# ÖNSÖZ VE TEŞEKKÜR

Bu tez çalışması, firmaların geçmişte yapmış oldukları satış verilerinden faydalanarak gelecekteki satış miktarları tahmin edilmeye çalışılmıştır.

Tez çalışmamızda desteğini esirgemeyen, çalışmalarıma yön veren, bana güvenen ve yüreklendiren danışmanım Doç. Dr. Sevinç İLHAN OMURCA’ya sonsuz teşekkürlerimizi sunarız.

Tez çalışmamızın tüm aşamalarında bilgi, anlayış ve destekleriyle katkıda bulunan hocam Arş. Gör. Ekin EKİNCİ’ ye teşekkür ediyoruz.

Mayıs – 2019 Mehmet Emin ÖZ, Arda CİÇİN

Bu dokümandaki tüm bilgiler, etik ve akademik kurallar çerçevesinde elde edilip sunulmuştur. Ayrıca yine bu kurallar çerçevesinde kendime ait olmayan ve kendimin üretmediği ve başka kaynaklardan elde edilen bilgiler ve materyaller (text, resim, şekil, tablo vb.) gerekli şekilde referans edilmiş ve dokümanda belirtilmiştir.

Öğrenci No: 150202085

Adı Soyadı: Mehmet Emin ÖZ

İmza:…………………………………..

Öğrenci No: 150202086

Adı Soyadı: Arda CİÇİN

İmza:…………………………………..

# İÇİNDEKİLER

[ÖNSÖZ VE TEŞEKKÜR i](#_Toc10199923)

[İÇİNDEKİLER iii](#_Toc10199924)

[ŞEKİLLER DİZİNİ v](#_Toc10199925)

[TABLOLAR DİZİNİ vii](#_Toc10199926)

[SİMGELER VE KISALTMALAR DİZİNİ viii](#_Toc10199927)

[ÖZET ix](#_Toc10199928)

[ABSTRACT x](#_Toc10199929)

[1. GİRİŞ 1](#_Toc10199930)

[2. GENEL TANIMLAR 2](#_Toc10199931)

[2.1. Veri Madenciliği 2](#_Toc10199932)

[2.2. Makine Öğrenmesi 3](#_Toc10199933)

[2.2.1. Gözetimli Öğrenme 3](#_Toc10199934)

[2.2.2. Gözetimsiz Öğrenme 5](#_Toc10199935)

[2.3. Zaman Serileri 5](#_Toc10199936)

[2.3.1. Zaman Serisi Bileşenleri 6](#_Toc10199937)

[3. LİTERATÜR TARAMASI 9](#_Toc10199938)

[4. KULLANILAN TEKNOLOJİLER VE AÇIKLAMALARI 17](#_Toc10199939)

[4.1. WEKA 17](#_Toc10199940)

[4.2. LibreOffice Calc 18](#_Toc10199941)

[4.3. Python 19](#_Toc10199942)

[4.4. Spyder 19](#_Toc10199943)

[4.5. Google Colab 20](#_Toc10199944)

[5. VERİ ANALİZİ 21](#_Toc10199945)

[5.1. Veri Tipleri 21](#_Toc10199946)

[5.2. Veri Kümesinde Karşılaşılabilecek Hatalı Veriler 22](#_Toc10199947)

[5.3. Veriyi Analiz Etmek için Yöntemler 23](#_Toc10199948)

[5.3.1. Kutu Diyagramı Yöntemi (Box and Whisker Plot) 24](#_Toc10199949)

[5.3.2. Standart Sapma ve Varyans 25](#_Toc10199950)

[5.4. Hatalı Verilerin Düzeltilmesi için Temel Yöntemler 26](#_Toc10199951)

[5.4.1. Kutulama Tekniği 27](#_Toc10199952)

[5.4.2. Kümeleme Tekniği 28](#_Toc10199953)

[5.4.3. Regresyon 28](#_Toc10199954)

[5.5. Normalizasyon 28](#_Toc10199955)

[5.5.1. Min-Max Normalizasyonu 29](#_Toc10199956)

[5.5.2. Z-Score Normalizasyonu 29](#_Toc10199957)

[5.5.3. Ondalık Ölçekleme ile Normalizasyon 29](#_Toc10199958)

[6. VERİ KÜMESİ 31](#_Toc10199959)

[6.1. Veri Kümesi İçeriği 31](#_Toc10199960)

[6.1.1. Eğitim kümesi 31](#_Toc10199961)

[6.1.2. Test Kümesi 33](#_Toc10199962)

[6.2. Weka ve LibreOffice Calc ile Veri Kümesi Analizi 34](#_Toc10199963)

[6.3. Veri Temizleme 39](#_Toc10199964)

[6.3.1. Python’da Box Plot Yöntemi ile Hatalı Verilerin Analizi ve Temizlenmesi 40](#_Toc10199965)

[6.4. Veriyi Etiketli Veri Haline Getirme 46](#_Toc10199966)

[7. LİTERATÜRDE BAHSEDİLEN YÖNTEMLER 48](#_Toc10199967)

[7.1. Regresyon Analizi 48](#_Toc10199968)

[7.1.1. Basit Lineer Regresyon Modeli 48](#_Toc10199969)

[7.1.2. Lojistik Regresyon Analizi 50](#_Toc10199970)

[7.2. Destek Vektör Makineleri (SVM) 51](#_Toc10199971)

[7.2.1. Doğrusal Destek Vektör Makineleri 51](#_Toc10199972)

[7.2.2. Doğrusal Olmayan Destek Vektör Makineleri 52](#_Toc10199973)

[7.3. Yapay Sinir Ağları Yöntemi 52](#_Toc10199974)

[7.3.1. Yapay Sinir Ağları Yapısı 52](#_Toc10199975)

[7.3.2. Yapay Sinir Ağlarının Türleri 53](#_Toc10199976)

[7.3.3. Yapay Sinir Ağları Öğrenme Şekillerine Göre Sınıflandırma 55](#_Toc10199977)

[7.4. Tekrarlamalı Sinir Ağları (RNN) 57](#_Toc10199978)

[7.5. Uzun Kısa Süreli Bellek (LSTM) 57](#_Toc10199979)

[7.6. Zaman Serisi Analizi Yöntemleri 61](#_Toc10199980)

[7.6.1. Otoregresif (AR) Modelleri 61](#_Toc10199981)

[7.6.2. Hareketli Ortalama (MA) Modelleri 62](#_Toc10199982)

[7.6.3. Otoregresif hareketli ortalama (ARMA) Modelleri 62](#_Toc10199983)

[7.6.4. Box-Jenkins Modeli (ARIMA) 63](#_Toc10199984)

[7.6.5. Vektör Otoregresyon (VAR) Analizi 64](#_Toc10199985)

[7.7. SVR (Support Vector Regression) 64](#_Toc10199986)

[7.8. Ridge Regresyon Analizi 66](#_Toc10199987)

[8. MODEL PERFORMANS ÖLÇÜTLERİ 68](#_Toc10199988)

[8.1. Hata Matrisi (Confusion Matrix) 68](#_Toc10199989)

[9. VERİ SETİ ÜZERİNDE MODEL PERFORMANS ÖLÇÜMLERİ 70](#_Toc10199990)

[9.1. LSTM Modeli 70](#_Toc10199991)

[9.2. Ridge Yöntemi 72](#_Toc10199992)

[10. SONUÇ 75](#_Toc10199993)

[11. KAYNAKÇA 76](#_Toc10199994)

# ŞEKİLLER DİZİNİ

Şekil 2.1. Doğrusal trend örneği [7] 6

Şekil 2.2. D. olmayan trend örneği [7] 6

Şekil 2.3. Mevsimsel dalgalanma grafiği [7] 7

Şekil 2.4. Döngüsel Dalgalanma Örneği [8] 7

Şekil 2.5. Düzensiz (rassal) hareket örneği [9] 8

Şekil 4.1. Weka’da veri çekme ve veri özelliklerini gösterme örneği [36] 17

Şekil 4.2. LibreOffice Calc’da elde olan veriden grafik çizdirme örneği [39] 18

Şekil 4.3. Örnek bir Python uygulamasının Google Colab üzerinden çalıştırılması 20

Şekil 5.1. Örnek box-plot çizimi 25

Şekil 6.1. Eğitim verisinden örnek bir bölüm 32

Şekil 6.2. Firma ID’lerinin bulunduğu veri kümesinden örnek bölüm 32

Şekil 6.3. Ürünlere ait ID ve kategori bilgileri 33

Şekil 6.4. Kategoriler ve bu kategorilerin ID bilgileri hakkında bir örnek 33

Şekil 6.5. Test kümesinden bir örnek 34

Şekil 6.6. LibreOffice Calc’da firma ID’lerinin çizgi grafiği ile doğruluk kontrolü 35

Şekil 6.7. LibreOffice Calc’da “item\_id” özelliğinin sütun grafiği ile doğruluk kontrolü 35

Şekil 6.8. “item\_category\_id” özelliğinin doğruluk kontrolü 36

Şekil 6.9. Weka uygulamasında eğitim kümesinin genel görünümü 36

Şekil 6.10. “item\_cnt\_day” özelliğinin weka üzerinden genel görünümü 37

Şekil 6.11. Weka ile çizdirilen veri dağılım grafiği (günlük ürün satış miktarı için) 38

Şekil 6.12. Weka ile çizdirilen veri dağılım grafiği (ürün fiyatları için) 39

Şekil 6.13. Verilerin okutulması ve veriler hakkında genel bilgi için Python kodları 40

Şekil 6.14. Eğitim kümesindeki hatalı verileri temizlemeden önceki verilerin analizi 40

Şekil 6.15. Verinin Box Plot Yöntemi ile analizi için gerekli Python kodları 41

Şekil 6.16. Şekil 5.15’deki kodun çıktısı 41

Şekil 6.17. Python’da veri kümemizdeki günlük ürün satışı ve ürün fiyatlarının medyan değeri 42

Şekil 6.18. Aykırı değerlere göre veri kümesinin tekrar düzenlenmesi 43

Şekil 6.19. Aykırı değerlere göre veri kümesinin tekrar düzenlenmesi 43

Şekil 6.20. Günlük satılan ürün miktarı fiyatlarının yapılan aykırı veri temizleme işlemi sonrası box plot yöntemi ile gösterimi 43

Şekil 6.21. Ürün fiyatlarının yapılan aykırı veri temizleme işlemi sonrası box plot yöntemi ile gösterimi 44

Şekil 6.22. Eğitim verimizde hatalı verileri temizledikten sonraki verilerin dağılımı 44

Şekil 6.23 Ön işleme sonrası veri setinin boyutu 45

Şekil 6.24. Train kümesi sütunlarının tipleri 45

Şekil 6.25. Date tipini değiştirme 46

Şekil 6.26. Zaman serisi eğitim kümesini etiketli hale getirme 46

Şekil 6.27. Etiketli zaman serisi 47

Şekil 7.1. En küçük kareler yöntemi için kullanılacak veri tablosu [47] 49

Şekil 7.2. Şekil 7.1’deki verilerin dizilme örneği. [48] 49

Şekil 7.3. 7.2'deki dizilen verilerin üzerine regresyon doğrusu oluşturma örneği. [48] 49

Şekil 7.4. Karar doğrusunu gösterimi [54] 51

Şekil 7.5. Sınır düzlemlerini gösterme 52

Şekil 7.6. SVM için veri girişleri 52

Şekil 7.7. Yapay sinir ağları örnek modeli [57] 53

Şekil 7.8. İleri beslemeli YSA [60] 54

Şekil 7.9. Geri beslemeli YSA [61] 55

Şekil 7.10. Eğiticili öğrenme blok diyagramı [62] 56

Şekil 7.11. Eğiticisiz öğrenme blok diyagramı 56

Şekil 7.12. Özdenetimli Öğrenme blok diyagramı 57

Şekil 7.13. RNN örneği [84] 57

Şekil 7.14. LSTM örneği 58

Şekil 7.15. Tahn tabakası [73] 59

Şekil 7.16. Yeni girdi ekleme [75] 59

Şekil 7.17. Yt serisinin VAR gösterimi [77] 64

Şekil 7.18. SVR örneği [79] 65

Şekil 7.19. . Ridge regresyon değişken katsayılar matrisi formülü [78] 66

Şekil 9.1. Verisetinden gereksiz sütünları silme 70

Şekil 9.2. Eğitim kümesini test ve eğitim kümesi olarak ayırma 71

Şekil 9.3. LSTM model oluşturma 71

Şekil 9.4. LSTM ile veriyi eğitme 71

Şekil 9.5. LSTM modeli doğruluk değeri 72

Şekil 9.6. LSTM modei eğitim ve test kümeleri loss grafiği 72

Şekil 9.7. Ridge yöntemi ile modeli fit etme 73

Şekil 9.8. Ridge modeli test etme 73

Şekil 9.9. Ridge regreston doğrusu 74

# TABLOLAR DİZİNİ

Tablo 2.1 Eğitim Kümesi Örneği 13

Tablo 2.2 Örnek Test Kümesi 13

Tablo 8.1. Hata Matrisi (Confusion Matrix) [34] 77

# SİMGELER VE KISALTMALAR DİZİNİ

**σ2** : Varyans sembolü

**σ** : Standart sapma

x̄ : x özelliğinin aritmetik ortalaması

β : Regresyon katsayısı

ε : Tesadüfi hata terimi

**Kısaltmalar**

IDE : Integrated Development Environment (Tümleşik Geliştirme Ortamı)

ID : Identification/Identity/Identifier (Kimlik /Benzersiz Tanıtıcı)

MAPE : Mean Absolute Percentage Error (Ortalama Mutlak Yüzde Hata)

MSE : Mean Squared Error (Ortalama Kare Hata)

RMSE : Root Mean Square Error (Kök Ortalama Kare Hata)

RBF : Radial Basis Function (Radayal Bazlı Fonksiyon)

SVM : Support Vector Machines (Destek Vektör Makineleri)

RNN : Recurrent Neural Network (Tekrarlanan Sinir Ağları)

LSTM : Long-Short Time Memory (Uzun-Kısa Süreli Bellek)

ACC : Accuracy (Doğruluk)

ERR : Error Rate (Hata Oranı)

SPC : Specificity (Belirleyicilik)

TPR : True Positive Rate (Doğru Pozitif Oranı)

AR : Autoregressive Model (Otoregresif Modeller)

MA : Moving Average Model (Hareketli Ortalama Modeli)

ARMA : Autoregressive Moving Average Model (Otoregresif Hareketli Ortalama Modeli)

ARIMA : Autoregressive Intergrated Moving Average Model (Otoregresif Bütünleşik Hareketli Ortalama Modeli)

EKK : En Küçük Katsayı

YSA : Yapay Sinir Ağları

VIF : Varyans Enflasyon Faktörü

I :Girdi

C :Hücre Durumu

f :Unutturma Geçiş Kapısı

O :Çıktı

H :Gizli Durum

SVR :Support Vector Regression (Destek Vektör Regresyonu)

VAR :Vector Autoregression (Vektör Otoregresyonu)

SVAR :Structural Vector Autoregression (Yapısal Vektör Otoregresyonu)

**SATIŞ FİRMALARININ GÜNLÜK SATIŞ VERİLERİNDEN GELECEK SATIŞLARININ TAHMİN EDİLMESİ**

# ÖZET

Bu çalışmanın amacı firmaların geçmişte yapmış oldukları satış verilerinden, gelecekte yapacakları satışları tahmin etmektir.

