu çalışmanın amacının ev satış fiyatlarının tahmin edilmesini sağlayacak bir makine öğrenimi sistemi tasarlamak olduğu göz önüne alındığında, eğitim veri seti ile çalışmanın en uygun yaklaşım olduğu belirlenmiştir. Bu tercihin arkasındaki mantık, eğitim veri setinin, makine öğrenimi modelinin çeşitli evle ilgili özellikleri satış fiyatlarıyla nasıl yorumlayacağını ve işleyeceğini öğretmek için önemli bir bilgi deposu olmasıdır. Bu veri seti, geniş ve kapsamlı bir örnekleme temsil eden toplam 1460 evin verilerini içermektedir. Veri setindeki her bir ev, evin boyutu, konumu, yaşı,

oda sayısı, yatak odaları ve banyolar dahil olmak üzere, garajlar veya havuzlar gibi ek olanakların varlığı ve daha fazlası gibi geniş bir özellik yelpazesi kapsayan 81 farklı özelliikle tanımlanmaktadır.

Bu özellikler, evin detaylı bir profilini oluşturarak, makine öğrenimi modeline ev fiyatlarını etkileyen faktörlere dair bir anlayış sağlar. 1460 farklı örneği ile veri setinin çeşitliliği, makine öğrenimi modelinin geniş bir yelpazeden öğrenmesine olanak tanıyarak, yeni ve görülmemiş evler hakkında tahmin yaparken iyi genelleme yapabilmesini sağlar. Veri toplama ve kullanımına bu kapsamlı yaklaşım, ev satış fiyatları için doğru ve güvenilir bir tahmin modeli geliştirmenin anahtarıdır.

Ev satış fiyatlarını tahmin etmek için kullanılan veri setinde, evlerin özellikleri farklı veri türleri karışımı ile temsil edilmektedir. Bu verilerden bazıları sayısal bilgiler olup, özellikleri ölçülebilir ve açık bir şekilde niceldir. Sayısal bilgiler genellikle evin metrekare cinsinden boyutu, mülkün yaşı, oda sayısı ve istatistiksel modellerde doğrudan kullanılabilir diğer nicel özellikleri içerir. Öte yandan, verilerin bir kısmı sözel veya kategorik olup, tanımlayıcı bilgilerden oluşmaktadır. Bu, evin stili, mahalle kalitesi ve diğer sübjektif veya niteliksel yönleri gibi özellikleri içerir.

Ancak, bu çalışma için, yalnızca sayısal verilere odaklanmaya ve veri setinden sözel bilgileri çıkarmaya karar verildi. Bu karar, sayısal verilerin, özellikle fiyat tahmini için kullanılan regresyon modellerinde uygulanan analitik yöntemlere daha uygun olması nedeniyle alındı. Sayısal veriler, bu modellere doğrudan beslenebilir ve verimli bir şekilde işlenebilir. Sözel veya kategorik veriler, değerli olsa da genellikle makine öğrenimi bağlamında analize uygun bir formata kodlama veya dönüştürme gibi ek ön işleme adımları gerektirir [38]. Sayısal verilere odaklanarak, çalışma modelleme sürecini basitleştirmeyi ve tahmin algoritmasının doğruluğunu ve verimliliğini artırmak hedeflenmektedir.

Tablo 2.1: Veri setinde yer alan sözel verilerin çıkarıldığı koda ilişkin tablo

<b>Python Kodu</b>
<code>data_train=data_train.select_dtypes(exclude=['object'])</code>

Veri analizinin ön işleme aşamasında, kullanılan analitik modellerin gereksinimlerine uyacak şekilde veri setinin iyileştirilmesi çok önemli bir adımdır. Bu süreç genellikle,

kullanılması planlanan algoritmalarla uyumlu olmayabilecek belirli veri türlerinin hariç tutulmasını gerektirir [39].

Bu çalışma bağlamında, Python programlama dili ve veri manipülasyonu ve analizi için güçlü bir araç olan pandas kütüphanesi, bu tür ön işleme görevleri için kullanılmıştır. Özellikle, pandas'ta bulunan 'select\_dtypes' yöntemi kullanılarak veri seti bir filtreleme işlemine tabi tutulur. Bu yöntem sütunların veri türlerine göre seçilmesini veya hariç tutulmasını kolaylaştırır. Yukarıdaki komut ile 'object' veri türü altında kategorize edilen sütunları hariç tutacak şekilde yapılandırılmıştır. Pandas'taki 'nesne' veri türü genellikle metin dizeleri veya metin ve sayılar gibi farklı türlerin bir karışımını içeren sütunları kapsar. Bu komutun çalıştırılmasıyla veri seti, sayısal olmayan veriler içeren tüm sütunlardan arındırılır ve böylece veri kümesi sayısal algoritmaların analitik gereksinimleriyle uyumlu hale getirilir.

Bu ön işleme adımı, verilerin makine öğrenimi modellerinin ve istatistiksel araçların gerektirdiği belirli girdi biçimine uymasını sağlamak ve böylece sonraki analizin sağlamlığını ve geçerliliğini artırmayı sağlar.

Sözel bilgi içeren özellikler veri setinden çıkarıldıktan sonra, analiz için toplam 38 sayısal özellik elde tutulmuştur. Bu özellikler, veri kümesindeki evlerin çeşitli yönlerine ilişkin kapsamlı bir nicel bilgiler sağlamaktadır. 'Id' her ev için benzersiz bir tanımlayıcı görevi görmektedir. 'MSSubClass' satışta yer alan konut tipini gösteren sayısal bir koddur. 'LotFrontage' mülke bağlanan caddenin doğrusal ayaklarını temsil eder ve 'LotArea' fit kare cinsinden arsa boyutudur. 'OverallQual' ve 'OverallCond' sırasıyla evin genel malzeme ve bitişinin ve genel durumunun derecelendirmeleridir.

Temel yapısal özellikler arasında orijinal inşaat tarihini gösteren 'YearBuilt' ve büyük tadilatların yapıldığı yılı gösteren 'YearRemodAdd' bulunmaktadır. 'MasVnrArea' duvar kaplama alanını feet kare cinsinden ölçer. Bodrum alanı, birinci ve ikinci bitmiş alanlar için 'BsmtFinSF1' ve 'BsmtFinSF2', bitmemiş alan için 'BsmtUnfSF' ve toplam bodrum alanı için 'TotalBsmtSF' ile detaylandırılır. '1stFlrSF' ve '2ndFlrSF' sırasıyla feet kare cinsinden birinci ve ikinci kat alanlarını tanımlarken, 'LowQualFinSF' düşük kaliteli bitmiş feet kareyi (tüm katlar) hesaplar. 'GrLivArea' zemin üstü yaşam alanını ölçer.

Banyolar 'BsmtFullBath', 'BsmtHalfBath', 'FullBath' ve 'HalfBath' ile detaylandırılır ve hem bodrumdaki hem de yer üstündeki tam ve yarım banyo sayısını temsil eder. Yatak odası sayısı 'BedroomAbvGr', mutfak sayısı ise 'KitchenAbvGr'dir. 'TotRmsAbvGrd' toplam odaları (banyolar hariç) detaylandırır. 'Fireplaces' şömine sayısını sayar.

Garaj özellikleri 'GarageYrBlt' (garajın inşa edildiği yıl), 'GarageCars' (araba kapasitesi olarak garajın boyutu) ve 'GarageArea' (fit kare olarak boyut) ile tanımlar. Dış mekân özellikleri arasında 'WoodDeckSF' (feet kare cinsinden ahşap güverte alanı), 'OpenPorchSF' (feet kare cinsinden açık sundurma alanı), 'EnclosedPorch' (feet kare cinsinden kapalı sundurma alanı), '3SsnPorch' (feet kare cinsinden üç mevsim sundurma alanı) ve 'ScreenPorch' (feet kare cinsinden perde sundurma alanı) bulunmaktadır. 'PoolArea' fit kare cinsinden havuz alanını temsil eder ve 'MiscVal' çeşitli özelliklerin değerini hesaplar. Son iki özellik olan 'MoSold' ve 'YrSold' satış ayı ve yılını, 'SalePrice' ise mülkün satış fiyatını göstermektedir.

Makine öğrenimi modeli için veri setini hazırlama sürecinde, verilerin bütünlüğüne ve kalitesine özel dikkat gösterildi, çünkü bu, modelin tahminlerinin doğruluğunu ve güvenilirliğini belirlemede hayati önem taşır. Bu doğrultuda, veri seti içindeki boş veya geçersiz bilgiler içeren örnekler belirlendi ve kaldırıldı. Boş veya geçersiz veriler, modele yanlışlık veya yanlışlık getirebilir ve modelin performansını olumsuz etkileyebilir [40]. Bu nedenle, çalışmanın sağlamlığını artırmak ve sonuçlarının güvenilirliğini sağlamak için, bu tür verileri veri setinden temizlemek gerekiyordu.

Tablo 2.2: Veri setinde yer alan eksik (NaN) değerlere ait satırların çıkarıldığı koda ilişkin tablo

Python Kodu
<pre>data_train=data_train.dropna()</pre>

Pandas 'dropna()', veri kümesinden eksik değerler (NaN'ler) içeren tüm satırları kaldırmak için tasarlanmıştır. Bu komutun uygulanması, verilerin bütünlüğünü ve doğruluğunu korumak için gerekli olan veri kümesinin eksik veya boş girişlerden arındırılmasını sağlar. Bu örneklerin kaldırılması, veri setini analiz için daha tutarlı ve güvenilir bir forma dönüştürmeyi amaçlayan veri ön işleme sürecinde yaygın bir uygulamadır. Bu titiz temizleme sürecinin sonucunda, eğitim veri setinin örnek boyutu azaldı. Başlangıçta daha fazla sayıda örneği içeren veri seti, 1121 örneğe daraltıldı.

Şimdi sorunlu verilerden arındırılmış bu veri seti, makine öğrenimi modeline öğrenmek ve tahminler yapmak için daha sağlam bir temel sağlar, bu da çalışmada daha doğru ve anlamlı sonuçlara ulaşma olasılığını artırır.

Yapay sinir ağı (YSA) modelinde kullanılan verilerin anlamlı ve güvenilir sonuçlar vermesini sağlamak ve sonraki değerlendirmelerin verimliliğini artırmak için tüm özellik değerlerinin normalizasyonu gerçekleştirildi. Bu süreç, veri setinin özelliklerinin aralığını 0 ile 1 arasında ölçeklendirmeyi içerir. Bu standartlaştırma tekniği, YSA bağlamında kritik öneme sahiptir, çünkü tüm girdi özelliklerinin modele eşit derecede katkıda bulunmasını sağlayarak öğrenme sürecini stabilize etmeye ve hızlandırmaya yardımcı olur. Bu ölçeklendirme olmadan, daha büyük sayısal aralıklara sahip özellikler modeli orantısız bir şekilde etkileyebilir, bu da çarpık sonuçlara yol açabilir ve modelin etkili bir şekilde öğrenme yeteneğini engelleyebilir [41].