Proje kapsamında Ocak 2013 ile Ekim 2015 yılları arasında en büyük Rus yazılım firmalarından biri olan 1C Şirketi tarafından sağlanan günlük satış verilerinden oluşan bir zaman serisi veri seti ile çalıştık.

Bu veri kümesi üzerinde çeşitli makine öğrenmesi ve veri madenciliği algoritmalarını kullanarak belirlemiş olduğumuz amaca ulaşmayı hedefledik. Bu algoritmaları Python dilinde ve Spyder ortamında yazmaya kadar verdik. Bununla birlikte birçok daha teknolojiden de bu proje aracılığıyla faydalanma imkanı bulduk. Weka ve Libreoffice Calc bu teknolojilere örnek olarak verilebilir.

**Anahtar kelimeler: Makine Öğrenmesi, Sinir Ağları, Python, Satış Veri Analizi, Zaman Serileri**

**PREDICTING THE FUTURE SALES OF SALES DATA FROM DAILY SALES COMPANY**

# ABSTRACT

The purpose of this study is to predict the future sales of the past sales data of the companies.

We worked with a time series data set between January 2013 and October 2015. The dataset provided by one of largest Russian compony which is named 1C.

We aimed to achieve this goal by using various machine learning and data mining algorithms. We have decided to write the algorithm with Python in Spyder IDE. However, we had the opportunity to benefit from more different technologies through this project. Weka and Libreoffice Calc are examples of these technologies.

**Keywords: Machine Learning, Neuron Network, Python, Sales Data Analysis, Time Series**

# GİRİŞ

Yapmış oldukları satışlar bir firmanın ayakta kalabilmesi için en önemli etkendir. Bundan dolayı firmaların satışlarını çok iyi analiz edebilmesi gerekir. Bu insanlar için büyük bir yük artı zaman kaybı ve firmalar için büyük bir maliyet gerektirirken bilgisayarlar bu işlemleri yazılan algoritmalarla birlikte daha güvenilir ve kısa sürede gerçekleştirebilir. Bu işlem firmalara zaman ve maliyetten kazanç sağlar. Yukarıda bahsetmiş olduğumuz algoritmalar verinin işlenmesi ile ilgili algoritmalardır. Bunlara Veri madenciliği ve makine öğrenmesi algoritmaları diyebiliriz. Verinin her alanda hayatımızda olduğu şu dönemlerde gerçek ve işe yarar bilgiye ulaşmak gün geçtikçe zorlaşıyor. Buna paralel olarak ta veri madenciliği ve makine öğrenmesi hızla daha önemli bir konuma geliyor. Basit bir tanım yapmak gerekir ise veri madenciliği, büyük ölçekli veriler arasından bilgiye ulaşma işidir. Ya da büyük veri yığınları içerisinden gelecek ile ilgili tahminde bulunabilmemizi sağlayabilecek bağıntıların bilgisayarlar kullanarak aranmasıdır [1]. Makine öğrenmesi ise matematiksel ve istatistiksel yöntemler kullanarak eldeki veriden çıkarımlar yapan ve bu çıkarımları modelleyerek bilinmeyen verilere ulaşma işlemidir. Bakıldığında kolayca görülebildiği üzere makine öğrenmesini veri madenciliğinden ayırmak mantıksızdır. Makine öğrenmesi model oluştururken veri madenciliği çıktılarını kullanarak sonuç üretir. Veri okuma, önişleme, model oluşturma gibi aşamalarda hem günümüzde makine öğrenmesinde en çok kullanılan dil olmasından dolayı hem de kullanışlı ve diğer dillere göre veri madenciliği işleminde daha kolay olması açısından Python dilini, kodlama ortamı olarak ta Spyder IDE’ sini kullanmaya karar verdik. Makine öğrenmesi ile ilgili günümüzde literatür bakımdan sayılı Türkçe kaynak bulunmaktadır. Yapılan bu projenin hem asıl amacı olan geçmiş satış verilerinden gelecek satış verilerinin tahmini, hem de ileriki çalışmalar için bir kaynak niteliği taşıması hedeflenmiştir.

# GENEL TANIMLAR

## Veri Madenciliği

Veri madenciliği en sade hali ile “büyük ölçekli veriler arasından değerli olan bilgiyi elde etme işidir” [2].

Bilgi ve iletişim alanında yapılan gelişmeler dikkate alındığında, özellikle elektronik ortamda depolanan veri miktarında bir hayli artış olmuştur bununla beraber bu bilgiyi yorumlama konusunda insan kapasitesinin bir yerden sonra yetersiz kalmıştır. Dolayısıyla büyük miktar veri içerisinden bilgiyi yorumlama işi günümüzde makinelere yani bilgisayarlara kalmıştır. Bu noktada bilgisayarlara bu işleri yaptırabilmek için algoritmalar tasarlanmıştır. Bu algoritmalar yapı olarak çok farklı olabilseler de genel amaç hepsinde veriden bilgiye ulaşmaktır. Bilgi kısaca, verinin yorumlanışı, yani veriyi faydalı ve kullanılabilir hale getirme olarak tanımlanabilir.

Veriden bilgiye ulaşma aşamaları kısaca şöyledir;

* Veri Temizleme: Gürültülü ve tutarsız verilerin kaldırılması
* Veri Bütünleştirme: Çoklu veri kaynaklarının birbirine bağlanması
* Veri Seçimi: Analiz görevi ile ilgili verinin veri tabanından çıkarılması
* Veri Dönüştürme (Normalizasyon): Verinin uygun formlara dönüştürülmesi ya da birleştirilmesi
* Veri Madenciliği: Veri örüntüsü (pattern) çıkarabilmek için algoritmaların uygulanması
* Örüntü Değerlendirme: Bazı ilgi çekicilik ölçütlerine dayalı bilgiyi temsil eden gerçekten ilgi çekici desenleri tanımlayabilmek

Veri madenciliği aşamasında bahsedilen algoritmalar temel olarak, sınıflandırma, kümeleme ve birliktelik kuralları şeklinde gruplandırılmaktadır. Bahsi geçen algoritmalardan proje kapsamında kullanılacak olanlar “Literatürde Bahsedilen Yöntemler” başlığı altında anlatılacaktır (Bölüm 7).

Giudici (2013)’ ye göre veri madenciliğinde kullanılan metodolojinin bir kısmı makine öğrenmesi, diğer kısmı da istatistiksel olmak üzere iki araştırma alanından gelmektedir. İstatistik, yapay zeka ve veri tabanı sistemleri veri tabanı sistemleri veri madenciliği tekniklerinin ve terminolojisinin alındığı üç genel kök olarak görülmektedir [2].

## Makine Öğrenmesi

Makine öğrenmesi; elimizde olan mevcut veri ve veri kümelerinden, matematiksel ve istatistiksel yöntemler kullanarak mevcut verilerden çıkarılar yapan, bu çıkarımlara göre tahmin yapan yöntem biçimidir [3]. Günümüzde makine öğrenmesine ait örneklere sıkça karşılaşılmaktadır. Makine öğrenmesi günlük hayatımızda satış stratejisi oluşturma, el yazısı tanıma, robot hareketleri, bilgisayar oyunları, ve diğer birçok alanda kullanılmaktadır [4].

**Terminoloji**• Gözlemler: Makine eğitmek veya test etmek için kullanılan veri parçası  
• Özellikler: Bir gözlemi temsil eden verilerdir.  
• Etiketler: Gözlemler için verilen kategorilerdir.  
• Eğitim Verisi: Algoritmanın daha rahat ve kolay öğrenmesi için kullanılan veri dizisi  
• Test Verisi: Algoritmanın eğitim verisi sonrası öğrendiği çıkarımların gerçeğe ne kadar yakın olduğunu test etmek için kullanılan veri seti

Makine öğrenmesi algoritmaları temelde gözetimsiz ve gözetimli olarak 2 kategoriye ayrılır.

### Gözetimli Öğrenme

Gözetimli öğrenme metotlarında makineye bir eğitim veri kümesi girilir. Bu eğitim veri kümesi, makinenin daha sonradan karşılaşabileceği benzer veri kümesinin nasıl bir çıktı üreteceğini yani problemi nasıl çözeceğini gösteren bir yol göstericidir. Örneğin bir araba galericisisiniz ve işe yeni bir eleman alıyorsunuz. Siz bir arabaya baktığında geçmiş tecrübelerinize dayanarak bir fiyat çıkarımı yapabilirken yeni elemanınız daha tecrübesiz olduğu için bu fiyatlandırmayı yapamamaktadır. Yeni elemanınıza işi öğretmek istiyorsunuz. Bundan dolayı farklı farklı marka ve modellerden oluşan bir fiyatlandırma kataloğunu yeni elemanınıza veriyorsunuz. Daha somut düşünebilmek için Tablo 2.1’i örnek olarak verebiliriz.

Tablo 2.1 Eğitim Kümesi Örneği

|  |  |  |  |
| --- | --- | --- | --- |
| Marka | Model | Yıl | Fiyat |
| BMW | 1.16d | 2018 | 140.000 |
| BMW | 1.16d | 2012 | 105.000 |
| BMW | 1.18i | 2015 | 35.000 |

Bu Tablo 2.1’i eğitim verisi olarak düşünebiliriz. Yeni elemanımız bu verilere çalıştıktan sonra aşağıdaki Tablo 2.2’deki örnek arabaya bir fiyat verebilecek düzeye gelir.

Tablo 2.2 Örnek Test Kümesi

|  |  |  |  |
| --- | --- | --- | --- |
| Marka | Model | Yıl | Fiyat |
| BMW | 1.16d | 2014 | ? |

İşte bu yöntem gözetimli öğrenme yöntemidir. Çalışanı bir makine olarak düşünürsek, gelen eğitim verisini birtakım kendi yorumları ile birleştirerek (makine gözetimli öğrenme algoritmaları) gelen hiç bilinmeyen yeni veri için bir tahminde bulunabilir.

Gözetimli öğrenmede amacı her girdi vektörüne sonlu sayıdaki ayrık kategorilerden birini atama olan durumlar **sınıflandırma (classification)** problemi olarak tanımlanmaktadır. Çıktı uzayındaki her elemente **sınıf (class)**, sınıflandırma problemini çözen öğrenme algoritmasına da **sınıflandırıcı (classifier)** adı verilmektedir. Eğer istenen çıktı bir ya da daha fazla sürekli değerden oluşmakta ise bu yönteme **regresyon** denilmektedir. [2].

### Gözetimsiz Öğrenme

Gözetimsiz öğrenme modeli, sonuçları tamamen veri kümesine bağlıdır. Veri kümesinden sadece girdi verilerini alır ve girdi verileri üzerinden işlem yapar. Gözetimsiz öğrenme modelinde, modele çıktı verileri verilmez. Bu model genellikle veri sıkıştırma işlemlerinde kullanılır. Bu işlemlerde genellikle olasılık dağılımına bağlıdır [4]. Yine galerici olduğumuz örneğe geri dönelim. Buradaki örnekte marka ve modellerde daha fazla çeşitlilik olduğunu ayrıca hangi arabanın hangi meslek grubuna ait kişi tarafından alındığını varsayalım. Gözetimsiz öğrenme algoritmalarında, benzer marka, model ve fiyat özelliklerine göre arabaların gruplandırılması sonucunda bir takım farklı kümeler elde etiğimizi göreceğiz. Bu kümelerdeki kişilerin ayrı ayrı mesleklerine bakıldığında buradan bir örüntü çıkarabiliriz. Örnek olarak BMW marka aracın en çok doktorlar tarafından alındığını görüyoruz. Bir galerici olarak satış politikamızı bu sonuçlara göre belirleyebiliriz. Bu örnekteki gruplandırma işlemini gözetimsiz öğrenme algoritmaları ile sağlarız.

Gözetimsiz öğrenme algoritmalarında yöntemler kümeleme üzerineyken gözetimli öğrenme algoritmalarında sınıflandırma üzerinedir.

## Zaman Serileri

Zaman serisi, büyüklüklerin zaman içinde sıralanmış ölçütlerinin kümesidir. Her seri bir zaman serisi değildir. Bir serinin zaman serisi olabilmesi için zamana bağlı, zamana göre değişebilen bir durum olmalıdır. Dolar endeksini örnek olarak gösterirsek dolar para biriminin önceki günkü kapanış değeri, ertesi günkü açılış ve ertesi günkü kapanış değerini etkileyebilmektedir. Dolayısıyla bir serinin zaman serisi olabilmesi için en az bir değişkeninin zamana bağlı bir değişken olması gerekmektedir. [5]

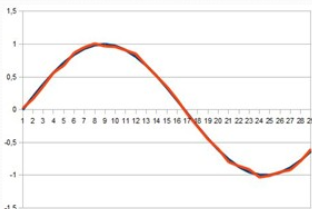
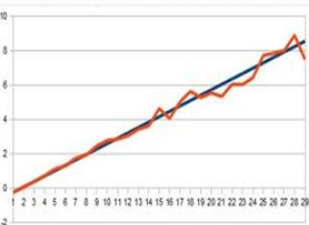
Zaman serilerinin birkaç kullanım amacı vardır.

* Aykırı Veri yakalama: Veriler zamana göre sıralandığında, sıranın veya serinin en dışında kalan veriye denmektedir. Aykırı verilerin yakalanmasını sağlamak için verilerin zamana göre hareketlerini tespit etmek ve bu hareketlerin dışında kalan verileri tespit etmek gerekmektedir.
* Tahmin: Zaman serisi analizi yaparak, olayların sonuçlarını tahmin edilmesidir. Kışın salep içeceğinin daha fazla satılıp, yazın satış sayısının daha az olmasının bilinmesiyle satış miktarının yaz-kış ayları arasında tahmini yapılabilir.
* Eksik verileri tamamlama: Eksik olan verilerin yerine gelebilecek yeni verilerin ne olacağı zaman serisi analizi yapılarak tahmin edilebilir. İşlem sonucu çıkan tahmini değer, veri kümesinde kayıp olan verinin yerine konur ve yapılacak işlemler tahmini veriler ile devam eder.
* Hata düzeltme: Aykırı değerlerin, diğer normal değerlere yakınlaştırılma çalışmasıdır.

### Zaman Serisi Bileşenleri

Zaman serileri geleceğin tahmini için kullanılan bir yöntemdir. Zaman serileri; trend, mevsimsel dalgalanma, düzensiz hareketler, döngüsel dalgalanmalardan oluşur.[6]

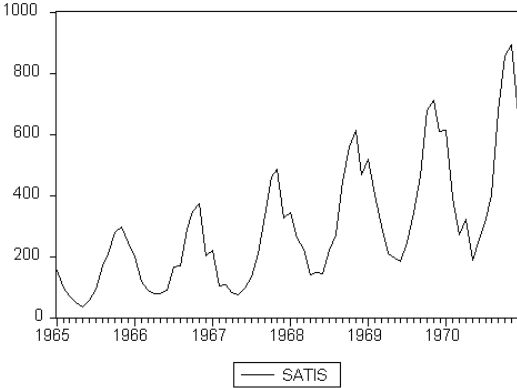
**Trend (Genel Eğilim):** Trend, zamana bağlı olarak gözlemlenen bir değişkenin uzun donem içinde gösterdiği artış veya azalışa denmektedir. Trendlerin, doğrusal trend ve doğrusal olmayan trend olmak üzere iki çeşidi vardır.



Şekil 2.1. Doğrusal trend örneği [7]

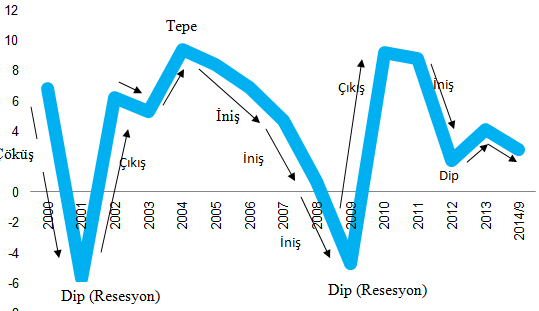
Şekil 2.2. D. olmayan trend örneği [7]

**Mevsimsel Dalgalanma:** Mevsimsel dalgalanma, bir yıl içinde oluşan etkilerin ele alınıp diğer yıllarla karşılaştırılıp, değişmelerin ifade edilmesidir. Mevsimlere bağlı olarak bazı değişkenlerin mevsimden mevsime etkilerinin arttığı görülmektedir. Örneğin dondurma ürününün yaz aylarında kış aylarına göre daha çok tüketildiğini görürüz.



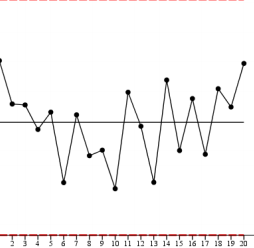
Şekil 2.3. Mevsimsel dalgalanma grafiği [7]

**Döngüsel dalgalanma (konjüktör):** Döngüsel dalgalanma, uzun bir dönemde seride oluşan değişmeleri veya dalgalanmaları göstermektedir. Periyodik olmayan ve düzensiz yapılardır. 5-10 yıl arasında dalgalanmalar ile tekrarlanır. Döngüsel dalgalanma hareketinin ortalama uzunluğu mevsimsel dalgalanmaların ortalama uzunluğundan daha uzundur.

****

Şekil 2.4. Döngüsel Dalgalanma Örneği [8]

**Düzensiz (Rassal) Hareketler:** Düzensiz (rassal) hareketler, zaman içinde düzensiz değişmelerdir. Hiçbir değişken düzensiz hareket ve düşmelerden sorumlu değildir. Düzensiz hareketlerin belli harmonik hareketleri yoktur, bir anda yükselip bir anda aşağı yönelebilirler.



Şekil 2.5. Düzensiz (rassal) hareket örneği [9]

Zaman serisi trend, mevsimsel dalgalanma, düzensiz hareketler, döngüsel dalgalanmalardan oluşur.

Zaman serisi = f(trend, mevsimsel dalgalanma, döngüsel dalgalanma, düzensiz hareketler)

Yani kısaca;

* Zaman serisi = İzlenen seyir + Hata terimi, olur.

Zaman serileri ile tahmin yöntemleri için (bknz. Bölüm 7).

# LİTERATÜR TARAMASI

Proje kapsamında yol gösterici olması bakımdan incelenen literatürler aşağıda açılanmıştır. Bu literatürler incelenirken veri analizi işlemlerine ve makine öğrenmesi için kullanılan yöntemlere dikkat edilmiştir.

[12] İncelenen araştırmada, otomotiv sektörünün satış rakamlarının nasıl daha iyi yönde ilerletilebileceği ve yükseltilebileceği hedeflenmiş. Araştırmada Ocak 2007- Haziran 2011 yılları arasında Türkiye otomobil piyasasının bir görünümünü ortaya koyarak otomobil fiyat fonksiyonu tahmin edilmiş ve talep öngörüsü yapılmış. Talep tahmini aşamasında kullanılacak yöntem olarak yapay sinir ağları yöntemi tercih edilmiş.

Kullanılan veri kümesinin bir örneği verilmiş. Buradaki veriler yöntemi çalıştırırken işe yarayacaklar.

Kullanılmış yöntem: Yapay sinir ağları yöntemi

Hata tespiti yapılırken MAPE, RSME, MSE değerlerine bakılmış. MAPE değerleri %10 altında olan tahmin modelleri “yüksek doğruluk” derecesine sahip, %10 ve %20 arasında olan modeller “doğru tahminler” olarak sınıflandırılmış.

Sonuç: Çalışmada Ocak 2007-Haziran 2011 değerleri alınmış ve yapay sinir ağları yöntemleri kullanılmış. Veriler ile yapılan işlemler sonucu MAPE değeri %16.82 olarak çıkmış ve tahmin “doğru tahmin” sınıfına girmiş, kontroller yapıldığında yapılan tahminlerin gerçek tahminlere yakın olduğu görülmüş.

[13] İncelenen araştırmada, yumurta tavukçuluğunda elde edilen satış gelirini; yaş, yaşama gücü, yumurta ağırlığı ve yumurta verimi verileri kullanılarak tahmin eden bir modelin geliştirmesi amaçlanmış. Araştırmada kullanılan veriler Kayseri Tavukçuluk Sanati ve Ticaret A.Ş’ den temin edilmiş.

Kullanılmış yöntem: Ridge regresyon yöntemi ve EKK metodu

Araştırmada Ridge-Regresyon ve EKK metodu ayrı ayrı denenmiş ve EKK yöntemi kullanılarak elde edilen analiz sonuçları, çoklu bağlantı durumunda yanlış sonuçlara ve yanlış modellerin oluşumuna neden olabileceği görülmüş. Yapılan çalışmalarda, çoklu bağlantı durumunda Ridge-Regresyon yönteminin, denklemin katsayılarını yanlış temin ederek tahminlerin varyanslarını azalttığı için EKK yöntemi yerine tercih edileceği belirtilmiş.

Sonuç: Çalışmada öncelikle EKK yöntemi ile parametre tahminleri yapılmış. Bağımsız değişkenler arasında korelasyon katsayılarının 1’e yakın olması, VIF değerleri de beklenenin sağlanamaması sonucu Ridge Regresyon yöntemini kullanılmaya gidilmiş. Ayrıca Ridge-Regresyon yöntemi uygulanması sonucunda çıkan değerler gerçek verilerle kıyaslanmış ve gerçek verilere yakın olduğu tespit edilmiş.

[14] İncelenen araştırmada, Türkiye’nin muz üretim ve ithalatının daha önceki yılların verisini alarak ithalatının ve üretim sayısının tahminini kapsamakta. 1981-2014 yıllarına ait muz üretim ve muz ithalat miktarı incelenmiş. Bu incelemede Delphi ve Box-Jenkins metotlarından biri olan Çift Üstel Düzetme yöntemi kullanılarak 2015-2019 yılları için öngörüde bulunulmaya çalışılmış.

Kullanılmış Yöntemler: Delphi ve box-jenkings yöntemleri

İthalat ve üretimin tahmininde hatayı en aza indirmek için Box-Jenkins metodu olan Çift Üstel Düzeltme Yöntemi kullanılmış. Bu yöntem veri kümesindeki verileri matematiksel olarak düzeltmek için kullanılmaktaymış. Delphi yöntemi ise karmaşıklıkları uzaklaştırıp sonuçların daha sağlıklı görülmesini sağlamış.

Sonuç: Yapılan çalışmalar sonucu kullanılan yöntemlerin sunduğu tahmin sonuçları yıllar geçtikçe başarıya ulaşmış ve kullanılan yöntemlerden elde edilen tahminle doğrulanmış.

[15] İncelenen araştırmada, elektrik enerjisi tüketimi tahmininde bulunulmuş. Veriler 1991-1998 arası Türkiye’de kullanılan elektrik miktarı verileriymiş. Tahmin yapılırken Yapay sinir ağları yöntemi kullanılmış. Yapay sinir ağları tekniği ile bulunan sonuçlar Box-Jenkins ve regresyon teknikleri ile karşılaştırılmış ve hangisinin daha iyi olduğu tespit edilmeye çalışılmış.

Kullanılan yöntem: Yapay sinir ağları, box-jenkins ve regresyon yöntemi

Analiz yapılırken AR(q), ARMA(p, q) gibi Box-Jenkins yöntemlerine başvurulmuştur.

Karşılaştırma: Zaman serisi analizi, yapay sinir ağları ve regresyon tekniği teknikleri arasında her biri ile işlemler yapılmış, test kümesi için en küçük OMYH değeri yapay sinir ağları tekniğine aitmiş. Bundan dolayı yapay sinir ağları tekniği kullanılmaya karar verilmiş.

Sonuç: Yapılan araştırmada, yapay sinir ağları tekniğinin oldukça iyi ve gerçeğe yakın sonuçlar verdiği görülmüş. Regresyon ve Box-Jenkins modellerinin iyi sonuç vermesi, veri sayısı ile alakalıymış.

[16] İncelenen araştırmada, Türkiye’nin kısa, orta, uzun dönem yük tahminleri yapılmak istenmekteymiş. Yapılan tahmin çalışmalarında 1980-1998 yılları arası nüfus, kişi başına düşen gayri safi milli hasılat, büyüme hızı gibi veriler kullanılmış. Yapılan araştırmanın amacı daha önceden belirtilen verileri kullanarak 2001, 2003 ,2005 yıllarında gerçekleşebilecek yük ve enerji tüketim değerlerini tahmin edilmiş.

Kullanılan yöntemler: Regresyon analizi, yapay sinir ağları yöntemi

Yapılan çalışmada regresyon analizi ve yapay sinir ağları yöntemi kullanılarak İstanbul ve Türkiye için ayrı ayrı hesaplamalar yapılmış ve çıkan sonuçlar birleştirilmiş.

Sonuç: Yapay sinir ağları (YSA) uygulamasında ortalama mutlak hatanın, regresyon analizine göre daha düşük olduğu gözlenmiş ve YSA uygulamasının regresyon analizi yöntemine göre gerçek değerlere daha yakın sonuçlar verdiği gözlenmiş.