Tablo 2.3: Veri setinde yer alan verilerin normalizasyonunun gerçekleştirildiği koda ilişkin tablo

Python Kodu
<pre>from sklearn import preprocessing import pandas as pd names = data_train.columns data_train = scaler.fit_transform(data_train) data_train = pd.DataFrame(data_train, columns=names) data_train.head()</pre>

Bu işlem, Python'da yer alan iki önemli kütüphanenin yardımıyla gerçekleştirilmiştir: pandas ve scikit-learn. Veri manipülasyon yetenekleriyle tanınan pandas, veri kümesini işlemek için kullanılırken, makine öğrenimi konusunda uzmanlaşmış bir kütüphane olan scikit-learn, normalleştirme için gerekli araçları sağladı.

Veri setinin normalleştirilmesi 'scaler.fit\_transform(data\_train)' komutu tarafından gerçekleştirilir. Burada 'scaler', scikit-learn'in ön işleme modülünden, muhtemelen kodda önceden tanımlanmış bir normalleştirme fonksiyonunun bir örneğidir. 'fit\_transform' yöntemi, ölçeklendirmeyi data\_train'e uygulayarak her özelliği 0 ile 1 arasında bir aralığa normalleştirir.

Normalizasyon, YSA'nın girdi deęerlerinin büyüklüğüne duyarlılığı göz önüne alındığında özellikle önemlidir. Girdi özellikleri çok farklı ölçeklere sahip olduğunda, bu özelliklerle ilişkili ağırlıklar önemli ölçüde deęişebilir, bu da optimizasyon algoritmasının daha uzun süre almasına veya bazı durumlarda optimal bir çözüm bulamamasına neden olabilir. Tüm özellikleri uniform bir aralığa ölçeklendirerek bu sorun hafifletilir. Orijinal ölçeğine bakılmaksızın her özellik, artık modelin öğrenmesini etkileme eşit bir fırsata sahiptir, bu da daha dengeli ve verimli bir eğitim sürecine yol açar. Ayrıca, bu ölçek birliği, bazı özelliklerin dięerleri üzerinde hakimiyetini önlemeye yardımcı olur ve her özelliğin nihai model çıktısına daha etkin bir katkıda bulunmasını sağlar [41].

Kapsamlı veri ön işleme işlemleri, sayısal olmayan özelliklerin çıkarılması ve eksik deęerlerin ele alınması dahil, tamamlandıktan sonra, bu çalışmanın temelini oluşturan veri seti, 1121 evin bilgilerini kapsayacak şekilde elde edilmiştir. Bu veri seti, her bir ev için potansiyel olarak ev satış fiyatlarını belirlemede etkili olabilecek 38 farklı özellik veya niteliğe odaklanmasıyla karakterize edilir. Ancak bu 38 özellik içinde, 'Id' ve 'SalePrice' iki özel deęişken özellikle önemlidir.

'Id' deęişkeni, her bir ev için benzersiz bir tanımlayıcı görevi görür ve tahmin modelleme sürecine katkıda bulunmaz. Dięer taraftan, 'SalePrice' hedef deęişkendir ve bu çalışmanın asıl amacı bu deęişkenin tahmin edilmesidir. Makine öğrenimi modelinin eğitilmesi amacıyla, bu iki deęişken veri setinden çıkarılır. 'Id', yalnızca tahmini bir deęer olmadığı için ve tahmin deęeri taşımadığı için modelin öğrenme sürecini bozmasını önlemek amacıyla kaldırılır. 'SalePrice' ise hedef olarak ayrılır ve modelin tahminleri ile karşılaştırılıp deęerlendirilecek bağımlı deęişken kümesini oluşturur.

Tablo 2.4: Eğitim veri setinin xtrain bölümünden hedef deęişkenin çıkarıldığı koda ilişkin tablo

Python Kodu
xtrain=data_train.drop(['SalePrice','Id'], axis=1)

Sonuç olarak, eğitim veri seti, 'Id' ve 'SalePrice' çıkarıldıktan sonra her biri 36 özellik içeren 1121 evi kapsayacak şekilde ayarlanır. Bu, her biri bir evi temsil eden 1121 satır ve her biri evin özelliklerine karşılık gelen 36 sütundan oluşan bir eğitim seti matrisi sonucunu doğurur. Bu optimize edilmiş veri seti boyutu ve yapısı, makine öğrenimi

modelinin etkili bir şekilde eğitilmesine olanak sağlayarak modelin ev özellikleri ile satış fiyatları arasındaki karmaşık ilişkileri yüksek doğrulukla öğrenmesini sağlar.

Test verileri, makine öğrenimi modelinin performansını değerlendirmede önemlidir. Bu amaçla, odak noktası özellikle modelin tahmin etmeyi hedeflediği hedef değişken olan 'SalePrice' sütunu üzerineydi. Çeşitli özellikler içeren eğitim verilerinin aksine, test verileri sadece bu tek sütunu içerecek şekilde düzenlendi. Bu karar, modelin ev satış fiyatlarını doğru bir şekilde tahmin etme yeteneğini ek değişkenlerin etkisi olmadan doğrudan değerlendirmek için alındı.

Tablo 2.5: Eğitim veri setinin ytrain bölümüne hedef değişkenin eklendiği koda ilişkin tablo

Python Kodu
ytrain=data_train["SalePrice"]

'SalePrice' sütunu, evlerin gerçek satış fiyatlarını temsil eder ve modelin tahminleri ile karşılaştırıldığı bir ölçüt olarak kullanılır. Test verilerinin boyutu, kapsamlı bir değerlendirme sağlamak için eğitim verilerindeki giriş sayısını yansıtacak şekilde ayarlandı. Böylece, test veri seti, 1121 eve karşılık gelen 1121 satır ve 'SalePrice' olan tek bir sütundan oluşan bir matris olarak yapılandırıldı. Bu matristeki her satır, bir evin gerçek satış fiyatını temsil eder ve modelin tahminlerinin doğruluğunu doğrulamak için net ve odaklanmış bir veri seti sağlar. Bu yaklaşım, makine öğrenimi modelinin gerçek dünya senaryolarında etkinliğini belirlemede önemlidir, burada tahmin yetenekleri bilinen sonuçlara karşı değerlendirilebilir.

## 2.7 Yöntem (Derin Öğrenme)

Derin öğrenme, verileri işleme ve karar verme için desenler oluşturma konusunda insan beyninin işleyişini taklit eden, yapay zekanın (YZ) bir alt kümesi olan makine öğreniminin bir dalıdır [42]. Temel olarak, birçok katmana sahip sinir ağlarının kullanılmasıyla karakterize edilir, bu nedenle 'derin' terimi kullanılır. Bu sinir ağları, insan beyninden ilham alan bir yapıda birbirine bağlı düğümlerden veya 'nöronlardan' oluşur. Derin öğrenme modelleri, otomatik olarak deneyimlerden öğrenir ve gelişir, karmaşık desenleri ve ilişkileri belirlemek için büyük miktarda veri işler. Bu yetenek, onları görüntü ve konuşma tanıma, doğal dil işleme ve otonom araç navigasyonu gibi

görevler için olağanüstü etkili kılar [43]. Geleneksel makine öğrenimi algoritmalarının verileri doğrusal olarak analiz etmesinin aksine, derin öğrenmenin çok katmanlı sinir ağları, hiyerarşik bir yaklaşımı mümkün kılar. Bu, her bir sonraki katmanın girdi olarak önceki katmandan çıktıyı kullanması anlamına gelir, bu da modelin verileri birden fazla, giderek daha karmaşık seviyelerde anlamasını sağlar. Derin öğrenme, birçok alanda devrim yaratan, hesaplama gücünde ve veri analizinde dikkate değer iyileştirmeler sunmaktadır [44].

Klasik makine öğrenimi tekniklerinde olduğu gibi, derin öğrenme de veri setinin eğitim ve değerlendirme amacıyla belirli alt kümelere ayrılmasını gerektirir. Bu süreç genellikle veri setinin 'eğitim' seti ve 'hold\_out' veya test seti olarak bölünmesini içerir. Eğitim setinin asıl amacı, modele çeşitli örnekler sunarak parametrelerini öğrenmesini ve uyarlamasını sağlamaktır. Bu set, verilerin önemli bir kısmını içerir ve hem girdi özelliklerini hem de karşılık gelen hedef değerleri içerir, böylece modelin aralarındaki ilişkileri kurmasını ve geliştirmesini sağlar [45].

Öte yandan 'hold\_out' veya test seti, farklı bir işleve hizmet eder. Modelin eğitim aşamasında görmediği veriler üzerinde performansını değerlendirmek için kullanılır, böylece genelleme yeteneklerine tarafsız bir değerlendirme sağlar. Derin öğrenmede, bu test setini iki ayrı alt kümeye ayırmak da yaygındır: biri test için, diğeri doğrulama için. Doğrulama seti, modelin parametrelerini ince ayar yapmak ve aşırı uyuma engel olmak için modelin eğitim süreci sırasında kullanılır. Bu, bir modelin eğitim verilerine çok yakın şekillendirildiği ve yeni verilere iyi genelleme yapamadığı durumdur. Test seti ise, eğitim ve doğrulama aşamaları tamamlandıktan sonra modelin nihai değerlendirmesi için saklanır [46].

Veri setini bu üç alt küme- eğitim, doğrulama ve test- ayırarak, derin öğrenme modelleri daha etkili bir şekilde eğitilebilir, doğrulanabilir ve test edilebilir. Bu yaklaşım, modelin sadece eğitim verilerindeki temel desenleri öğrenmesini değil, aynı zamanda bu öğrenmeyi yeni, görülmemiş verilere genelleştirmesini sağlar, böylece tahmin yeteneklerinin daha sağlam ve doğru bir değerlendirmesini sunar [47].

Tablo 2.6: Veri setinin üç alt küme- (eğitim, doğrulama ve test) ayrılması koduna ilişkin tablo

<b>Python Kodu</b>
--------------------



```
from sklearn.model_selection import train_test_split
X_train, X_holdout, y_train, y_holdout = train_test_split(xtrain, ytrain,
test_size=0.3, random_state=0)
X_val, X_test, y_val, y_test = train_test_split(X_holdout, y_holdout, test_size=0.5,
random_state=0)
```

Veri setinin eğitim ve test alt kümelerine ayrılması, makine öğrenimi sürecinde kritik bir adımdır ve scikit-learn kütüphanesinde yer alan 'train\_test\_split' metodu kullanılarak gerçekleştirilir. Scikit-learn, makine öğrenimi için yaygın olarak kullanılan bir Python kütüphanesidir. 'train\_test\_split' fonksiyonu, özellikle veri setini rastgele ayrı gruplara ayırmak için tasarlanmıştır. Bu rastgele bölünme, her iki setin de genel veri setini temsil etmesini sağlamak için önemlidir, böylece verilerin bütünlüğü ve çeşitliliği korunur.

Veri setinin %70'lik bir kısmının eğitim amaçları için ayrılmış, kalan %30'luk kısmı ise test için ayrılmıştır. Bu oran, modeli etkili bir şekilde eğitmek için yeterince büyük bir eğitim veri setine ihtiyaç duyarken aynı zamanda modelin performansını doğrulamak için önemli bir kısmı test için ayırır. Eğitim seti (%70'lik veri), modele veriler içindeki desenleri ve ilişkileri öğrenmesi ve tanımlaması için kullanılır. Test seti, aynı zamanda 'hold\_out' seti olarak da bilinir, verilerin %30'unu içerir ve modelin yeni, görülmemiş veriler üzerindeki performansını değerlendirmek için kullanılır, böylece tahmin yeteneklerine tarafsız bir değerlendirme sağlar.