[17] İncelenen araştırmada, kapanış hisse senedi değerleri kullanılarak iki etkin modelin tahminde gösterdikleri performansları karşılaştırmaya çalışmış. Uygulamada veri kümesi olarak 2012-2016 yılları arasına ait veriler ile Çin, Türkiye, İngiltere ve daha birçok şehir ele alınmış, haftalık kapanış hisse senedi değerleri kullanılarak yöntemler arası karşılaştırma yapılmış.

Kullanılan yöntemler: Bulanık mantık ve yapay sinir ağları yöntemi

Yapılan çalışmada bulanık mantık ve yapay sinir ağları yöntemleri ayrı ayrı kullanılarak işlemler yapılmış ve bu işlem sonuçları karşılaştırılmış.

Sonuç: Bulanık mantık ve yapay sinir ağları yöntemleri kullanılarak yapılan çalışmaların birbirlerine yakın değerler çıktığı görülmüş. Bulanık mantık yöntemi yapay sinir ağları yöntemine göre az da olsa da iyi sonuç verdiği görülmüş. Bu sayede tahmin yöntemlerinde yapay zeka yöntemlerinin kullanılabilirliği de ispatlanmış.

[18] İncelenen araştırmada, Türkiye’nin otomobil piyasasının Türkiye ve diğer orta seviye gelir düzeyi grubu ülkelerinin milli gelir düzeylerini hesaba katarak, 1996-1999 yılları arasındaki verileri kullanılmış ve 2005 yılı için talep tahmini yapılmış.

Kullanılmış olan yöntem: En küçük katsayı yöntemi

Yapılan çalışmada en küçük katsayılar yöntemi kullanılmış ve elde edilen sonuçlar gerçek değerleri ile karşılaştırılıp doğru verilere yakınlığı tespit edilmeye çalışılmış.

Sonuç: En küçük katsayı yöntemi kullanılmış ve elde edilen verilerin gerçek verilerle kıyaslaması yapılmış. Yöntem kullanılarak elde edilen veri tahminlerinin veri kümesindeki verilere yakın olduğu saptanmış ve hata payının az olduğu tespit edilmiş.

[19] İncelenen araştırmada, şirketlerin yemek alımlarında ortalamam yenen yemeğe değil de kişisel tahminlerinden dolayı yemekler israf olmaktaymış. Bunun önüne geçmek ve günlük tüketim miktar tahmini yapmak için talep tahmin yöntemleri kullanılmış. Veri kümesi olarak 01.06.2016-31.05.2017 yılları arasındaki Hacettepe üniversitesi yemekhanesinin günlük yemek tüketim verileri ve menüleri kullanılmış. Açık kaynak kodlu WEKA yazılım aracından da faydalanılmış.

Kullanılan Yöntemler: Yapay sinir ağları yöntemi, destek vektör makinesi, regresyon analizi yöntemleri kullanılmış.

Sonuç: Yemekhane sisteminden elde edilen veriler kullanılarak yapay sinir ağları, destek vektör makinesi ve regresyon analizi ile verilerin analizi yapılmış ve günlük tüketim tahmini yapılmış.

Tahminlerin gerçeklere yakınlığının karşılaştırması yapılmış ve bu tahminler diğer üniversitelere de sunulmuş.

[20] İncelenen araştırmada, İstanbul borsa endeksinin 2005-2012 tarihleri arasındaki Almanya, Fransa gibi çeşitli borsa endeks değerlerine sahip veri kümesini kullanarak geleceğe yönelik borsa endeks tahmini yapılmış.

Kullanılan yöntemleri: Yapay sinir ağları, destek vektör makinaları

Yapay sinir ağları yöntemi ve destek vektör makinaları yöntemleri tek tek denenmiş. Yapay sinir ağları yöntemi 1.yöntem, destek vektör makinaları yöntemi 2.yöntem olarak seçilmiş.

Sonuç: Yapay sinir ağları yöntemi günümüzde pek çok alanda uygulanıp başarılı sonuçlar veren bir yöntemmiş. Kolay ve tutarlı bir yöntemdir. Her iki yöntem sonuçlarının da gerçeğe oranla hata payları oldukça düşük ve her 2 yönteminde tahmin konusunda güven verdiği ve bu tür ölçümlerde kullanılabileceği saptanmış.

[21] İncelenen araştırmada, Türkiye’deki konaklama işletmelerine yurtdışından gelen turistlerin oluşturduğu dış turizmin işletmelerde yarattığı yatak doluluk oranının tahmini gerçekleştirilmiş. Turizm Bakanlığından alınan “1990-2002” tarihleri arasındaki aylık olarak alınan veriler yapılan tahmin işleminde veri kümesi olarak kullanılmış.

Kullanılan yöntemler: Yapay sinir ağları yöntemi

Sonuç: Yapay sinir ağları yöntemi ile tahmin çalışmaları gerçekleştirilmiş. Tahmin sonuçlarının gerçek sonuçlarla karşılaştırılması yapılmış, sonuçların gerçek verilere çok yakın olduğu ve hata oranının oldukça az olduğu tespit edilmiş. Bu sonuç konaklama şirketleri için gelecek planlamasında önemli rol oynamış.

[22] İncelenen araştırmada, bankaların kredi verecekleri müşterilerinin kredi borçlarını ne sıklıkla ödeyeceklerini veya düzenli olarak ödeyip ödemeyeceklerini öngörmek için bazı tahmin yöntemleri ve işlemleri kullanılmış. Veri seti olarak bir bankadan daha önceden bireysel kredi kullanmış 1639 kişi seçilmiş ve geri ödeme planlarına uyup uymadıklarına veya ne kadar uydukları verileri kaydedilmiş ve bu veriler kullanılmış.

Kullanılan yöntemler: Yapay sinir ağları yöntemi, lojistik regresyon yöntemi

Yapay sinir ağları yöntemi ve lojistik regresyon yöntemleri ile veri kümesi ayrı ayrı denenmiş ve birbirleriyle karşılaştırılmış.

Sonuç: Bankaların kredi vermiş oldukları kişilerin, taksitlerini düzgün ödeyip ödemeyeceği veya ne sıklıkla ödeyeceği ile ilgili tahmin yapılmış. Tahminde kullanılan yapay sinir ağları yönteminin doğruluk oranı lojistik regresyon yönteminin doğruluk oranından daha yüksek çıkmış. Bu sonuç yapay sinir ağları yöntemini banka kredi sistemi için lojistik regresyon yönteminden daha güvenli bir yöntem olduğunu göstermiş.

[23] İncelenen araştırmada, İMKB hisse senetlerinin getirilerini farklı değişkenler kullanarak tahmin edilmeye çalışılmış. Bu tahmin işlemleri 30 adet şirketin 2004 yılına ait Fiyat/Kazanç vb. değişkenlerini içerir. Yapılan araştırmanın hedefi, şirketlere hisse senedi getirilerinin ileriki süreç için tahminmiş.

Kullanılan yöntemler: Lojistik regresyon yöntemi

Lojistik regresyon yöntemi kullanarak veri kümesi içindeki 30 şirketten aldığı veriler ile tahmin işlemleri yapılmış ve gerçek veriler ile karşılaştırılmış.

Sonuç: Finansal oranlar kullanılarak lojistik regresyon analizi yapılmış, lojistik regresyon analiziyle getirileriyle finansal oranlar belirlenmiş, düşük ve yüksek gelirli şirketler tespit edilmiş. Elde edilen sonuçlarla gerçek veriler karşılaştırılmış.

[24] İncelenen araştırmada, Türkiye’nin önemli bir gelir kaynağı olan turizm sektörünün Ocak 2000-Aralık 2007 yılları arasında konaklayan yerli ve yabancı turist sayılarını ele alınarak yapılacak işlemler ile 2008 yılının ilk 6 ayı için doluluk oran tahmini yapılması hedeflenmiş.

Kullanılan yöntemler; Hareketli ortalama yöntemi, basit üstel düzleştirme yöntemi, Holt yöntemi, Winter yöntemi

Kullanılan programlar: Minitab 15, excel

Sonuç; Kullanılan bütün yöntemler ile işlemler yapılarak tahminler yapılmış ve karşılaştırılmış. Yöntemler arasında en az hata payı olan ve gerçeklere en yakın değer veren yöntem Winter yöntemi olarak gözlenmiş.

[25] İncelenen araştırmada, İMKB Ulusal 100 endeksinin ertesi günkü değerinin tahmin edilmesi için yapay sinir ağları kullanılmış. Bu tahmin için en çok kullanılan hatayı geriye yayma yöntemi kullanılmış. Veri kümesi olarak İMKB Ulusal 100 değerinin işlem yapılan günkü değeri alınmış.

Kullanılan Yöntemler; Yapay sinir ağları

Yapay sinir ağları yöntemi kullanılmıştır, analizler yapılırken hatayı geriye yayma yöntemi ve bu yöntemin momentum ile desteklenmiş şekli ile yapılmış.

Sonuç; Yapay sinir ağları ile yapılan tahminler sonucu elde dilen tahmin ile işlem yapılan veriler arasında yüksek bir fark ortaya çıkmış. Bu fark, yapay sinir ağları yönteminin ülkemizin piyasa kullanımı için tam olarak doğru sonuçlar vermediği saptanmış.

[26] İncelenen araştırmada, tıp sektöründe doğum sonrasında bebeklerin ağırlığı ile ilgili tahminler yapılmaya çalışılmış. Veriler Sivas ili Doğumevi hastanesinde 2004 yılına ait anket verileri kullanılmış. Anketler her 3 ayda 1 olacak şekilde gerçekleştirilmiş.

Kullanılan Yöntem: Lojistik regresyon yöntemi

Lojistik regresyon yöntemi kullanılarak anket verileri alınmış ve işlemler gerçekleştirilmiş.

Sonuç: Lojistik regresyon yöntemi ile yapılan işlem sonuçları ile gerçek veriler karşılaştırılmış. Karşılaştırma sonucu hata oranı oldukça düşük çıkmış. Tahmin sonucu ortaya çıkan veriler, gerçek verilere yakın ve benzer elde edilmiş.

[27] İncelenen araştırmada İzmir ili içerisinde gerçekleşen ileride oluşabilecek trafik kazalarının tahmin edilmesi hedeflenmiş. Bu tahminler için 1986-2005 tarihleri arasında yapılan trafik kaza sayısını ve ay bazında yapılan trafik kaza sayıları kullanılmış.

Kullanılan yöntemler: Regresyon analizi yöntemi, yapay sinir ağları yöntemi, genetik algoritma yöntemi

Sonuç: Regresyon analizi, yapay sinir ağları yöntemi, genetik algoritma yöntemleriyle ayrı ayrı işlemler yapılmış ve gelecekte oluşabilecek trafik kaza sayı tahminleri gerçekleştirilmiş. Tahminler yapılırken nüfus, araç ve kaza sayısı değişkenleri kullanılmış. Yapay sinir ağları metodunun diğer metotlara göre daha iyi performans ve daha doğru sonuç verdiği gözlemlenmiş.

[28] İncelenen araştırmada; C16 sınıfından toplam 50 adet küp numuneye belirli işlemler yapılmış ve bu işlemlerin sonunda numunelerin basınç dayanıklılığı tahmin edilmeye çalışılmış.

Kullanılan yöntemler: çoklu lineer regresyon yöntemi, bulanık mantık yöntemi

Lineer regresyon ve bulanık mantık yöntemleri kullanılarak numuneler üzerindeki çeşitli testler sonucu alınan veriler ile işlemler yapılmış.

Sonuç: Testler sonucu elde edilen veriler karşılaştırıldığında, yöntemler ile betonlarda basınç dayanımlarının tahminin mümkün olduğu bulanık mantık modelinin çoklu lineer regresyona göre daha düşük bir hata oranıyla tahmin ettiği görülmüş.

[29] İncelenen araştırmada; yer ivmesi hareketlerinin daha önceden gerçekleşmiş olan 47 adet deprem verisi ile çalışılmış, tahmin yöntemleri kullanarak testler gerçekleştirilmiş.

Kullanılan yöntemler: Yapay Sinir ağları yöntemi, regresyon yöntemi

Sonuçlar: Yapay sinir ağları ile en büyük yer ivmesi tahmini yapılmış ve elde edilen sonuçlar regresyon yöntemi ile karşılaştırılmış. Regresyon ile karşılaştırıldığında, ivmeler yapay sinir ağı tarafından daha yüksek korelasyon katsayısı ile tahmin edilmiş. Yapay sinir ağları gerçeklere yakın değerler ile sonuçlanmış.

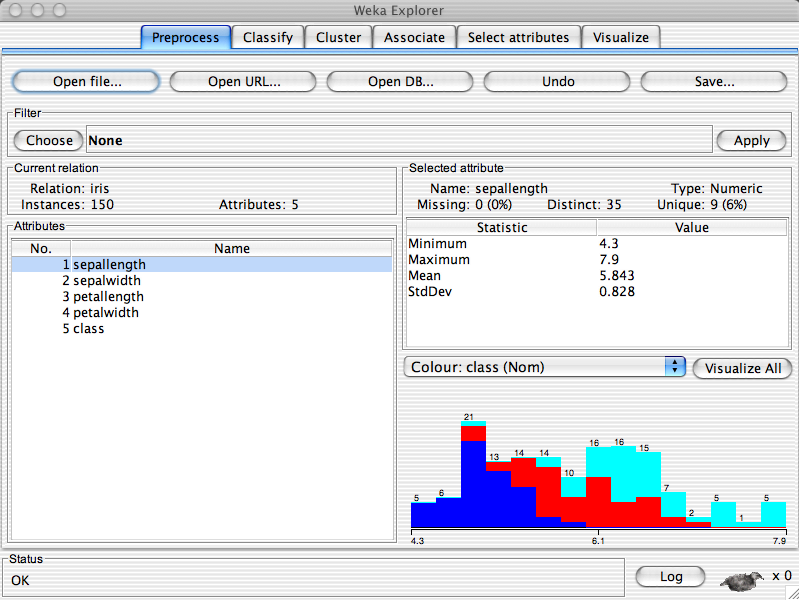
# KULLANILAN TEKNOLOJİLER VE AÇIKLAMALARI

## **WEKA**

Makine öğrenmesi konusunda kullanılan paketlerden birisidir. Waikato üniversitesinde Java dili üzerinde açık kaynak kodlu olarak geliştirilmiştir ve GPL lisansı ile dağıtılmaktadır. Waikato Environment for Knowledge Analysis kelimelerin baş harflerinden oluşur. [35]

Weka verileri dosyadan okur ve değerleri sayısal, nominal değerler olarak kendi içinde değiştirebilir. Ayrıca verileri veri tabanından da çekebilir.

Weka’da makine öğrenmesi ve istatistik yapılırken hazır kütüphaneler kullanılabilir çünkü Weka, bu konularla ilgili kütüphaneleri kendi içinde hazır olarak barındırır. Örneğin veri ön işleme (data preprocessing), kümeleme (clustering), sınıflandırma, karar ağaçları gibi modelleri hazır olarak veri kümemize sadece gerekli birkaç parametreyi değiştirerek hızlıca uygulayabilir ve modelin başarı oranını da kolayca öğrenebiliriz. (Örnek için bknz. Bölüm 6.2)



Şekil 4.1. Weka’da veri çekme ve veri özelliklerini gösterme örneği [36]

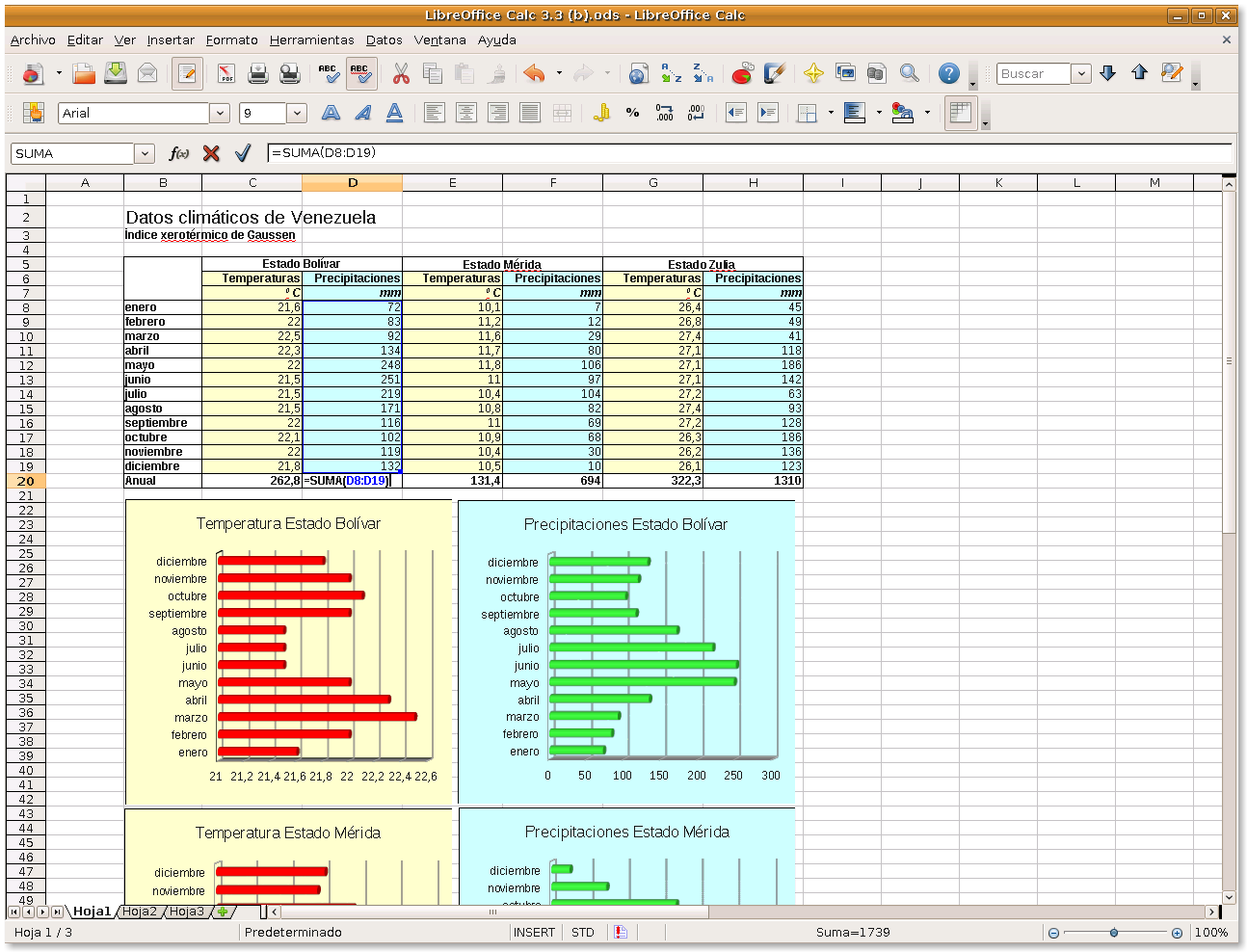
Weka “.arff” ve “.csv” gibi formatta dosyaları rahatlıkla okuyabilir. Şekil 4.1 de görüldüğü üzere sınıflandırma algoritmasında kullanılmak üzere veri dosyadan okunmuştur. Bu yaprak türleriyle alakalı veri kümesi klasiktir ve Weka içerisinde hazır olarak bulunmaktadır.

## LibreOffice Calc

LibreOffice özgür ve açık kaynakları uygulamalarından biridir.

LibreOffice Calc uygulaması kendi içinde hesap tablosu, sunu, çizim ve akış şeması uygulamasını, veri tabanı ve matematiksel uygulama yazılımını içinde barındırır. [37]

LibreOffice Calc ile birçok grafik çizdirebilir, numerik verilerde standart sapma, aritmetik ortalama, varyans hesabı gibi birçok matematiksel işlemi kolaylıkla gerçekleştirebiliriz. [38].



Şekil 4.2. LibreOffice Calc’da elde olan veriden grafik çizdirme örneği [39]

Şekil 4.2’de de görüldüğü üzere bir takım sayısal veriler okunmuş ve daha anlaşılabilir olması için grafiklerle açıklanmıştır.

## Python

İlk sürümü Guido van Rossum tarafından 1991 yılında yazılmış bir programlama dilidir. Python’un diğer programlama dilleri ile kıyaslama yapıldığında bazı avantajları vardır;

* Daha kolay öğrenilebilir, diğer programlama dillerine göre daha basittir.
* Diğer programlama dillerine nazaran derleyici gerektirmemektedir.
* Hem daha okunaklı, hem de daha temiz kodsal söz dizinimine sahiptir. [40]

Günümüzde Python, yazım kolaylığı ve sahip olduğu geniş hazır kütüphaneler sayesinde oldukça popülerleşmiş ve yaygınlaşmıştır. Python hemen hemen her türlü platform da çalışabilir. Bu özelliği de Python’u yaygın bir dil haline gelmesinde önemli bir özelliktir. [41]

Çalışma kapsamında kullandığımız bazı kütüphaneler;

**NumPy**: Python’ da tanımlanmış, hesaplamalar ve işlemler için kullanılan temel pakettir. Bu paket, matematiksel ve mantıksal işlemler, seçme, sıralama gibi bünyesinde pek çok işlem bulunduran bir kütüphanedir. [42]

**Seaborn**: Python’ da verileri grafiklere dökmeyi veya ara yüz yapmamızı sağlayan bir kütüphanedir. [43]

**Pandas**: Veri analizi ve veri ön işlemeyi kolaylaştıran açık kaynak kodlu bir kütüphanedir. [44]

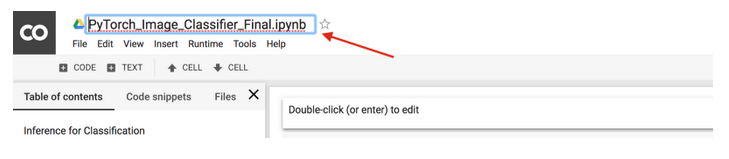
## Spyder

Spyder, Python ile yazılmış ve Python geliştirme için kullanabileceğiniz, açık kaynak bir IDE’dir. Genel olarak akademik çalışmalar ve veri bilimi için hazırlanmış olsa da herkesin kolayca Python kodlayabileceği bir ortamdır.  
Spyder’ın Özellikleri:

* Çapraz Platform – Linux, Windows ve Mac OS
* Ücretsiz ve Açık Kaynak
* Kod renklendirmeyi destekliyor
* Çoklu dil desteğine sahip
* Dökümantasyon görüntüleme, grafik ve veri görselleştirme vb.
* Eklentiler ve API’ler sayesinde fonksiyon uzantılarını destekler [45]

## Google Colab

Google Colab, Google tarafından üretilmiş GPU’yu destekleyen ücretsiz bir bulut hizmetidir. Google Colab’ı diğer bulut uygulamalarından ayıran en büyük özellik tamamen ücretsiz olmasıdır. Bu bulut hizmetini almak isteyen müşteriler hiçbir ücret ödemeden bu platformu kullanabilmektedirler. Google Colab üzerinden Python ile programlama yapabilir, Keras, Tensorflow gibi bilindik ve çok kullanılan kütüphaneler ile uygulamalar geliştirilebilir. [70]



Şekil 4.3. Örnek bir Python uygulamasının Google Colab üzerinden çalıştırılması

**GPU (Graphics Processing Unit):** Ekranı olan neredeyse bütün cihazlarda bulunan GPU, grafikleri işleme ve ekrana getirme ünitesidir. GPU’lar bilgisayarlarımızda harici veya dahili olmak üzere mevcut olan ekran kartlarımızın merkezinde yer alırlar. Grafikleri işlemektedirler ve işleme işlemlerinin ardından ekrana verme işlemini yapmaktadırlar. [[72](https://www.teknotechnic.com/gpu-nedir-tanimi-ve-gpu-hakkinda-temel-bilgiler/)]

# VERİ ANALİZİ

Verilerin makine öğrenmesi algoritmalarında kullanılmadan önce birtakım işlemlerden geçmesi gerekir. Bunun sebebi verilerin gerçek hayatta her zaman doğru ve kusursuz olarak bulunmuyor oluşudur. Gerçek hayatta veriler gürültülü, eksik ve tutarsız olabilir. Bu gibi veriler öğrenme modeli çok iyi bile olsa modelin vereceği potansiyel doğru sonuçları etkiler. Dolayısıyla modelin doğru sonuç vermesini sağlamak için veriyi öncelikle birtakım işlemlerden geçirerek veriyi kullanılabilir hale getirmeliyiz. Bu işlemlere literatürde “veri ön işleme” denilmektedir. [32]

Veri ön işleme süreci (Han ve Kamber (2006)) genel olarak beş temel adımda gerçekleştirilmektedir. Bunlar;

* Veri özetleme
* Veri temizleme
* Veri bütünleştirme ve dönüştürme (Normalizasyon)
* Veri indirgeme

Veri ayrıklaştırma ve kavram hiyerarşisi yaratma

Kendi verimiz üzerinde bu adımları gerçekleştirmeden önce bu adımların neler olduğunu ve bu adımlarda kullanılabilecek metotlara kısaca göz atalım.

Bilindiği gibi veri madenciliği kısaca tek başına anlam ifade etmeyen verilerin birleştirilmesinden, ayrıştırılmasından vb. İşlemler sonrası anlamlı bir sonuç çıkarabilir hale getirme işlemidir. (Ahsan ve Shah, 2006). Bizimde genel amacımız aslında budur. Bu adımların gerçekleştirilebilmesi için öncelikle temel birkaç kavramdan bahsedeceğiz.

## Veri Tipleri

Kullanılacak veri kümesi farklı veri tiplerinin birleştirilmesiyle oluşturulmuş olabilir. Veri kümesinin niteliklerini tam olarak anlayabilmek için kullanılan veri tipini anlamak önemli bir adımdır [33].