Sonuç olarak, veri seti dört bileşene ayrılır: Eğitim ve test için sırasıyla özellik verilerini içeren 'X\_train' ve 'X\_holdout' ve karşılık gelen hedef değişkenleri içeren 'y\_train' ve 'y\_holdout'. 'X' bileşenleri, modelin tahmin yapmak için kullandığı girdi özelliklerini temsil ederken, 'y' bileşenleri modelin tahmin etmeye çalıştığı gerçek hedef değerleri temsil eder. Eğitim ve test alt kümelerine bu bölünme, makine öğrenimi iş akışında temel olup modelin doğru ve güvenilir tahminler elde etmek için etkili bir şekilde eğitilmesini ve test edilmesini sağlar.

## 2.7.1 Yapay Sinir Ağı Modelinin Oluşturulması

Bu araştırmada, ana hedef özellikle tek bir sonucun tahminine odaklanan bir tahmin modeli geliştirmektir. Bu nedenle, istatistiksel ve makine öğrenimi metodolojilerinde temel bir yaklaşım olan regresyon modeli kullanılmaktadır. Regresyon analizi, bir

veya daha fazla girdi özelliğine dayanarak sürekli bir sonucun tahmin edilmesi gereken bu tür görevler için özellikle uygundur. Bu çalışma bağlamında, model, verilerin çeşitli özelliklerine dayanarak tekil, nicel bir sonucu tahmin etmeyi amaçlamaktadır [48].

Kategorik sonuçlar tahmin eden sınıflandırma modellerinin aksine, regresyon modelleri sayısal, genellikle ondalık bir değeri tahmin etmek için tasarlanmıştır. Bu, fiyatlar, sıcaklıklar, mesafeler veya bu çalışmada olduğu gibi, tahmin edilen sonucu temsil eden belirli bir değer gibi herhangi bir sürekli nicelik olabilir. Regresyon modeli, girdi özellikleri (bağımsız değişkenler) ve sürekli sonuç (bağımlı değişken) arasında bir ilişki kurarak çalışır. Veri noktalarına en uygun çizgiyi (doğrusal regresyonda) veya eğriyi (doğrusal olmayan regresyonda) çizmeye çalışır ve veri setindeki tahmin edilen değerler ile gerçek değerler arasındaki farkı en aza indirmeye çalışır. Bu model, birden fazla özelliği veya özelliği olan çeşitli veri türlerini ele alabilme becerisine sahiptir ve çeşitli alanlardaki karar verme süreçleri için gerekli olan kesin, ölçülebilir tahminler sağlayabilir [48].

Bu çalışmada, toplam dört farklı katmandan oluşan bir ileri beslemeli yapay sinir ağı (YSA) modeli tasarlanmıştır. Bu tür ağ mimarisi, girdi katmanından çıktı katmanına bilginin herhangi bir geriye dönük veya döngüsel bağlantı olmadan ardışık bir şekilde akmasıyla karakterize edilir, bu nedenle 'ileri beslemeli' terimi kullanılır. Bu mimarideki ilk katman girdi katmanıdır ve başlangıçtaki verileri almakla sorumludur. Bu katmanın temel işlevi, gelen verileri daha fazla işlemek üzere sonraki katmanlara dağıtmaktır.

Girdi katmanını takiben, model iki gizli katman içerir. Bu katmanlar bir sinir ağına hayati öneme sahiptir çünkü hesaplamaların çoğunu gerçekleştirirler. Gizli katmanlar, modelin ağırlıklı bağlantılar ve aktivasyon fonksiyonları aracılığıyla girdi verilerini işlediği ve verilerde yer alan karmaşık desenleri ve ilişkileri kademeli olarak çıkardığı ve öğrendiği yerlerdir. Birden fazla gizli katmanın varlığı, verilerin daha derin ve daha nüanslı bir anlayışına katkıda bulunur ve modelin doğru tahminler yapma yeteneğine katkıda bulunur.

Bu sinir ağı modelindeki son katman çıktı katmanıdır. Görevi, modelin nihai sonuçlarını veya tahminlerini sunmaktır. Bu çalışma bağlamında, çıktı katmanı, girdi ve gizli katmanlardan öğrenilen desenlere dayanarak tahmin edilen değerleri sağlar.

Girdi katmanı, iki gizli katman ve çıktı katmanını kapsayan tüm model, kolektif olarak eğitilir. Eğitim süreci sırasında, model bu katmanlar arasındaki bağlantıların ağırlıklarını tahmin hatalarını en aza indirmek için ayarlar, böylece gelecekteki tahminlerde doğruluğunu ve verimliliğini artırır.

Yapay sinir ağı modelinin performansını ve doğruluğunu artırmak için toplu normalleştirme (batch normalization) katmanı dahil edildi. Bu katmanın entegrasyonu, görev için en etkili ve uygun modeli bulmak amacıyla sinir ağı mimarisinin ince ayarını yapmakta kritik bir adımdır. Toplu normalleştirme, ağındaki belirli bir katmana girdileri standartlaştırmak için kullanılan bir tekniktir. Bu işlem, önceki katmanın aktivasyonlarını ayarlayarak ve ölçeklendirerek ağındaki daha hızlı ve daha stabil bir şekilde eğitilmesini sağlar. Bu süreç, eğitim sırasında her katmanın girdilerinin dağılımının değişmesi ve ağındaki yakınsamasını zorlaştıran içsel kovaryant kayması gibi sorunlarla mücadelede özellikle faydalıdır.

Farklı yapılandırmaların etkinliğini kapsamlı bir şekilde değerlendirmek için toplamda 10 model çalıştırması gerçekleştirildi. Bunlardan 5'inde özellikle toplu normalleştirme uygulandı. Bu yaklaşım, toplu normalleştirme olmayan ve olan modeller arasında kapsamlı bir karşılaştırma yapılmasına olanak tanıyarak, bu tekniğin modelin öğrenme süreci ve genel performansı üzerindeki etkisine dair değerli içgörüler sağladı. Farklı çalıştırmalarda toplu normalleştirme kullanımını değiştirerek, çalışma modelin doğruluğunu ve yakınsama hızını artırmadaki rolünü daha doğru bir şekilde belirleyebildi. Bu karşılaştırmalı analiz, deneysel araştırmalarda hayati öneme sahiptir, çünkü sinir ağı modelleri için optimal strateji ve yapılandırmaların belirlenmesine olanak tanır ve daha verimli ve doğru tahmin modellerinin geliştirilmesine rehberlik eder.

Tasarlanan yapay sinir ağının mimarisinde, girdi katmanı modelin başlangıç yapısını tanımlamada kritik bir rol oynamaktadır. Bu katman, veri setinde bulunan özellikleri barındıracak şekilde özel olarak yapılandırılmıştır ve her özellik türü bir nöron tarafından temsil edilir. Bu durumda, veri seti 36 farklı özellik türünü içermekte olup her biri veri hakkında benzersiz ve değerli bilgiler sağlar. Buna göre, girdi katmanı 36 nörondan oluşmakta ve her bir nöron veri setinden belirli bir özellik türünü işlemek üzere tasarlanmıştır.

Girdi katmanındaki nöron sayısı, sinir ağı tasarımının kritik bir yönüdür, çünkü bu doğrudan girdi verilerinin boyutluluğuna karşılık gelir. 36 nöron olması, modelin aynı anda 36 farklı girdiyi alıp işleyebileceği anlamına gelir. Bu tasarım, veri seti içindeki tüm ilgili bilgilerin, sinir ağının başlangıç aşamasında etkili bir şekilde yakalanıp kullanılmasını sağlar. Böylece, girdi katmanının boyutu bir (1) olarak tanımlanır, bu da bu katmandaki her bir nöronun karşılık gelen özelliğin tek boyutlu bir temsili olduğu anlamına gelir. Özellikler ile nöronlar arasındaki bu eşleme, verilerin sinir ağına doğru bir şekilde beslenmesi için önemlidir ve bunun sonucunda sonraki katmanlar bu girdiye dayanarak karmaşık hesaplamalar yapabilir.

Yapay sinir ağı modelinin eğitim aşamasında, gizli katmanların yapısını belirlemek için kapsamlı bir yaklaşım benimsendi. Modelin her gizli katmanındaki nöron sayısının optimal sayısını belirlemek için çeşitli yapılandırmalar denendi. Özellikle, ilk gizli katman beş farklı nöron sayısı ile test edildi: 20, 40, 60, 80 ve 100 nöron. Benzer şekilde, ikinci gizli katman da farklı bir nöron sayısı seti ile yapılandırıldı: 10, 20, 30, 40 ve 50 nöron. Her gizli katmandaki nöron sayısındaki bu sistematik varyasyon, modelin karmaşıklığının performansını nasıl etkilediğini araştırmayı amaçladı.

Nöron sayısını değiştirmenin yanı sıra, çalışma her nöron yapılandırmasıyla birlikte bir toplu normalleştirme katmanının uygulanmasının etkisini de inceledi. Toplu normalleştirme katmanı, her iki gizli katmandaki belirli bir nöron sayısı için dahil edildi veya çıkarıldı. Bu, her mini-yığın için bir katmana girdileri standartlaştıran bir teknik olan toplu normalleştirmenin dahil edilmesinin, modelin öğrenme verimliliğini ve genel performansını iyileştirip iyileştirmediğini ayrıntılı bir şekilde incelemeye olanak tanıdı.

Ayrıca, her iki gizli katmanda aktivasyon fonksiyonu olarak Düzeltilmiş Doğrusal Aktivasyon Fonksiyonu (Rectified Linear Activation Function-RELU) kullanıldı. RELU'nun seçimi, derin sinir ağlarında yaygın olarak karşılaşılan kaybolan gradyan problemine çözüm sunma yeteneği nedeniyle önemlidir. RELU, basitliği ve verimliliği ile bilinir, negatif değerleri etkili bir şekilde sifira ayarlayarak ve pozitif değerleri koruyarak modellerin eğitim sırasında daha hızlı yakınsamasını sağlar. Bu aktivasyon fonksiyonu, verilerden karmaşık desenleri öğrenmede gizli katmanların etkinliğine katkıda bulunarak eğitim sürecinde önemli bir rol oynadı.

Tasarlanan sinir ağı modelinin mimarisinde, regresyon analizine odaklanıldığı için çıktı katmanının yapılandırılması son derece önemlidir. Modelin regresyon doğası göz önünde bulundurularak, çıktı katmanı, sınıflandırma modellerinde sıklıkla bulunan daha karmaşık yapıların aksine, tek bir nörondan oluşan bir katman olarak tasarlanmıştır. Çıktı katmanındaki bu tek nöron, modelin girdi verilerine dayanarak tahmin edilen sonucu temsil eden sürekli bir değeri üreten, modelin nihai çıktısını üretmede hayati bir rol oynar.