* Nominal: Kategorik tipteki değişkenlerdir. Veriyi gruplara ayırmada kullanılır. Örnek vermek gerekirse Kişileri eğitim seviyelerine göre gruplandırma “ilkokul”, “ortaokul”, “lise”, “lisans” vb.
* İkili (Binary): Nominal değişkenlerin sadece iki kategoriye ayrıştırılmasıyla oluşturulan verilerdir. Genel olarak bir durum ve onun karşıt durumunu temsil etmek için kullanılır. Örneğin; “Evet”,” Hayır”, “Erkek”, “Kadın”, “Çalışkan”, “Tembel” vb.
* Numerik: Sayısallık içeren verilerdir. Örneğin; -15, 25, 0.1658 vb.
* Sıralı (Ordinal): Eğer nominal veriler mantıksal bir sırayla yazılabiliyorsa sıralı nitelikler olarak adlandırılacaktır. Örneğin bir okuldaki öğrenci başarısının “çok düşük/düşük/orta/yüksek/çok yüksek” biçiminde değerlendirilebilmesi.
* Mantıksal: Sadece TRUE (Doğru) veya FALSE (yanlış) değeri alabilen verilerdir.
* DataFrame: Birçok tipten verinin birleşmesiyle oluşturulan bir dizi veri topluluğu. (Veri matrisi)

Veri tipleri gözden geçirildikten sonra, niteliklere ait ortalama, maksimum, minimum değerler, frekans bilgisi, mod, medyan gibi basit tanımlayıcı istatistiki bilgiler incelenmelidir. Bu ilişkiler; kutu diyagramı (box and whisker plot), serpilme grafiği (scatter plot), pasta ya da sütun grafiği gibi veriyi görselleştirme yolu ile tespit edilebilir [bknz. Bölüm 5.3 Veriyi analiz etmek için yöntemler]. Bu aşamalarda veriyi analiz ederek kayıp veriler (missing data impulation) olup olmadığı, normalizasyona (normalization) ihtiyaç olup olmadığı, ayrıklaştırma (discretization), gürültü (noise) ya da aykırı değerlerin (outliers) tespit edilmesi gibi ek bir işleme ihtiyaç duyulup duyulmadığı belirlenmeye çalışılır [33].

## **Veri Kümesinde Karşılaşılabilecek Hatalı Veriler**

Aykırı Değer (Outliers, Uç Noktalar)

Aykırı değerler, veri kümesi içerisindeki diğer gözlemlerden gözle görülür bir farkla ayrılmış verilerdir. Örneğin; Maaşın “-10” olması. Bu tip verileri gözlemlemek ve tespit etmek bazı koşullarda kolay bazı koşullarda ise zordur. Aykırı değer tespiti için birçok yöntem vardır. Veri kümesinin uç nokta bulundurduğundan şüphelendiğimiz özelliğini grafiksel olarak bazı programlar (bknz. Bölüm 4) yardımı ile çizdirebiliriz. Bunu dışında yine nümerik değerler için kutu grafiği çizimleri yapılarak ilgili niteliğe ait aykırılıklar gözlemlenebilir. [32]

Tekrar Eden Veriler

Veri kümesindeki tekrar eden satırlar bir yandan analiz sırasında vakit kabı yaşatacak diğer yandan veri kümesinin gereksiz biçimde şişmesine neden olacaktır. Ayrıca sonuçlara etkisi de vardır çünkü; veri madenciliği için oluşturduğumuz model tekrarlı verileri farklı bir yeni veri olarak algılayarak ve onu da işlemlere sokacaktır. Eğer veri kümemizde tekrarlı veri bulunuyorsa bunların veri kümesinden çıkarılmasına analiz sonuçlarına göre karar verilir.

Kayıp Değerler

Veri kümesi içerisinde bazı durumlarda ölçülemeyen veriler olabilir bu verilere kayıp veriler denilmektedir. Bu tip kayıp verilerin tamamlanması için en basit yöntem; eğer kayıp olan veri nümerikse o değerin ait olduğu özellik kümesinin ortalaması yazılır, eğer kayıp veri kategorik bir veri ise yine ait olduğu özellik kümesindeki değerler içerisinde en çok tekrar eden değer yazılır. Kayıp değerlerinde analize dahil edilmesi de analiz sonuçlarında birtakım sorunlar yaratacağı muhtemeldir. Kayıp değerleri tamamlamak adına kullanılabilecek birtakım yöntemler; [33]

* Eğer bir gözleme ait çok sayıda kayıp değer varsa, ilgili gözlem tümüyle veri setinden silinebilir.
* Kayıp değerler daha önceden belirlenmiş “N/A” gibi sabit bir sabit değerin atanmasıyla düzeltilebilir.
* Yukarıda da belirtildiği gibi eksik verinin bulunduğu özelliğin ortalamasına veya frekansına göre bir atama işlemi gerçekleştirilebilir.
* Regresyon yöntemi ile kayıp değerler tamamlana bilir. (bknz. Bölüm 7.1)
* Gözlemlenen en son değer kopyalanabilir.

## Veriyi Analiz Etmek için Yöntemler

Veri kümesi içerisinde karşılaşılabilecek hatalı verileri tanımlamıştık (bknz. 5.2. Veri kümesinde karşılaşılabilecek hatalı veriler). Bu kısımda bu hatalı verileri nasıl veri kümesi içerisinde tespit edebileceğimizi, veriyi nasıl yorumlayabileceğimize bakacağız. Bunun için öncelikle birkaç temel tanımı bilmemiz gerekiyor.

* Frekans: Verinin dizi içerisinde geçme sayısıdır.
* Mod: Dizi içerisinde en çok tekrar eden elemandır. Yani frekansı en büyük olandır.
* Medyan: Diziyi küçükten büyüğe veya büyükten küçüğe doğru sıraladığımızda dizi tek sayıda veriden oluşuyorsa ortadaki veridir. Eğer çift sayıda veriden oluşuyorsa ortadaki iki verinin aritmetik ortalamasıdır.
* Aritmetik Ortalama: Numerik verilerde dizinin elemanlarının toplanıp dizi eleman sayısına bölünmesiyle bulunur.
* Minimum Değer (min): Veri kümesinde analizini yaptığımız numerik verinin en küçük değeri
* Maximum Değer (max): Veri kümesinde analizini yaptığımız numerik verinin en büyük değeri

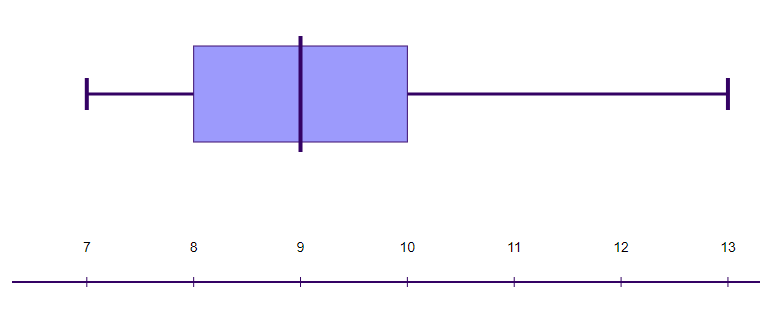
### Kutu Diyagramı Yöntemi (Box and Whisker Plot)

Veri analizinde kullanılan önemli metotlardan birisidir. Verinin dağılımını çizilen basit bir diyagram yarımı ile görebiliriz. Bu konuyu bir örnekle açıklayalım.

Örnek dizimiz şu şekilde olsun,

7, 8, 8, 8, 9, 9, 9, 9, 10, 10, 13

Box-Plot Çizimi;



Şekil 5.1. Örnek box-plot çizimi

Şekil 4.1’deki gibi olur. Bu grafik üzerinden verimizin özelliklerine bakacak olursak;

* Minimum değer = 7
* Maksimum değer = 13
* Q1 (İlk Çeyrek) = 8
* Q3 (Üçüncü Çeyrek) = 13
* Q2 (Medyan) = 9

Değerlerini rahatça görebiliriz. Bu şekilde verinin nasıl bir dağılım sergilediğine dair bir yorumda bulunabiliriz.

### Standart Sapma ve Varyans

Varyanslar ve standart sapma, veri dağılımı ölçümleri olup, bir veri dağıtımının yayılımını göstermektedir: Düşük bir standart sapma, verilerin izlenmesinin ortalamaya çok yakın olduğu anlamına gelirken, yüksek bir standart sapma verilerin geniş bir değer aralığında yayılım gösterdiğini gösterir [32].

: Varyans değeri

σ : Standart sapma

: Aritmetik ortalama

N : Dizideki eleman sayısı

(5.1)

Denklem 4.1 bize dizinin varyans değerini göstermektedir. Bu varyansın karekökünü aldığımızda elde ettiğimiz sayı ise bizim standart sapmamızdır.

Örnek vermek gerekirse şekil 4.1’deki örnek üzerinden standart sapma ve varyans değerlerimizi hesaplayalım;

Örnek veri kümesi: 7, 8, 8, 8, 9, 9, 9, 9, 10, 10, 13

= (7+8+8+8+9+9+9+9+10+10+13) / 11

9

Bütün değerlerimizden bu değerini çıkarıp karesini alıp topladığımızda;

= 4 + 1 + 1 + 1 + 1 + 1 + 16 = 25

25 / 11 2.3

2.3

σ 1.5

Şeklinde bulabiliriz. Buradan standart sapmanın yaklaşık 1.5, varyansın 2.3 olduğunu görüyoruz.

## **Hatalı Verilerin Düzeltilmesi için Temel Yöntemler**

Yukarıda da bahsettiğimiz gibi hatalı veriler küçük ve önemsiz gibi görünseler de sonuç üzerinde büyük hatalar almamıza sebep olabilir. Bunun önüne geçmek için birçok yöntem geliştirilmiştir biz temel olan birkaç tanesinden bahsedeceğiz.

* Kutulama (Binning)
* Kümeleme (Clustering)
* Regresyon

### **Kutulama Tekniği**

Bu yöntemi uygulamak için veriler öncelikle sıralanır (küçükten büyüğe ve büyükten küçüğe) daha sonra belirlediğimiz kutu boyutuna göre veriler eşit aralıklarla bölünür. Bu yöntem kendi içerisinde iki farklı yönteme ayrılır.

* Kutu ortalamasına göre
* Kutu Sınırına Göre

Sıralı veriler daha önceden belirlediğimiz kutu boyutuna göre parçalara ayrılır. Daha sonra kutunun üst ve alt sınırına göre aradaki değerler hangisine yakınsa ona göre değiştirilir.

Kutu Ortalamasına Göre

Sıralı veriler bir önceki yöntemdeki gibi kutu boyutuna göre parçalanır daha sonra kutu içerisindeki veriler kutu ortalamasına göre değiştirilir.

Kutulama tekniğinin gerçeklenmesine dair bir örnek aşağıda gösterilmiştir;

Sıralı veri:

4, 8, 15, 21, 21, 24, 25, 28, 34

Bölme genişliği (Kutu boyutu) = 3’ e göre

4,8,15

21,21,24

25,28,34

Kutu sınırına göre düzeltme;

4,4,15

21,21,24

25,25,34

Kutu ortalamasına göre düzeltme;

9,9,9

22,22,22

29,29,29

### **Kümeleme Tekniği**

Benzer veriler aynı kümede olacak şekilde gruplandırmak amaçlanmaktadır. Veride analiz edeceğimiz özellikler aynı küme içerisine alınır. Daha sonra küme dışarısında kalan veriler aykırı olarak tespit edildikten sonra analiz işleminden çıkarılır.

### **Regresyon**

Regresyon yöntemi eksik verilerin tamamlanmasında ve aykırı verilerin tespitinde kullanılan önemli bir yöntemdir. Bilinen verilerden bir regresyon doğrusu çizilerek bilinmeyenler tahmin edilmektedir. Ayrıca bu doğruya uzak olan veriler gürültülü veri olarak tespit etmek ve bunlar içinde bir tahminde bulunmak mümkündür. (bknz. Bölüm 7.1).

## **Normalizasyon**

Numerik veriler arasındaki büyük farklar mesafe ölçümüne dayalı veri madenciliği ve makine öğrenmesi algoritmalarında (örneğin; k-means, KNN gibi) sonuç üzerinde olumsuz sonuçlar oluşturabilmektedir. Bu gibi sorunlarla karşılaşmamak için veri normalize edilir ve aradaki fark düşürülerek algoritmaların daha iyi sonuç vermesi sağlanabilir. Bu normalizasyon işlemleri için birtakım yöntemler geliştirilmiştir [Han ve Kamber, 2006];

* Min-max normalizasyonu
* Z-score normalizasyonu
* Ondalık ölçekleme normalizasyonu

### **Min-Max Normalizasyonu**

v; A niteliğine ait normalize edilmek istenen değer,

v’; v’nin normalize edilen değeri

minA : A niteliğinin en küçük değeri

maxA : A niteliğinin en büyük değeri

yeni\_minA: Normalizasyon sonucu elde edilmek istenen en küçük değer

yeni\_maxA : Normalizasyon sonucunda elde edilmek istenen en büyük değer olsun

(5.2)

Denklem 5.2. min-max normalizasyonu denklemi [32].

### **Z-Score Normalizasyonu**

vı ; A niteliğine ait normalize edilmek istenen değer,

v’; v’ nin nomalize edilen değeri

Ā: A niteliğinin ortalaması

σA: A niteliğinin standart sapması olsun

(5.3)

Denklem 5.3 sonucunda hesaplanan v’ değeri bizim normalize olmuş yeni değerimizdir.

### **Ondalık Ölçekleme ile Normalizasyon**

Max(|v’|) < 1 koşulunu sağlayan en küçük tam sayı j olmak üzere;

vı : A niteliğine ait normalize edilmek istenen değer,

v’ : v’nin normalize edilen değeri

Denklem 5.4

Denklem 5.4’teki işlemler sonucunda v’ değeri bizim normalize olmuş yeni değerimizdir.

Bir veri seti için normalizasyon gerekli olup olmadığına, veri setinin normalize edilmiş ve normalize edilmemiş değerleri ile analizler tekrar gerçekleştirilerek karar verilir. Bu aşamaların başarıyla tamamlanması sonrası model oluşturma işlemine geçilebilir.

# VERİ KÜMESİ

Proje kapsamında kullanılacak olan veri kümesi en büyük Rus yazılım firmalarından biri olan 1C Şirketi tarafından sağlanan günlük satış verilerinden oluşmaktadır. Veri kümemiz Ocak 2013 tarihinden Ekim 2015 tarihine kadar olan farklı firmaların günlük satmış oldukları ürünlerin sayısını ve fiyatını içermektedir [31].

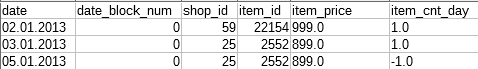
## Veri Kümesi İçeriği

Veri kümemiz “kaggle.com” internet sitesi üzerinden elde edilmiştir [31]. Web sayfası üzerinden elde edilmiş olan bu veri kümesi içerisinde eğitim kümesi ve test kümesi ayrılmıştır. Ayrıca küme içerisinde bulunan verinin açıklaması detaylı olarak aşağıda verilmiştir.

### Eğitim kümesi

Eğitim kümesi (bknz. 2.2.1 gözetimli öğrenme) içerisinde 6 adet sütun (özellik, attiribute) bulunmaktadır. Bunlar [31];

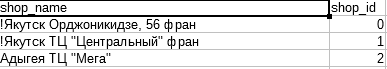
* date: gg / aa / yyyy biçiminde verilmiş olan tarihlerdir. (string)
* date\_block\_num = Kolaylık sağlamak için kullanılan ardışık ay numarası. (Örneğin: Ocak.2013 0, Şubat.2013 1, ..., Ekim 2015 33) (numerik)
* shop\_id: Verinin elde edildiği firmaları tanımlamak için kullanılan ID numarası.(numerik)
* item\_id: Satılan ürünler için kullanılan ID numarası (numerik)
* item\_price: ürünün mevcut fiyatı (numerik)
* item\_cnt\_day: Satılan ürün sayısı. Eğitim sonrası tahmin edilecek olan alan. (numerik)



Şekil 6.1. Eğitim verisinden örnek bir bölüm

Örneğin Şekil 6.1’de de görüldüğü gibi 02.11.2013 tarihinde *shop\_id’si* “59” olan firma *item\_id’si* “22154” olan üründen bir tane 999 birim fiyata satmış. Fakat görüldüğü gibi 05.01.2013 tarihinde 25 id’li firmanın satmış olduğu ürünün sayısı “-0.1” olarak görülüyor bu durum mümkün olmadığından satırda bir hata olduğunu fark edebiliyoruz. Bunun gibi verilere aykırı veriler denir. Aykırı verilerin düzeltilmesi için gerekli işlemlere ve daha detaylı bilgi için; [bknz. 5.4 Hatalı verilerin düzeltilmesi için temel yöntemler.]

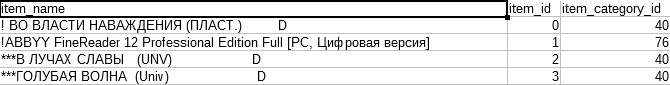
Veri kümemiz 59 tane farklı firmanın günlük satış verilerini içermektedir. Eğitim kümemizde görünen “shop\_id” kısmında bu firmaların ID’leri görünmektedir. Fakat bunların hangi firmaya ait olduklarını ayrı bir “shop.csv” dosyamızdan okumaktayız. Bu dosya içerisinde firmaların ID’leri ve isimleri eşleştirilmiştir. Örnek olması için veri kümesinden 3 satır şekil 6.2 aşağıda görünmektedir.



Şekil 6.2. Firma ID’lerinin bulunduğu veri kümesinden örnek bölüm

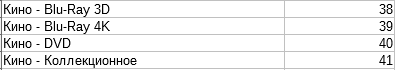
Firma isimlerinde yapmış olduğumuz araştırma ve gözlemlere göre ilk kısım firmanın bulunduğu şehri, ikinci kısım firmanın adını göstermektedir.

Aynı şekilde ürünler hakkında daha fazla bilgi almak için “item.csv” veri kümemizden örnek birkaç veri Şekil 6.3’te görülmektedir.



Şekil 6.3. Ürünlere ait ID ve kategori bilgileri

Şekil 6.3’te ürünlerin ID atamalarını ve kategori bilgilerini görüyoruz. Peki bu kategori ID’lerinin nerden geldiğini öğrenmek istiyorsak? O zaman kategori ID’lerinin bulunduğu veri kümesine bakmamız gerekiyor. Onunla ilgili görselde aşağıda Şekil 6.4 ismi ile paylaşılmıştır.



Şekil 6.4. Kategoriler ve bu kategorilerin ID bilgileri hakkında bir örnek

Örnek olarak bu iki şekilden (şekil 6.3 ve şekil 6.4) anladığımız kadarıyla, ürün ID’ si 0,2 ve 3 olan ürünler DVD kategorisinde bulunmaktaymış.

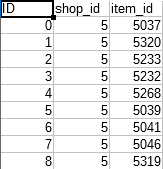
### Test Kümesi

Test kümemiz aynı firmaların Kasım 2015 tarihinde yapmış oldukları ürün satışlarını içerir. Bizim bu veri setinde tahmin etmemiz gereken kısım hangi üründen kaç tane satmış olduklarıdır. Buda yukarıda da bahsedildiği gibi eğitim kümemizde bulunan “item\_cnt\_day” özelliğidir. Bu sayı ile satılan ürünlerin fiyatlarının çarpımı (bu bilgide elimizde bulunmaktadır. Bknz. Bölüm 6.1.1. Eğitim kümesi) bize o firmanın Kasım 2015’te yapmış olduğu satışlarından toplam kazandığı para miktarını gösterir.

Test kümemizin içerisinde veriye ait 3 adet özellik vardır. Bunlar;

* ID (numerik)
* shop\_id (numerik)
* item\_id (numerik)

Bu veri bloklarının açıklamaları eğitim kümesi başlığı altında anlatılmıştır. Bölüm 6.1.1 de anlatılan eğitim kümesi içerisinde geçen özelliklerle aynıdır. Daha fazla uzatmamak adına bu kısımda bu özelliklerin açıklamaları anlatılmayacaktır.



Şekil 6.5. Test kümesinden bir örnek

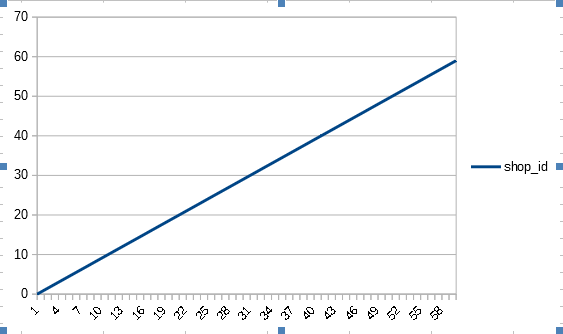
Şekil 6.5’te de görüldüğü üzere “ID” bloğu test veri kümemizdeki (market, ürün) ikilisini temsil eden benzersiz bir numarayı göstermektedir. Aldığı değerler [0-214119] arasındaki tam sayılardır. Bunun dışında “shop\_id” ve “item\_id” bilgisine bakıldığında hangi firmanın hangi ürünü sattığını görüyoruz. Şekil 6.5’te ki örnek veri kümemizde test verimizin [0-8] aralığındaki ilk 9 verisi görülmektedir. Sadece bu örnek dikkate alındığında yorumlama şu şekilde yapılabilir; “5” ID’sine sahip firmanın Kasım 2015 tarihinde satmış olduğu 9 ürün”.

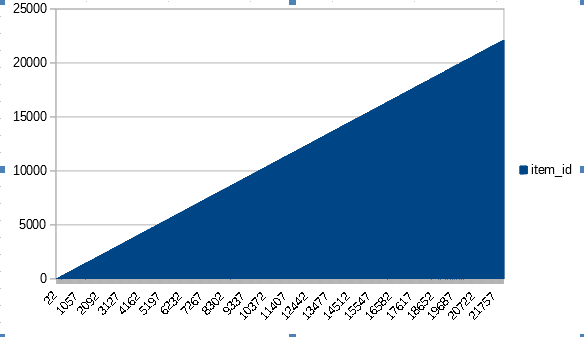
Yazacağımız model ile bu test veri kümesindeki firmaların ID’ si verilen üründen kaç tane satmış olduğunu tahmin etmeye çalışacağız.

## Weka ve LibreOffice Calc ile Veri Kümesi Analizi

Öncelikle firma isimleri ve firma ID’lerinin bulunduğu “shops.csv” dosyamızın içeriğini kontrol ediyoruz. Elimizde bulunan 60 adet firmanın ID’leri ile eşleştirildiği tabloda yapılan kontroller sonucu bir hata olmadığı görünmekte. Kontrol ettiğimiz veri kümemiz “shops.csv” dosyası gibi küçük boyutlu ise ID’lerinin benzersiz (unique) olup olmadığını dosyayı bir “.csv görüntüleyici” kullanarak açıp inceleyebilirsiniz (Şekil 6.6). Fakat ürünlerin isim, ID ve kategori ID’lerinin bulunduğu “item.csv” dosyamız gibi büyük boyutlu dosyalarımızda bu bilgiyi görmek biraz daha zordur. Bunun için “Libre Office Calc” (bknz. Bölüm 4.2) uygulamasından faydalandık. İlk satırdan son satıra kadar “item\_id” özelliğimizi çizdirdiğimizde Şekil 6.7’de de

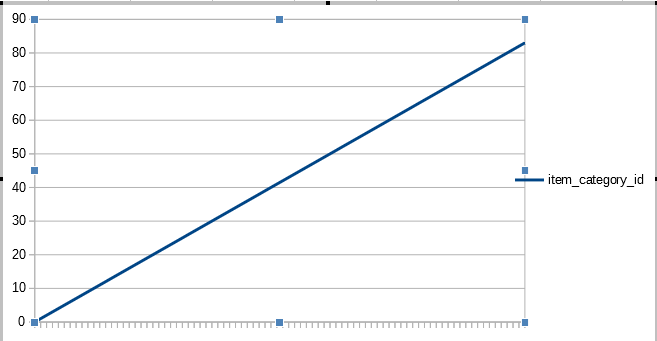
görüldüğü üzere grafikte bir kırılma olmamıştır. Buda demek oluyor ki ID’ler tekrar etmiyor ve o aralıkta bütün değerleri alıyor.

Şekil 6.6. LibreOffice Calc’da firma ID’lerinin çizgi grafiği ile doğruluk kontrolü



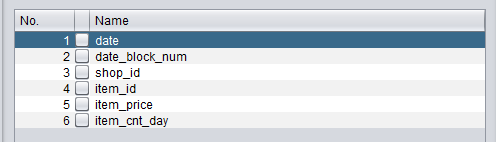
Şekil 6.7. LibreOffice Calc’da “item\_id” özelliğinin sütun grafiği ile doğruluk kontrolü

Aynı şekilde ID benzersizlik kontrolümüzü ürün kategorilerimizin ve bunlara ait ID bilgisinin tutulduğu “item\_categories.csv” dosyamızda da gerçekleştiriyoruz.



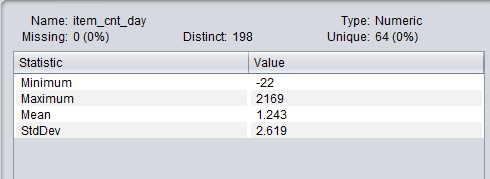
Şekil 6.8. “item\_category\_id” özelliğinin doğruluk kontrolü

Eğitim kümemizde kullanılan parametrelerin bazılarının doğruluk değerlerine bu işlemler sonucu bakmış olduk. Bu sayede artık eğitim kümemizin diğer parametrelerindeki gürültülü verileri kontrol etmeye başlayabiliriz. Bu özelliklere bakmak için Weka uygulamasından faydalanacağız (bknz. Bölüm 4.1). Öncelikle eğitim verimizi Weka uygulamasında açıyoruz. (Şekil 6.9)



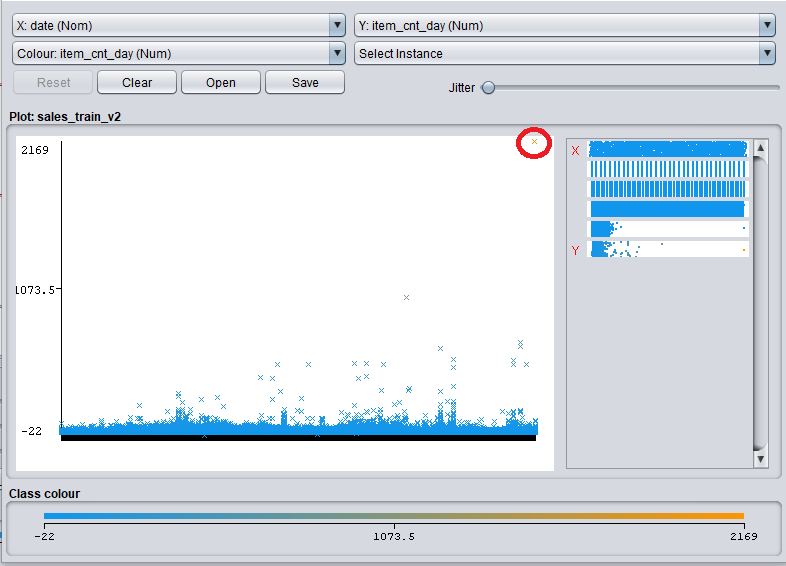
Şekil 6.9. Weka uygulamasında eğitim kümesinin genel görünümü

Veri kümesini okutma işleminden sonra (şekil 6.9’daki gibi) eğitim kümemizdeki verilerin özelliklerinin listelendiği bir bölüm karşımıza çıkıyor. Burada eğitim kümemizin özelliklerini görebiliyoruz. Numerik verilerimiz için minimum değer, maksimum değer, veri boyutu, kaç tane verimizin birbirinden farklı olduğu, aritmetik ortalama, standart sapma gibi hesaplamaları Weka bizim için yapıyor. Örneğin Şekil 6.10’da Günlük ürün satışlarının gösterildiği özelliğimizde verilerin 198 farklı değer aldığını (), verilerin numerik olduğunu, özellikteki tüm verilerin aritmetik ortalamasının (mean) 1.243 olduğunu, standart sapmanın (standard deviation) 2.619 olduğu gibi özellikleri görebiliyoruz.