Ayrıca, çıktı katmanındaki bu nöron için aktivasyon fonksiyonu olarak RELU kullanılmıştır. RELU'nun çıktı katmanında kullanılması, verilerdeki doğrusal olmayan ilişkileri kolaylaştırma verimliliği göz önünde bulundurularak yapılmıştır. RELU, eğitimde diğer aktivasyon fonksiyonlarına kıyasla daha hızlı yakınsama sağlama yeteneği ve kaybolan gradyan problemiyle mücadeledeki etkinliği nedeniyle birçok sinir ağı modelinde tercih edilir. Bu regresyon modeli bağlamında, RELU fonksiyonu, çıktı nöronunun girdi özelliklerini etkili bir şekilde tahmin edilmesini sağlayarak modelin verilerdeki karmaşık ilişkileri yakalama ve temsil etme yeteneğini artırır.

Tablo 2.7: Toplu normalleştirme (batch normalization) koduna ilişkin tablo

<b>Python Kodu (Batch Normalization)</b>
<pre><b>from</b> tensorflow <b>import</b> keras <b>from</b> tensorflow.keras.layers <b>import</b> BatchNormalization <b>def</b> create_nn(nodes1=100, nodes2=50):     # Input layer     inputs = keras.layers.Input(shape=(xtrain.shape[1],), name='input')     # Dense layers     layers_dense = keras.layers.BatchNormalization()(inputs)     layers_dense = keras.layers.Dense(nodes1, 'relu')(layers_dense)     #layers_dense = keras.layers.Dense(nodes1, 'relu')(inputs)     layers_dense = keras.layers.Dense(nodes2, 'relu')(layers_dense)     # Output layer     outputs = keras.layers.Dense(1, 'relu')(layers_dense) <b>return</b> keras.Model(inputs=inputs, outputs=outputs, name="SalePrice_Predict_Model") model = create_nn(20, 10) model.summary()</pre>

Makine öğrenimi için açık kaynaklı bir yazılım kütüphanesi olan TensorFlow ve yüksek seviyeli bir sinir ağları API'si olan Keras kullanılarak oluşturulan tahmin modeli için sinir ağı mimarisinin özelliklerine detaylıca değinilmektedir. Kod parçacığı, satış fiyatlarını tahmin etmek için tasarlanmış özel bir sinir ağı fonksiyonu olan 'create\_nn'in yapısını ortaya koymaktadır.

'create\_nn' fonksiyonu, ilk ve ikinci gizli katmanlardaki nöron sayısını temsil eden 'nodes1' ve 'nodes2' adında iki parametre ile tanımlanmıştır. Varsayılan olarak, bunlar sırasıyla 100 ve 50 nöron olarak ayarlanmıştır. Sinir ağı mimarisi, 'keras.layers.Input' kullanılarak oluşturulan bir girdi katmanı ile başlar. Girdi katmanı, eğitim verilerindeki özelliklerin sayısına (xtrain.shape[1]) uygun şekilde tanımlanmıştır.

Girdi katmanını takiben, 'keras.layers.BatchNormalization()' kullanılarak bir toplu normalleştirme katmanı uygulanır. Bu katman, girdi katmanından gelen aktivasyonları normalize eder, daha stabil ve daha hızlı bir eğitim sürecine katkıda bulunur. Ardından, iki gizli katman (tamamen bağlı katmanlar) eklenir. İlk gizli katman 'nodes1' ile belirtilen sayıda nörona, ikinci gizli katman ise 'nodes2' ile tanımlanan sayıda nörona sahiptir. Her iki katman da aktivasyon için RELU kullanır, bu, gizli katman tanımında 'relu' ile gösterilmiştir.

Sinir ağının son kısmı çıktı katmanıdır, bu da tek bir nörona sahip bir katmandır (keras.layers.Dense(1, 'relu')). Bunun nedeni, modelin tek bir sürekli değeri (satış fiyatı) tahmin etmeyi amaçlayan bir regresyon modeli olmasıdır. Çıktı katmanı, gizli katmanlar gibi, aktivasyon için RELU aktivasyon fonksiyonunu kullanır.

Model, 'create\_nn(20, 10) ' çağırarak belirli nöron sayılarıyla gizli katmanlarda somutlaştırılır. Bu, ilk gizli katmanda 20 nöron ve ikinci katmanda 10 nöron olduğu anlamına gelir. 'model.summary()' komutu, modelin mimarisine dair kapsamlı bir genel bakış sağlar, her katmanın yapısını ve parametre sayısını içerir. Bu detaylı mimari, modelin eğitim verilerinden öğrenmesinde ve ev satış fiyatları hakkında doğru tahminler yapmasında önemli bir rol oynamaktadır. Daha sonra nöron sayıları sırasıyla yukarıda belirtilen sayılar ile değiştirilir ve model tekrar tekrar çalıştırılır.

## 2.7.2 Modelin Derlenmesi

### 2.7.2.1. Hata Fonksiyonu

Sinir ağı modellerinin geliştirilmesinde, özellikle öğrenme aşamasında, kayıp (loss) fonksiyonunun (aynı zamanda hata (error) fonksiyonu olarak da bilinir) seçimi önemlidir. Hata fonksiyonu, sinir ağının yapısının modelin öğrenme aşamasında nasıl optimize edileceğine dair bir rehber görevi görür, modelin tahmin edilen ve gerçek sonuçlar arasında gözlemediği hataya yanıt olarak parametrelerini nasıl ayarlaması gerektiğini etkili bir şekilde yönlendirir. Modelin yaptığı tahminler ile gerçek veriler arasındaki farkı sayısallaştırır ve modelin performansına ölçülebilir bir gösterge sağlar [48].

Bir sinir ağını eğitmenin özü, bu kaybı en aza indirmek etrafında döner, bu süreç modelin ağırlıklarının (weights) ve yanlılıklarının (biases) yinelemeli ayarlamalarını gerektirir. Bu nedenle uygun bir hata fonksiyonu seçimi, modelin etkili bir şekilde öğrenme yeteneğini doğrudan etkilediği için önemlidir. Farklı türdeki problemler farklı hata fonksiyonları gerektirir. Örneğin, sürekli değerleri tahmin etmeyi amaçlayan regresyon problemleri genellikle Ortalama Karesel Hata (Mean Squared Error-MSE) veya Ortalama Mutlak Hata (Mean Absolute Error-MAE) gibi hata fonksiyonlarını kullanır. Buna karşın, kategorik sonuçları tahmin etmeyi içeren sınıflandırma problemleri Çapraz Entropi (Cross-Entropy) veya İkili Çapraz Entropi (Binary Cross-Entropy) hata fonksiyonlarını kullanabilir [48].

Uygun şekilde seçilen bir hata fonksiyonu, modelin sadece bir çözüme yakınsamasını sağlamakla kalmaz, aynı zamanda bu çözümün ele alınan problem bağlamında anlamlı olmasını da sağlar. Doğru hata fonksiyonu, modelin hatalarından öğrenme yeteneği ile eğitim verilerinden görülmemiş verilere genelleme yapma yeteneği arasında etkili bir denge kurar. Bu denge hem doğru hem de sağlam olan, gerçek dünya senaryolarında güvenilir tahminler yapabilen sinir ağı modelleri oluşturmak için önemlidir [48].

Tahmin modellemesinde, bu çalışmada olduğu gibi, asıl hedef, model tarafından tahmin edilen değerlerin gerçek, gözlemlenen değerlere yakın olmasını sağlamaktır. Tahmin edilen ve gerçek değerler arasındaki farkı sayısallaştırmak ve böylece modelin eğitim sürecini daha büyük doğruluğa yönlendirmek için belirli hata fonksiyonları kullanılır. Regresyon görevlerinde yaygın olarak kullanılan iki hata fonksiyonu Ortalama Karesel Hata (MSE) ve Ortalama Mutlak Hata (MAE)'dir.

MSE, hataların karelerinin ortalamasını hesaplar, yani tahmin edilen değerler ile gerçek değer arasındaki ortalama kare farkını hesaplar. Diğer taraftan, MAE, tahmin edilen değerler ile gözlemlenen değerler arasındaki mutlak farkların ortalamasını hesaplar. Her ikisi de yaygın olarak kullanılsa da birçok çalışmada MAE yerine genellikle MSE tercih edilir. Bu tercih, MSE'nin hataları karesel olarak hesaplayarak daha büyük hatalara daha fazla ağırlık verme eğiliminde olmasından kaynaklanır. MSE'nin bu özelliği, özellikle daha büyük hataların daha önemli olduğu ve daha küçük hatalardan daha en aza indirilmesi gerektiği durumlarda faydalıdır [48].

Bu çalışmada, hata fonksiyonu olarak MSE kullanılmıştır. Daha büyük tahmin hatalarını daha etkili bir şekilde en aza indirme hedefiyle uyumlu olan MSE kullanım kararı, daha doğru ve tutarlı bir model performansı sağlamayı amaçlamaktadır. Hataların karesini azaltmaya odaklanarak, MSE, modelin tahminlerini mümkün olduğunca gerçek değerlere yaklaştırmada yardımcı olur.

#### 2.7.2.2. Optimizasyon Algoritması

Makine öğrenimindeki sinir ağı eğitim sürecinde, optimizasyon algoritmaları ayrı bir öneme sahiptir. Bu algoritmalar, modelin eğitim aşamasında hataları azaltmak amacıyla, ağırlıklar (weights) ve öğrenme hızı (learning rate) gibi sinir ağının anahtar niteliklerini değiştirmek için tasarlanmış özel yöntemlerdir. Bu optimizasyon algoritmaları, sinir ağının, üzerinde çalıştığı verilerden öğrendikçe iç parametrelerini nasıl ayarlayıp ince ayar yapacağını yönlendirmede temel bir işlev görür. Amaç, hata fonksiyonunu en aza indirerek modelin performansını iyileştirmektir [48].

Çeşitli sinir ağı modellerinde kullanılan birkaç tanınmış optimizasyon algoritması bulunmaktadır, her birinin benzersiz özellikleri ve mekanizmaları vardır. Temel bir optimizasyon algoritması olan Gradient Descent, hata fonksiyonunun gradyanını veya eğimini hesaplayarak ve kaybı en aza indirecek şekilde ters yönde hareket ederek çalışır. Stokastik Gradient Descent (SGD), her seferinde sadece tek bir eğitim örneğini kullanarak model parametrelerini güncelleyen bu yaklaşımın bir varyasyonudur, bu da yerel minimumlardan kaçmada yardımcı olabilecek bir rastgelelik derecesi ekler.

Mini-Batch Gradient Descent, Gradient Descent ve SGD'nin konseptlerini birleştirir, parametreleri tek bir veri noktası veya tüm veri seti yerine küçük, rastgele bir veri alt



kümesine göre günceller. Bu yöntem, tam toplu öğrenmenin sağlamlığı ile stokastik öğrenmenin verimliliği arasında bir denge sağlar. Momentum, başka bir optimizasyon tekniği, tutarlı gradyanlarla yönlerde hızlandıran bir momentum terimi ekleyerek SGD üzerine inşa eder, böylece yakınsama hızını iyileştirir.

Nesterov Hızlandırılmış Gradient- Nesterov Accelerated Gradient (NAG), momentum yaklaşımını gradyan hesaplamasına düzeltici bir ayarlama yaparak daha doğru ve hızlı bir yakınsama sağlayarak daha da geliştirir.

Agadrad, öğrenme hızını farklı parametrelere uyarlayarak, nadiren karşılaşılan parametreler için daha büyük ve sık karşılaşılanlar için daha küçük güncellemeler sağlar. Adagrad'ın hızla azalan öğrenme hızlarını ele alan AdaDelta, sürekli eğitim döngüleri boyunca daha sağlam hale gelir.

Son olarak, Adam (Adaptive Moment Estimation), Momentum ve RMSprop (başka bir optimizasyon algoritması) adı verilen bir algoritmanın unsurlarını birleştirerek, gradyanların ilk ve ikinci anlarına dayalı olarak öğrenme hızını uyarlayarak verimli bir optimizasyon sunar.

Bu optimizasyon algoritmalarının her birinin belirgin avantajları vardır ve belirli ihtiyaçlara ve eğitilen sinir ağı modelinin özelliklerine göre seçilirler. Etkili ve verimli bir eğitim için doğru uygulanmaları, modelin doğru ve hızlı bir şekilde öğrenme yeteneğini doğrudan etkiler.

Modeli en kısa sürede ve en verimli şekilde eğitmek için Adam en başarılı optimize edicidir [48]. Bu çalışmada optimize algoritması olarak Adam kullanılmıştır.

### 2.7.2.3. Metrik

Modelin öğrenme ve test aşamaları boyunca performansını izlemek ve ölçmek için metrikler kullanılır. Regresyon modelleri sürekli bir çıktıya sahiptir. Bu nedenle, tahmin edilen çıktı verileri ile gerçek çıktı verileri arasındaki mesafeyi hesaplayan bir metriğe ihtiyaç vardır. Ortalama Mutlak Hata (Mean Absolute Error-MAE), Ortalama Karesel Hata (Mean Squared Error-MSE), Kök Ortalama Karesel Hata (Root Mean Squared Error-RMSE) ve  $R^2$  (R-Squared-R-Kare) bu tür metriklerin örneklerindedir [48].

MAE, tahmin edilen değerler ile gerçek değerler arasındaki ortalama mutlak farkı hesaplar. Bu metrik, ortalama hata büyüklüğünün doğrudan bir göstergesi olarak sezgisel ve kolay anlaşılır bir hata ölçüsü sunar. MSE, hataların karelerinin ortalamasını ölçer ve her hatayı karesi alındığı için büyük hataları küçüklerden daha fazla vurgular. Bu, büyük hataların daha az istenildiği durumlarda özellikle faydalı olabilir.

RMSE, MSE'nin bir türevidir ve tahmin edilen değerler ile gerçek değerler arasındaki ortalama kare farklarının kökünü sağlar. RMSE, hata değerini verilerin orijinal birimlerine geri döndürerek daha yorumlanabilir hale getirir. Son olarak,  $R^2$  (R-Kare) veya belirleme katsayısı olarak da bilinen, bağımlı değişkendeki varyansın bağımsız değişkenlerden ne kadarının tahmin edilebileceğini temsil eden istatistiksel bir ölçümdür. Modelin görünmeyen örnekleri ne kadar iyi tahmin edebileceğine dair bir fikir verir.

Bu metrikler, MAE, MSE, RMSE ve  $R^2$ , regresyon modellerinde modelin tahminlerinin gerçek değerlere ne kadar yakın olduğunu nicel olarak değerlendirmede önemlidir. Ortalama hata oranlarından açıklanan varyansın oranına kadar modelin performansı hakkında çeşitli bakış açıları sunarak, modelin tahmin yeteneklerinin kapsamlı bir değerlendirilmesini sağlarlar [48].

Modelde, Kök Ortalama Karesel Hata (RMSE) ve Ortalama Mutlak Hata (MAE) metrikleri birlikte kullanıldı.

Tablo 2.8: Hata fonksiyonu, optimizasyon algoritması ve metriklerin belirlendiği koda ilişkin tablo

Python Kodu
<pre>model.compile(optimizer='adam', loss='mse', metrics=[keras.metrics.RootMeanSquaredError(), keras.metrics.MeanAbsoluteError()])</pre>

TensorFlow ve Keras kütüphaneleri kullanılarak modelin geliştirme aşamasındaki bu kod parçacığı, sinir ağını eğitime hazırlamak için kullanıldı. 'model.compile' metodu, modelin öğrenme davranışını yönetecek anahtar parametrelerin tanımlandığı modelin kurulum sürecinde temel bir işlemdir.

Öncelikle, 'optimizer' olarak 'adam' belirlenmiştir. Adam optimizasyonu, seyrek gradyanları verimli bir şekilde ele alması ve farklı problemler ve veri türlerine göre uyum sağlama kapasitesi nedeniyle sinir ağlarını eğitmede popüler bir seçimdir. Adam seçilerek, model eğitim sırasında öğrenme hızını otomatik olarak ayarlayan ve modelin performansını optimize eden bir optimizatör ile donatılmıştır.

Ardından, 'loss' fonksiyonu 'mse' olarak ayarlanmıştır, bu Ortalama Kare Hata anlamına gelir. MSE, tahmin edilen ve gerçek değerler arasındaki farkların karelerinin ortalamasını en aza indirmeyi amaçladığı regresyon problemleri için yaygın bir seçimdir. Bu hata fonksiyonu, modelin eğitim sırasında tahmin hatalarını azaltmasına rehberlik eder.

Son olarak, 'metrics' parametresi iki metrik içerir: Kök Ortalama Kare Hata (RMSE) ve Ortalama Mutlak Hata (MAE). Bu metrikler, modelin performansını değerlendirmede önemlidir. RMSE, tahmin edilen ve gerçek değerler arasındaki ortalama kare farklarının kökünü alarak modelin hatalarının ortalama büyüklüğünü ölçer. Öte yandan, MAE tahminler ve gerçek değerler arasındaki mutlak farkların ortalamasını hesaplar. Hem RMSE hem de MAE'nin dahil edilmesi, modelin doğruluğuna dair kapsamlı bir görünüm sunar; RMSE daha büyük hatalara karşı hassas olurken, MAE daha basit bir ortalama hata değeri sağlar.

Bu özel ayarlar – Adam optimizatörü, MSE hata fonksiyonu ve RMSE ve MAE metrikleri – ile model, etkili öğrenme ve doğru performans değerlendirme için ayarlanmıştır.

### 2.7.3 Modelin Eğitilmesi

Sinir ağı modelinin eğitim sürecinde, modelin verilerden öğrenmesinin gerçekleştiği aşama olan uyarlama (fitting) fazı bulunmaktadır. Bu aşama, etkili bir eğitim sağlamak için dikkatlice ayarlanması gereken birkaç anahtar parametre içerir. Bu parametreler arasında, 'epoch' sayısı, her adımda modele beslenecek örnek boyutu (batch\_size) ve model hakkında takip etmek istediğimiz bilgilerin detay düzeyini belirleyen detaylılık (verbose) değeri özellikle önemlidir.

'Epoch' sayısı, öğrenme algoritmasının tüm eğitim veri seti üzerinden kaç kez geçeceğini belirtir. Bir 'epoch', algoritmanın eğitim verilerini tamamen bir kez

işlemesini temsil eder. Doğru sayıda 'epoch' ayarlamak önemlidir; çok az 'epoch', yetersiz uyumlu bir modele neden olabilirken, çok fazla 'epoch' aşırı uyuma yol açabilir. Modelin yeni verilere iyi genelleştirmesi için bu dengeyi bulmak gereklidir [48].

'Batch size', her 'epoch' adımında modelin göreceği ve öğreneceği veri alt kümesinin boyutunu belirtir. Modelin iç parametrelerini güncellemeden önce işlenecek örnek sayısını tanımlar. 'Batch size' seçimi, modelin performansını ve hızını etkileyebilir. Daha küçük bir 'batch size' genellikle düzenleyici bir etki sağlar ve daha düşük genelleştirme hatasına yol açabilir. Ancak, modelin ağırlıklarının daha sık güncellenmesi nedeniyle öğrenme sürecini yavaşlatabilir. Diğer taraftan, daha büyük bir 'batch size', daha fazla veriyi aynı anda işleyerek daha hızlı eğitim sağlar, ancak kayıp fonksiyonunun global minimumuna daha az doğru bir şekilde yakınsama sonucu doğurabilir [48].

Uyarılama fazındaki detaylılık (verbose) parametresi, modelin eğitim sırasında çıktı olarak vereceği bilgi detay düzeyini belirleyen kullanıcı dostu bir özelliktir. Daha yüksek bir detaylılık düzeyi ayarlamak, eğitim sürecinin ilerleyişine dair daha ayrıntılı mesajlar verir ki bu, izleme ve hata ayıklama için faydalı olabilir. Daha düşük bir detaylılık düzeyi, çıktı miktarını azaltır, bu da eğitim sürecini daha az karmaşık yapar ancak potansiyel sorunları izlemeyi zorlaştırabilir.

Bu parametrelerin- 'epoch', 'batch size' ve detaylılık- ayarlanması, modelin eğitim verilerinden ne kadar iyi ve ne kadar hızlı öğrendiğini doğrudan etkileyen model uyarılama sürecinin hayati bir parçasıdır. Bu ayarların ince ayarlanması, daha verimli ve etkili model eğitime yol açabilir ve sonuç olarak daha iyi performans sağlar.

Çalışmada 'batch size' 32, 'epoch' sayısı 500 ve detaylılık (verbose) seviyesi 2 olarak ayarlandı. Ayrıca, bu değerler çalışma boyunca sabit tutuldu ve sonuçlar, gizli katmanlardaki nöron sayısını değiştirerek ve toplu normalleştirme katmanını ekleyip çıkararak izlendi.

Tablo 2.9: Modelin eğitilmesi koduna ilişkin tablo

Python Kodu
history = model.fit(X_train, y_train,

```
batch_size=32,  
epochs=500,  
verbose=2)
```

'model.fit' fonksiyonu, belirtilen parametrelerle modeli eğitmek için kullanılır. Bu fonksiyon, modelin eğitim verilerinden öğrenmesinin gerçekleştiği eğitim sürecinin merkezindedir.

X\_train ve y\_train parametreleri sırasıyla eğitim veri setinin özelliklerini ve hedef değişkenlerini temsil eder. Bunlar, modelin öğreneceği verilerdir. 'batch\_size' parametresi 32 olarak ayarlanmıştır, bu da modelin eğitim sırasında her seferinde 32 örnek işleyeceği anlamına gelir. Bu 'batch size', bir seferde işlenen veri miktarı ile modelin ağırlıklarının güncellenme sayısı arasında bir denge kurar ve hem öğrenme sürecinin hızını hem de kalitesini etkiler.

'epochs' parametresi 500 olarak ayarlanmıştır, bu da tüm eğitim veri setinin sınırından 500 kez geçirileceğini gösterir. Bu tekrarlanan işlem, modele verilerdeki desenleri daha etkili bir şekilde öğrenmesini sağlar, ancak aşırı uyuma karşı dikkatli olunmalıdır.

'verbose' parametresi 2 olarak ayarlanmıştır, bu da eğitim sırasında sağlanan çıktının detay düzeyini kontrol eder. 2 seviyesindeki detaylılık ayarı, her 'epoch' için ilerleme gösterir, ancak her 'batch' için ayrıntılı bilgi göstermez. Bu ayar, eğitim sürecinin genel ilerleyişini aşırı detay ile çıktıyı doldurmadan izlemeye yardımcı olur.

Genel olarak, 'model.fit' komutu, modelin girdi verilerine dayanarak kayıp fonksiyonunu en aza indirmek için ağırlıklarını yinelemeli olarak ayarladığı sınır ağının eğitim rejimini kapsar, bu da zamanla tahmin doğruluğunu artırır.

## Bölüm 3

### Analiz Sonuçları

Yapılan çalışmada, Ames, Iowa'daki evlerin satış fiyatlarını tahmin etmek için kapsamlı bir yapay sinir ağı (YSA) modeli geliştirildi. Bu model, bu evlerin özelliklerini kapsayan kapsamlı bir veri seti kullanarak, gayrimenkul piyasası eğilimleri ve değerlendirme konusunda içgörüler sunabilecek doğru ve güvenilir bir tahmin aracı oluşturma odaklıydı.

Sinir ağı modelinin mimarisinin merkezinde, doğrusal olmayan verilerle etkili bir şekilde başa çıkma ve derin öğrenme modellerinde yaygın bir sorun olan kaybolan gradyan sorununu önleme verimliliği nedeniyle hem gizli katmanlarda hem de çıkış katmanında aktivasyon fonksiyonu olarak Düzeltilmiş Doğrusal Aktivasyon Fonksiyonu (RELU) kullanımı yer alıyor.

Bu çalışmanın kritik bir yönü, YSA'nın gizli katmanlarındaki nöronların optimal yapılandırmasını belirlemektir. Bunu başarmak için, model, gizli katmanlardaki çeşitli nöron yapılandırmaları ile kapsamlı testlerden geçti. Özellikle, model iki gizli katmanda (20, 10), (40, 20), (60, 30), (80, 40) ve (100, 50) nöron kombinasyonlarını keşfeden 10 farklı iterasyonda çalıştırıldı. Bu iterasyonlar, bir toplu normalleştirme (batch normalization) katmanı kullanılarak ve kullanılmadan gerçekleştirildi. Bu da bu katmanın model performansı üzerindeki etkisini kapsamlı bir şekilde araştırmaya olanak tanıdı.

Farklı yapılandırmalarda toplu normalleştirme katmanının dahil edilmesi veya çıkarılması, bu katmanın gerekliliği ve etkinliği hakkında değerli içgörüler sağladı. Toplu normalleştirme katmanı, öğrenme sürecini stabilize etme ve hızlandırma yeteneği ile bilinir. Bu da onu sinir ağı tasarımında düşünülmesi gereken önemli bir bileşen haline getirir. Bu yinelemeli deneyler aracılığıyla, çalışma sadece fiyat tahmini

için en etkili nöron yapılandırmasını belirlemeyi değil, aynı zamanda toplu normalleştirme katmanının modelin genel tahmin doğruluğu ve verimliliği üzerindeki etkisini anlamayı hedefledi.

Bu kapsamlı çalışmada, ev satış fiyatlarını tahmin etmek için on farklı yapay sinir ağı modeli detaylı bir şekilde değerlendirildi. Her model, yapısında bir toplu normalleştirme katmanının varlığı ve iki gizli katmanda bulunan nöron sayısı açısından farklılık gösteriyordu. Her modelin performansı hem eğitim hem de test aşamalarında iki ana metrik olan Kök Ortalama Karese Hata (RMSE) ve Ortalama Mutlak Hata (MAE) kullanılarak değerlendirildi.

Tablo 3.1: Model sonuçlarının karşılaştırılmasına ilişkin tablo

Model	Batch Norm.	# layers_ dense 1 nodes	# layers_ dense 2 nodes	Train RMSE	Test RMSE	Train MAE	Test MAE
Model 1	No	20	10	0,0176	0,0691	0,0126	0,0431
Model 2	Yes	20	10	0,0258	0,0445	0,0153	0,0304
Model 3	No	40	20	0,0100	0,0601	0,0078	0,0422
Model 4	Yes	40	20	0,0228	0,0549	0,0127	0,0369
Model 5	No	60	30	0,0065	0,0505	0,0048	0,0364
Model 6	Yes	60	30	0,0245	0,0477	0,0109	0,0339
Model 7	No	80	40	0,0075	0,0466	0,0063	0,0340
Model 8	Yes	80	40	0,0174	0,0563	0,0111	0,0362
<b>Model 9</b>	<b>No</b>	<b>100</b>	<b>50</b>	<b>0,0026</b>	<b>0,0475</b>	<b>0,0020</b>	<b>0,0322</b>
Model 10	Yes	100	50	0,0151	0,0592	0,0085	0,0360

Modeller, Model 1'den Model 10'a kadar sistematik bir şekilde kategorize edildi. Toplu normalleştirme katmanı içermeyen Model 1, ilk gizli katmanda 20 ve ikinci gizli katmanda 10 nörona sahipti. Bu model, eğitimde 0,0176 ve testte 0,0691 RMSE değerleri ile eğitim ve test MAE değerleri sırasıyla 0,0126 ve 0,0431 olarak kaydedildi. Buna karşın, aynı nöron yapılandırmasına ancak toplu normalleştirme katmanı eklenmiş Model 2, test performansında iyileşme göstererek 0,0445 test RMSE ve 0,0304 test MAE değerlerine ulaştı, ancak eğitimde biraz daha yüksek RMSE ve MAE gösterdi.

Gizli katmanlardaki nöron sayıları arttıkça, modellerin performanslarında belirgin değişiklikler gözlemlendi. İlk gizli katmanda 40 ve ikinci gizli katmanda 20 nörona sahip Modeller 3 ve 4 bu eğilimi gösterdi ve Model 4'te toplu normalleştirme katmanının eklenmesi, Model 3'e kıyasla test RMSE ve MAE'de iyileşme sağladı.

Model 5'ten 8'e kadar olan modeller gizli katmanlardaki nöron sayılarını daha da artırdı ve (60, 30) ve (80, 40) yapılandırmalarını keşfetti. Burada, toplu normalleştirme katmanı olmayan modeller (Model 5 ve 7) genellikle daha iyi eğitim RMSE ve MAE gösterirken, toplu normalleştirme katmanı olan modeller (Model 6 ve 8) daha iyi test RMSE ve MAE'ye sahip oldu ve bu da görülmemiş verilere daha iyi genelleme yapabildiklerini gösterdi.

En iyi performans Model 9'da gözlemlendi, bu modelde toplu normalleştirme katmanı bulunmuyordu ve gizli katmanlarda en yüksek nöron sayıları olan (100, 50) ile yapılandırılmıştı. Bu model, eğitim ve test metrikleri arasında en uygun dengesini sağlayarak, 0,0475 test RMSE ve 0,0322 test MAE ile sırasıyla 0,0026 ve 0,0020 olan olağanüstü düşük eğitim RMSE ve MAE değerlerini kaydetti. Model 10, aynı nöron yapılandırmasına sahip olmasına rağmen toplu normalleştirme katmanının eklenmesi ile Model 9'un performansını geçemedi.

Model 9'un etkinliği hem eğitim hem de test aşamalarında elde ettiği performansla nicel olarak vurgulanmaktadır. Modelin veri setinden öğrendiği eğitim aşamasında, 0.0026 gibi düşük bir Kök Ortalama Kare Hata (RMSE) elde etmiştir. RMSE, tahmin edilen değerler ile gözlemlenen değerler arasındaki farkların ikinci örnek momentinin kare kökünü temsil ederek, hatanın ortalama büyüklüğünü ölçen standart bir istatistik ölçümüdür. RMSE, bir modelin doğruluğunu değerlendirmede kritik bir metriktir. Modelin tahmin ettiği değerler ile gerçek değerler arasındaki hatanın ortalama büyüklüğünü ölçer. Daha düşük bir RMSE, daha yüksek bir doğruluk gösterir ve bu durumda, 0.0026 değeri eğitim sırasında olağanüstü derecede hassas bir model olduğunu göstermektedir.

Benzer şekilde, eğitim sırasında Ortalama Mutlak Hata (MAE) 0.0020 olmuştur. MAE, tahmin edilen değerler ile gerçek sayılar arasındaki mutlak hataların ortalamasını sağlayarak, modelin tahmin doğruluğuna dair net bir değerlendirme

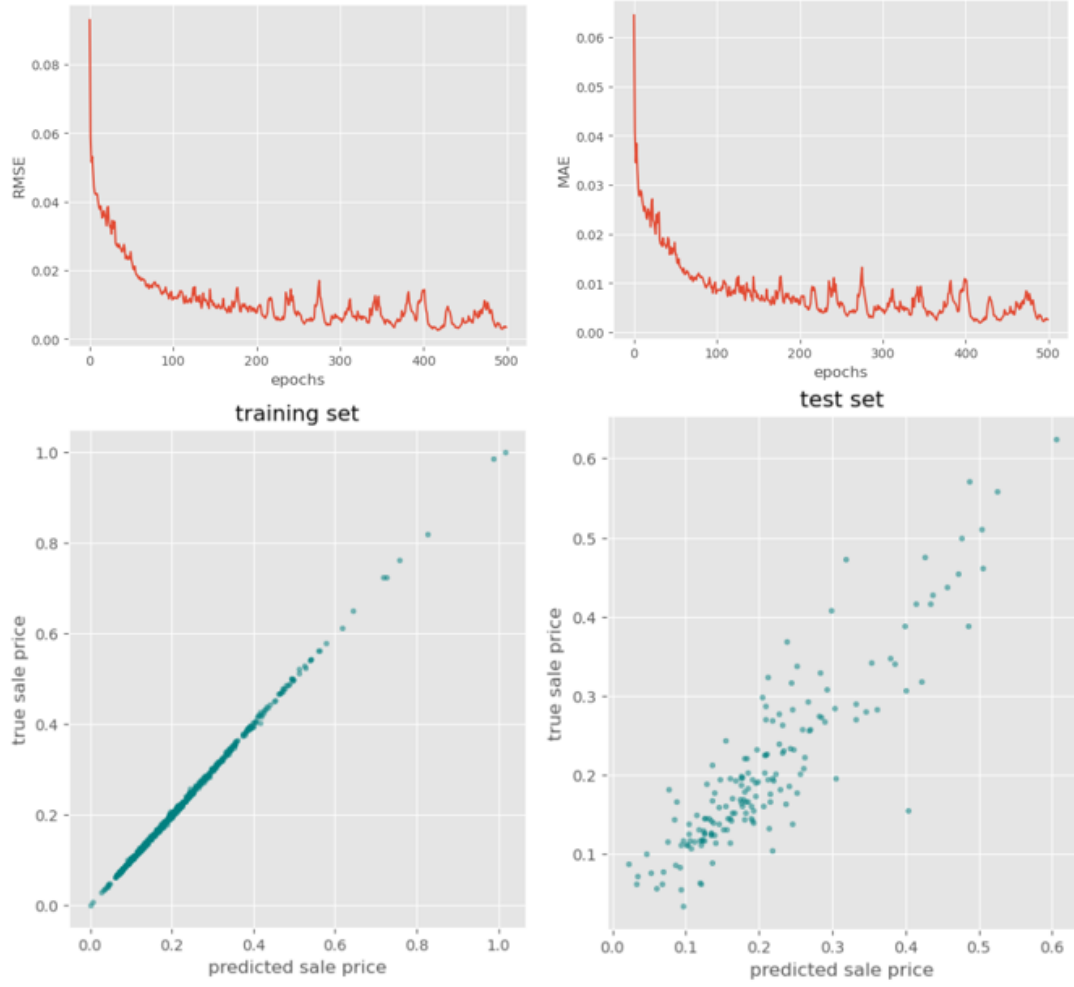


yapar. Model 9'da gözlemlenen düşük MAE, gerçek değerlerden minimum sapma ile hassas tahminler gösterir.

Ayrıca, Model 9'un sağlamlığı ve güvenilirliği, test aşamasında daha da doğrulanmaktadır. Bu aşamada, model gerçek dünya uygulamalarını simüle eden yeni, görülmemiş verilere maruz bırakılmıştır. Burada, model düşük bir test RMSE'si olan 0.0475 ve test MAE'si olan 0.0322'yi korumuştur. Eğitim RMSE'sinden biraz daha yüksek olmasına rağmen, bu değer hala yüksek bir doğruluk seviyesi ve yeni, görülmemiş verilere iyi genelleme yapabilme yeteneğini yansıtmaktadır. Bu, modelin gerçek dünya senaryolarında sağlamlık ve güvenilirliğinin önemli bir göstergesidir. Test MAE'si, eğitim MAE'sinden daha yüksek olmasına rağmen, modelin yeni verilerdeki tahmin yeteneğinde övgüye değer bir doğruluk seviyesini göstermektedir. Bu rakamlar özellikle önemlidir. Çünkü modelin sadece eğitim veri setinden etkili bir şekilde öğrenme yeteneğini değil, aynı zamanda yeni verilere iyi genelleme yapma kabiliyetini de göstermektedir.

Hem eğitim hem de test aşamalarında RMSE ve MAE'nin düşük değerleri, Model 9'un üstün tahmin yeteneğini vurgulamaktadır. İki gizli katmanlı ve (100, 50) nöronlu yapılandırması, veri seti ve belirtilen özellikler için açıkça uygun olduğunu göstererek, doğru ve güvenilir veri tahminleri için olağanüstü bir seçenek haline gelmiştir.

Bu detaylı analiz, verilen veri seti ve problem için, yapay sinir ağı modelinin ev satış fiyatlarını etkili bir şekilde tahmin etmesi için optimal yapılandırmanın, Model 9'da örneklenen gibi, ilk gizli katmanda 100 ve ikinci gizli katmanda 50 nöron olacak şekilde toplu normalleştirme katmanı olmadan kurulması gerektiğini göstermektedir. Bu yapılandırma, modelin tahminlerini en doğru şekilde yapmasını sağlayarak, eğitim doğruluğu ile yeni, görülmemiş verilere genelleme yapma yeteneği arasında iyi bir denge kurmuştur.



Şekil 3.1: Sonuçlara ilişkin grafikler

## Bölüm 4

Bu bölüm, bu çalışmanın tüm aşamalarını temel alarak sonuçların tartışılmasını ve gelecek araştırmalar için bir yol haritası sunmayı ve gayrimenkul fiyatlandırması için makine öğrenimi alanında daha ileri gelişmelere rehberlik etmeyi amaçlamaktadır.

### Tartışma ve Sonuç

Bu bölüm, bu çalışmanın temel çalışmalarını temel alarak gelecek araştırmalar için bir yol haritası sunmayı ve gayrimenkul fiyatlandırması için makine öğrenimi alanında daha ileri gelişmelere rehberlik etmeyi amaçlamaktadır.

Bu çalışmanın amacı, yapay sinir ağı modeli için optimal gizli katman nöron sayısını belirlemek ve toplu normalleştirme katmanının doğru tahminler için gerekli olup olmadığını anlamaktır. Model, gizli katmanlardaki nöron sayılarının değiştiği, toplu normalleştirme katmanı kullanılarak ve kullanılmayarak 10 farklı şekilde çalıştırıldı. Yapay sinir ağı modelleri, verilerdeki karmaşık desenleri yakalayabilme yetenekleri nedeniyle ev satış fiyatlarını tahmin etmek için popüler bir makine öğrenimi tekniği haline gelmiştir. Ancak, yapay sinir ağı modelindeki gizli katman nöronlarının optimal sayısını seçmek, doğru tahminler elde etmek için kritik öneme sahiptir.

Gizli katmanlardaki nöron sayısı, yapay sinir ağı modelinin girdi verilerindeki özellikleri yakalama ve bu verilerden öğrenme kapasitesini belirler. Eğer nöron sayısı çok az ise, model verileri yetersiz uyum sağlayabilir ve bu da kötü performansa yol açabilir. Tersine, nöron sayısı çok fazla ise, model verilere aşırı uyum sağlayabilir ve bu da yeni, görülmemiş verilerde genelleme ve doğrulukta düşüşe neden olabilir [49].

Bu nedenle, yapay sinir ağı modelindeki gizli katmanların optimal nöron sayısını belirlemek, en iyi performansı elde etmek için hayati öneme sahiptir. Farklı gizli katman nöron sayılarına sahip modelin performans ölçümlerini karşılaştırarak, yetersiz

uyum ve aşırı uyum arasında bir denge sağlayan optimal nöron sayısını belirlemek mümkündür [19].

Ayrıca toplu normalleştirme katmanı, yapay sinir ağı modellerinin performansını ve istikrarını iyileştirmek için kullanılan bir tekniktir. Girdi grubunun ortalamasını çıkararak ve standart sapmasına bölerek önceki katmanın çıktısını normalleştirir. Bu sayede sonraki katmana giren değerlerin merkezlenmesi ve benzer bir ölçeğe sahip olması sağlanarak modelin performansı iyileştirilebilir ve aşırı uyum önlenir [50].

Ancak, toplu normalleştirme katmanının etkinliği, veri setine ve model mimarisine bağlı olarak değişebilir. Bazı durumlarda, performansta önemli bir iyileşme sağlamayabilir veya performansı azaltabilir. Bu nedenle bazı durumlarda toplu normalleştirme katmanının ayarlanarak ek parametreler eklenebilir. Örneğin momentum parametresi veya öğrenme oranı gibi [50].

Ev satış fiyatlarını tahmin etmek bağlamında, gizli katman nöronlarının optimal sayısını doğru bir şekilde belirlemek; gayrimenkul profesyonelleri, alıcılar ve satıcılar için daha doğru tahminler ve daha iyi bilgilendirilmiş karar vermeleri konusunda yardımcı olabilir. Ayrıca, yanlış fiyatlandırma veya bir mülkün değerinin düşük tahmin edilmesi nedeniyle oluşabilecek finansal kayıp riskini azaltmaya yardımcı olabilir [49]. Ancak, doğru tahminler elde etmek için toplu normalleştirme katmanının kullanılması her zaman gerekli olmayabilir. Çalışmanın sonuçları, toplu normalleştirme katmanının dahil edilmesinin yapay sinir ağı modelinin performansını önemli ölçüde iyileştirmediğini göstermektedir. Bu, veri setinin doğası nedeniyle olabileceği gibi toplu normalleştirme katmanı tarafından sağlanan ek girdi normalleştirmesine ihtiyaç da duymayabilir [50].

Bu nedenle, toplu normalleştirme katmanının etkinliğini duruma göre değerlendirmek ve modelin performansında önemli bir iyileşme sağlayıp sağlamadığını belirlemek önemlidir. Bu, modelin toplu normalleştirme katmanı ile ve olmadan performansını karşılaştırarak ve en iyi sonuçları sağlayan seçeneği belirleyerek yapılabilir [51].

Sonuçlar, tasarlanan model için gizli katmanlardaki optimal nöron sayısının toplu normalleştirme katmanı olmadan (100, 50) olduğunu gösterdi. Bu, model çalışmalarının RMSE ve MAE performans ölçümleri karşılaştırılarak belirlendi. Toplu normalleştirme katmanı olmadan (100, 50) nöron kullanan Model 9, 0.0026 eğitim

RMSE, 0.0475 test RMSE, 0.0020 eğitim MAE ve 0.0322 test MAE ile en iyi performansı gösterdi.

Sonuçlar, gizli katmanlarda daha fazla nöron kullanmanın modelin ev satış fiyatlarını tahmin etmedeki performansını iyileştirebileceğini öne sürmektedir. Ancak, toplu normalleştirme katmanının eklenmesi, modelin performansını önemli ölçüde iyileştirmedir.

Sonuç olarak, bu çalışmanın sonuçları, regresyon tabanlı makine öğrenimi yöntemlerinin, özellikle yapay sinir ağı modellerinin, ev satış fiyatlarını tahmin etmede etkili olabileceğini göstermektedir. Gizli katmanlardaki optimal nöron sayısının toplu normalleştirme katmanı olmadan (100, 50) olduğu bulundu. Bu sonuçlar, gizli katmanlarda daha fazla nörona sahip bir modelin tahmin doğruluğunu iyileştirebileceğini öne sürmektedir.

Bu çalışmanın bulguları, özellikle yapay sinir ağları kullanılarak, regresyon tabanlı makine öğrenimi yöntemlerinin ev satış fiyatlarını tahmin etmedeki etkinliğini vurgulamaktadır. İleri araştırmalar, modelin tahmin yeteneklerine daha derin içgörüler veya daha nüanslı bir anlayış sağlayabilecek alternatif performans ölçümlerini keşfedebilir. Ayrıca, bu çalışmanın kapsamının ötesindeki farklı parametrelerle deneyler yapmak, modelin doğruluğunu geliştirebilir. Bu, farklı katman tasarımları veya aktivasyon fonksiyonları gibi yapay sinir ağı yapılandırılmalarını incelemeyi içerebilir.

Başka bir gelecek araştırma alanı, veri ön işleme tekniklerinin ve özellik seçimi yöntemlerinin model performansı üzerindeki etkisini incelemek olabilir. Bu yöntemlerin modelin verileri yakalama ve öğrenme yeteneğini nasıl etkilediğini anlamak, daha sağlam ve doğru tahminlere yol açabilir.

Ayrıca, modeli gerçek zamanlı veri analizini içerecek şekilde genişletmek, gayrimenkul profesyonelleri için dinamik fiyatlandırma içgörülerini sunabilir. Ekonomik göstergeler veya demografik bilgiler gibi ek veri kaynaklarını entegre etmek de modelin tahmin gücünü artırabilir.

Son olarak, diğer makine öğrenimi yaklaşımları, örneğin destek vektör makineleri veya karar ağaçları ile karşılaştırmalı çalışmalar, ev fiyat tahmini için en etkili

tekniklerin kapsamlı bir anlayışını sağlayabilir. Bu tür karşılaştırmalı analizler, alanı ilerletmede ve pratik uygulamaları yönlendirmede önemli olacaktır.

# Kaynaklar

- [1] S. J. A. P. Zavei and M. M. Jusan, "Exploring Housing Attributes Selection based on Maslow's Hierarchy of Needs," *Procedia Soc Behav Sci*, vol. 42, 2012, doi: 10.1016/j.sbspro.2012.04.195.
- [2] D. L. Foley, "The sociology of housing.," *Annual review of sociology: volume 6*, 1980, doi: 10.1146/annurev.so.06.080180.002325.
- [3] V. Maliene and N. Malys, "High-quality housing-A key issue in delivering sustainable communities," *Build Environ*, vol. 44, no. 2, 2009, doi: 10.1016/j.buildenv.2008.04.004.
- [4] E. Mifflin and R. Wilton, "No place like home: Rooming houses in contemporary urban context," *Environ Plan A*, vol. 37, no. 3, 2005, doi: 10.1068/a36119.
- [5] S. Farhan, S. Alzamly, and A. Allaami, "The effect of the phenomenon of housing transition and housing fragmentation on the housing need for the cities of Al Hayy and Al Azizia," *Journal of STEPS for Humanities and Social Sciences*, vol. 1, no. 1, 2022, doi: 10.55384/2790-4237.1007.
- [6] J. R. Coakley and C. E. Brown, "Artificial neural networks in accounting and finance: modeling issues," *International Journal of Intelligent Systems in Accounting, Finance & Management*, vol. 9, no. 2, 2000, doi: 10.1002/1099-1174(200006)9:2<119::aid-isaf182>3.0.co;2-y.
- [7] N. Nghiep and C. Al, "Predicting housing value: A comparison of multiple regression analysis and artificial neural networks," *Journal of real estate research*, vol. 22, no. 3, 2001.

- [8] A. Khalafallah, "Neural Network Based Model for Predicting Housing Market Performance," *Tsinghua Sci Technol*, vol. 13, no. SUPPL. 1, 2008, doi: 10.1016/S1007-0214(08)70169-X.
- [9] S. C. Goh, "Toward a learning organization: The strategic building blocks," *SAM Advanced Management Journal*, vol. 63, no. 2, 1998.
- [10] H. Selim, "Determinants of house prices in Turkey: Hedonic regression versus artificial neural network," *Expert Syst Appl*, vol. 36, no. 2 PART 2, 2009, doi: 10.1016/j.eswa.2008.01.044.
- [11] A. Pannu, "Artificial Intelligence and its Application in Different Areas," *International Jpurnal of Engineering and Innovoative Technology*, vol. 4, no. 10, 2015.
- [12] K. Alanne and S. Sierla, "An overview of machine learning applications for smart buildings," *Sustainable Cities and Society*, vol. 76, 2022. doi: 10.1016/j.scs.2021.103445.
- [13] E. Alzain, A. S. Alshebami, T. H. H. Aldhyani, and S. N. Alsubari, "Application of Artificial Intelligence for Predicting Real Estate Prices: The Case of Saudi Arabia," *Electronics (Switzerland)*, vol. 11, no. 21, 2022, doi: 10.3390/electronics11213448.
- [14] C. V. Ramamoorthy, S. Shekhar, and V. Garg, "Software Development Support for AI Programs," *Computer (Long Beach Calif)*, vol. 20, no. 1, 1987, doi: 10.1109/MC.1987.1663354.
- [15] R. O. Duda and E. H. Shortliffe, "Expert systems research," *Science*, vol. 220, no. 4594, 1983. doi: 10.1126/science.6340198.
- [16] E. Turban, "Review of Expert Systems Technology," *IEEE Trans Eng Manag*, vol. 35, no. 2, 1988, doi: 10.1109/17.6007.
- [17] K. ArulRaj, M. Karthikeyan, and D. Narmatha, "A View of Artificial Neural Network Models in Different Application Areas," in *E3S Web of Conferences*, 2021. doi: 10.1051/e3sconf/202128703001.



- [18] T. L. Fine, “Fundamentals of Artificial Neural Networks [Book Reviews],” *IEEE Trans Inf Theory*, vol. 42, no. 4, 2005, doi: 10.1109/tit.1996.508868.
- [19] N. Lange, C. M. Bishop, and B. D. Ripley, “Neural Networks for Pattern Recognition.,” *J Am Stat Assoc*, vol. 92, no. 440, 1997, doi: 10.2307/2965437.
- [20] M. Islam, G. Chen, and S. Jin, “An Overview of Neural Network,” *American Journal of Neural Networks and Applications*, vol. 5, no. 1, 2019, doi: 10.11648/j.ajna.20190501.12.
- [21] D. W. Pearson and G. Dray, “Applications of artificial neural networks,” in *Systems: Theory and Practice*, 1998. doi: 10.1007/978-3-7091-6451-8\_11.
- [22] R. Garcia, V. Sreekanti, N. Yadwadkar, and others, “Context: The missing piece in the machine learning lifecycle,” 2018.
- [23] B. Pérez-Sánchez, O. Fontenla-Romero, B. Guijarro-Berdiñas, and D. Martínez-Rego, “An online learning algorithm for adaptable topologies of neural networks,” *Expert Syst Appl*, vol. 40, no. 18, 2013, doi: 10.1016/j.eswa.2013.06.066.
- [24] D. Gibson, V. Kovanovic, D. Ifenthaler, S. Dexter, and S. Feng, “Learning theories for artificial intelligence promoting learning processes,” *British Journal of Educational Technology*, vol. 54, no. 5, 2023, doi: 10.1111/bjet.13341.
- [25] Y. Pan, S. W. Sung, and J. H. Lee, “Data-based construction of feedback-corrected nonlinear prediction model using feedback neural networks,” *Control Eng Pract*, vol. 9, no. 8, 2001, doi: 10.1016/S0967-0661(01)00050-8.
- [26] J. Lee, “Analysis of precision and accuracy in a simple model of machine learning,” *Journal of the Korean Physical Society*, vol. 71, no. 12, 2017, doi: 10.3938/jkps.71.866.
- [27] P. Rossini, “Application of Artificial Neural Networks to the Valuation of Residential Property,” in *Third Annual Pacific-Rim Real Estate Society Conference, Palmerston New Zealand*, 1997.

- [28] E. Bellodi, R. Zese, F. Riguzzi, and E. Lamma, “Introduction to Machine Learning,” in *Machine Learning and Non-volatile Memories*, 2022. doi: 10.1007/978-3-031-03841-9\_1.
- [29] D. Martinez, “Architecture for machine learning techniques to enable augmented cognition in the context of decision support systems,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2014. doi: 10.1007/978-3-319-07527-3\_14.
- [30] P. Kadlec and B. Gabrys, “Architecture for development of adaptive on-line prediction models,” *Memet Comput*, vol. 1, no. 4, 2009, doi: 10.1007/s12293-009-0017-8.
- [31] Q. Wang, Y. Ma, K. Zhao, and Y. Tian, “A Comprehensive Survey of Loss Functions in Machine Learning,” *Annals of Data Science*, vol. 9, no. 2, 2022, doi: 10.1007/s40745-020-00253-5.
- [32] D. Eckhard, A. S. Bazanella, C. R. Rojas, and H. Hjalmarsson, “Cost function shaping of the output error criterion,” *Automatica*, vol. 76, 2017, doi: 10.1016/j.automatica.2016.10.015.
- [33] J. Bell, “What is machine learning?,” in *Machine Learning and the City: Applications in Architecture and Urban Design*, 2022. doi: 10.1007/978-3-319-18305-3\_1.
- [34] T. Kluyver *et al.*, “Jupyter Notebooks—a publishing format for reproducible computational workflows,” in *Positioning and Power in Academic Publishing: Players, Agents and Agendas - Proceedings of the 20th International Conference on Electronic Publishing, ELPUB 2016*, 2016. doi: 10.3233/978-1-61499-649-1-87.
- [35] F. Ersoz, T. Ersoz, and M. Soydan, “Research on Factors Affecting Real Estate Values by Data Mining,” *Baltic Journal of Real Estate Economics and Construction Management*, vol. 6, no. 1, 2018, doi: 10.2478/bjreecm-2018-0017.

- [36] S. C. Bourassa, D. R. Haurin, J. L. Haurin, M. Hoesli, and J. Sun, "House price changes and idiosyncratic risk: The impact of property characteristics," *Real Estate Economics*, vol. 37, no. 2, 2009, doi: 10.1111/j.1540-6229.2009.00242.x.
- [37] A. Ilyas, S. M. Park, L. Engstrom, G. Leclerc, and A. MÄ...dry, "Datamodels: Predicting Predictions from Training Data," in *Proceedings of Machine Learning Research*, 2022.
- [38] A. H. Bacaksiz and E. Esgin, "Extraction of Numerical data from Categorical Data Set and Artificial Neural Networks," in *3rd International Symposium on Multidisciplinary Studies and Innovative Technologies, ISMSIT 2019 - Proceedings*, 2019. doi: 10.1109/ISMSIT.2019.8932767.
- [39] S. García, J. Luengo, and F. Herrera, "Data Preprocessing in Data Mining," *Intelligent Systems Reference Library*, vol. 72, 2015.
- [40] A. Jain *et al.*, "Overview and Importance of Data Quality for Machine Learning Tasks," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2020. doi: 10.1145/3394486.3406477.
- [41] I. Nino-Adan, E. Portillo, I. Landa-Torres, and D. Manjarres, "Normalization Influence on ANN-Based Models Performance: A New Proposal for Features' Contribution Analysis," *IEEE Access*, vol. 9, 2021, doi: 10.1109/ACCESS.2021.3110647.
- [42] M. Vogt, "An Overview of Deep Learning and Its Applications," 2019. doi: 10.1007/978-3-658-23751-6\_17.
- [43] A. Anaya-Isaza, L. Mera-Jiménez, and M. Zequera-Diaz, "An overview of deep learning in medical imaging," *Informatics in Medicine Unlocked*, vol. 26. 2021. doi: 10.1016/j.imu.2021.100723.
- [44] Q. Zhao and Z. Shang, "Deep learning and Its Development," in *Journal of Physics: Conference Series*, 2021. doi: 10.1088/1742-6596/1948/1/012023.

- [45] M. Romero, Y. Interian, T. Solberg, and G. Valdes, “Training Deep Learning models with small datasets,” *arXiv*. 2019.
- [46] F. Schorfheide and K. I. Wolpin, “On the use of holdout samples for model selection,” in *American Economic Review*, 2012. doi: 10.1257/aer.102.3.477.
- [47] Y. Xu and R. Goodacre, “On Splitting Training and Validation Set: A Comparative Study of Cross-Validation, Bootstrap and Systematic Sampling for Estimating the Generalization Performance of Supervised Learning,” *J Anal Test*, vol. 2, no. 3, 2018, doi: 10.1007/s41664-018-0068-2.
- [48] J. Heaton, “Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning,” *Genet Program Evolvable Mach*, vol. 19, no. 1–2, 2018, doi: 10.1007/s10710-017-9314-z.
- [49] V. Teslyuk, A. Kazarian, N. Kryvinska, and I. Tsmots, “Optimal artificial neural network type selection method for usage in smart house systems,” *Sensors (Switzerland)*, vol. 21, no. 1, 2021, doi: 10.3390/s21010047.
- [50] P. L. Kevin Ding, S. Martin, and B. Li, “Improving batch normalization with skewness reduction for deep neural networks,” in *Proceedings - International Conference on Pattern Recognition*, 2020. doi: 10.1109/ICPR48806.2021.9412949.
- [51] S. Santurkar, D. Tsipras, A. Ilyas, and A. Madry, “How does batch normalization help optimization?,” in *Advances in Neural Information Processing Systems*, 2018.