Şekil 6.10. “item\_cnt\_day” özelliğinin weka üzerinden genel görünümü

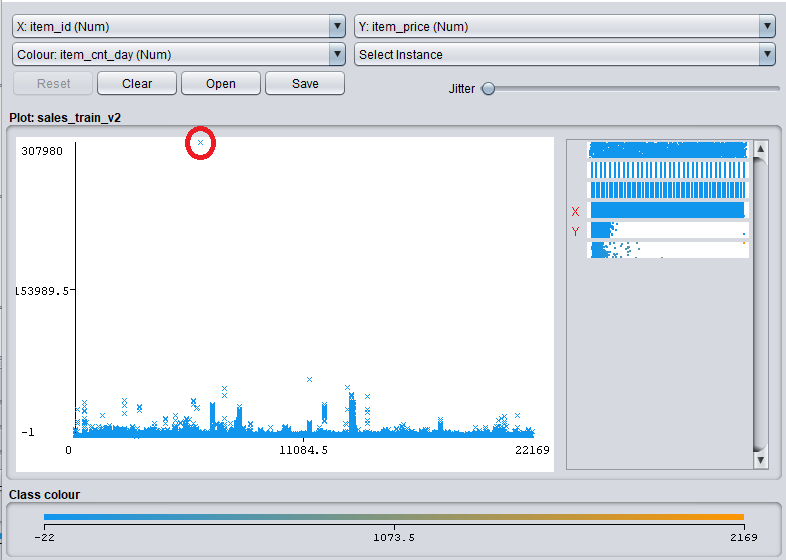
Şekil 6.10’da göründüğü üzere veri kümemiz içerisinde “item\_cnt\_day” özelliğinin aldığı en küçük değer olarak -22 görünüyor. Bir gün içerisinde satılan ürün sayısı negatif bir değer çıkamayacağından dolayı bu veri özelliğimizde gürültülü veri veya veriler olduğu anlaşılıyor. Bu gürültülü verilere daha yakından bakmak için Weka uygulamasından kolayca oluşturduğumuz iki boyutlu veri dağılım grafiği sayesinde gürültünün nerede ve ne boyutta olabildiğini görebiliriz (Şekil 6.11).



Şekil 6.11. Weka ile çizdirilen veri dağılım grafiği (günlük ürün satış miktarı için)

Bu grafik çiziminde x ekseni için “date” özelliği, y ekseni için “item\_cnt\_day” özelliği tercih edilmiştir. Görüldüğü üzere bir tane nokta (kırmızı ile daire içine alınmış) veri kümesinden çok farklı davranmaktadır. Buradan anlıyoruz ki burada bir gürültü mevuttur. Onun dışında grafikten bazı verilerin negatif değerler aldığını görebiliyoruz. (Bu durumun ne tip hatalara sebebiyet verdiğini görmek için bknz. Bölüm 5. Veri analizi)

Aynı şekilde diğer özelliklerimize de teker teker bakıyoruz ve ürünlerin fiyatı haricinde başka özelliklerde gürültü verisine rastlanmamıştır. Problemin olmadığı özellikler konunun yeterince anlaşıldığı düşünüldüğünden ve karmaşıklıktan kurtulmak adına çalışma içerisinde yer verilmemiştir. Fakat yukarıda da bahsedildiği üzere ürün fiyatında da benzer bir problem vardır. Bununla alakalı grafikte benzer şekilde aşağıda verilmiştir (Şekil 6.12).



Şekil 6.12. Weka ile çizdirilen veri dağılım grafiği (ürün fiyatları için)

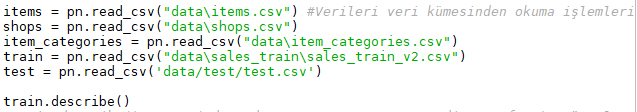
Şekil 5.12’de de görüldüğü üzere “kırmızı daire” ile işaretlenen veri ve fiyatına sahip veriler normal veri dağılımına göre çok farklı davranış sergilemektedir dolayısıyla aykırılık mevcuttur. Diğer fiyat dağılımları birbirine yakın olmasına rağmen kırmızı ile işaretlenen verinin büyüklüğü nedeniyle ortalama bir hayli yüksek çıkmıştır buda bizim öğrenme algoritmalarımızı gerçekledikten sonra büyük bir problem olarak karşımıza çıkabilir. Bundan dolayı bu gürültü verilerinden kurtulmamız gerekir.

## Veri Temizleme

Hatalı veriler hakkında bilgi tez kapsamında verilmiştir (bknz Bölüm 5). Verimizi analiz ettiğimiz bölümde günlük veri satışı ve ürün fiyatları özelliklerinde gürültülü ve aykırı veriler olduğunu görmüştük. Bu aşamada bunlardan kurtulmaya çalışacağız.

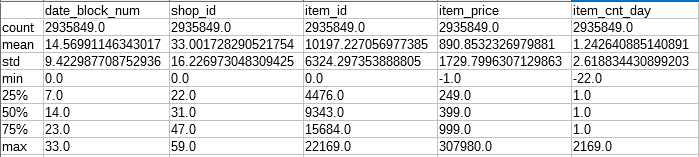
Programlama dili olarak Python dilini IDE olarak ta Spyder ve Jupyter Notebook’u kullandık.

Python da işlemlerimize veri kümemizi pandas ile okutarak başladık. Daha sonra eğitim verisi hakkında genel bir bilgi almak için “describe” fonksiyonunundan faydalandık (Şekil 6.13).



Şekil 6.13. Verilerin okutulması ve veriler hakkında genel bilgi için Python kodları

Şekil 6.13’ te yazdığımız kodların çıktısını LibreOffice Calc’a yazdırdığımızda şekil 6.14’teki gibi bir çıktı ile karşılaştık.

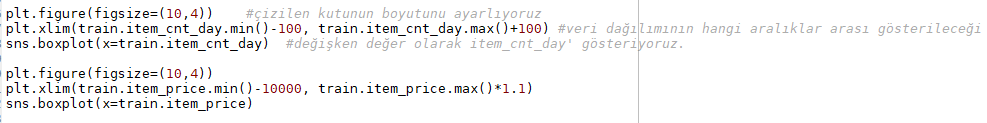


Şekil 6.14. Eğitim kümesindeki hatalı verileri temizlemeden önceki verilerin analizi

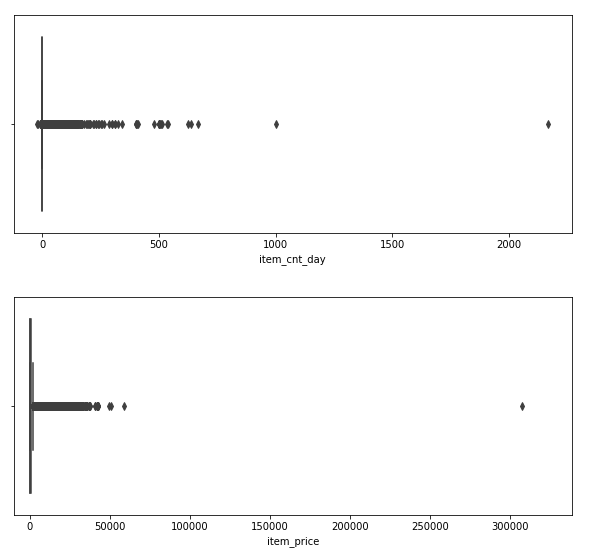
Buradan daha önceden Weka ile yapmış olduğumuz gibi eğitim kümemizi yorumlayıp analiz edebiliriz. Şekil 6.14’te de görüldüğü üzere ürün fiyatlarında ve günlük satılan ürün sayısı hatalı veriler içermektedir. Bu özelliklerin dağılımını ve hatalarını birde box plot yöntemli ile inceleyelim.

### Python’da Box Plot Yöntemi ile Hatalı Verilerin Analizi ve Temizlenmesi

Python’da bulunan “seaborn” kütüphanesi yardımıyla günlük satılan ürün miktarı ve ürünlerin fiyatları hakkındaki bilgileri içeren özellikleri box plot yöntemiyle çizdiriyoruz.



Şekil 6.15. Verinin Box Plot Yöntemi ile analizi için gerekli Python kodları

Şekil 6.15’teki kodun çıktısı olarak grafikleri elde ettik. Bu özelliklere ait box plot gösterimleri şekil 6.16’da görülmektedir. 

Şekil 6.16. Şekil 5.15’deki kodun çıktısı

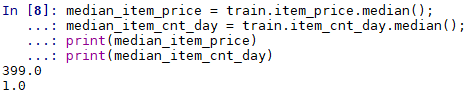
Şekil 6.16’ya baktığımızda buradan veri üzerinde aykırı değerler olduğunu görebiliyoruz. Şekil 6.16’da üstte bulunan grafikte günlük ürün satış miktarı 0’ın altına inmektedir. Ayrıca bu grafik üzerinden anlayamadığımız bir firma tarafından bir günde 2500 civarında ürün satıldığı görünmektedir. Bu durumun gerçekleşme olasılığı olmasına rağmen algoritma doğruluğu açısından 1000 üzeri satış yapıldığı tespit edilen kısımları aykırı kabul etmeliyiz.

0 <= satış sayısı <=1000 (6.1)

Şekil 6.16’da alt kısımda bulunan grafikte de ürün fiyatları gösterilmektedir. Burada da ürün fiyatının 0 olarak görüldüğü veriler vardır. Ayrıca 300000’ in üzerinde fiyatı olan üründe görülmektedir. Bunları da aykırı veri olarak kabul ediyoruz. Biz bu aykırılık aralığını eşitsizlik 6.2 de ki gibi kabul edelim.

0 < ürün fiyatı <= 60000 (6.2)

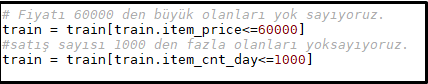
Şekil 6.16’ya baktığımızda günlük ürün satış miktarlarının [0-750] civarında yoğunlaştığını ürün fiyatlarının da [0-50000] arası yoğunlaştığını görebiliyoruz. Az önce belirlediğimiz aralıklarda veriyi tekrar düzenlemek için belirlediğimiz aralıklar dışında kalan verileri yok sayabiliriz ya da bu verilere ortalamaya, medyana, mod değerine göre değerler atayabiliriz (bknz bölüm 5. Veri analizi).



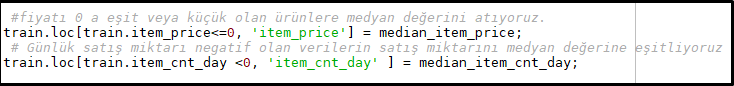
Şekil 6.17. Python’da veri kümemizdeki günlük ürün satışı ve ürün fiyatlarının medyan değeri

Şekil 6.17’de de görüldüğü üzere günlük ürün satışında medyan değerimiz 1 iken ürünlerin medyan değeri 399 olarak bulunmuştur. Bu değerleri az önce veri kümemizde tespit ettiğimiz aykırı verilerin yerine kullanılabilir.

Biz şimdilik daha önceden söylediğimiz yöntemler olan “aykırı verileri yok sayma” ve “aykırı veriler yerine kabul edilebilir değerler atma” işlemlerinin ikisini de gösterebilmek için ikisini farklı kısımlarda kullanalım.



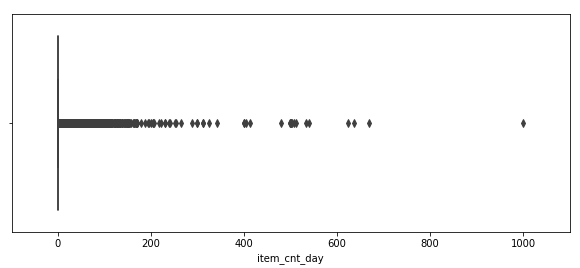
Şekil 6.18. Aykırı değerlere göre veri kümesinin tekrar düzenlenmesi

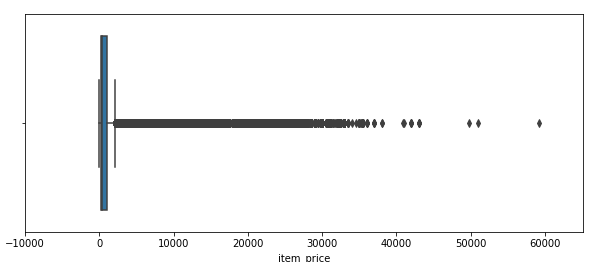
Şekil 6.18’da görüldüğü gibi eğitim verimizde ürün fiyatının 60000’in üzerinde olanlar ve günlük satış sayısı 1000’den fazla olan veri blokları yok sayılıyor. 

Şekil 6.19. Aykırı değerlere göre veri kümesinin tekrar düzenlenmesi

Şekil 6.19’de yapılan işlem ürün fiyatı 0’dan küçük veya eşit olan verilerin ürün fiyatlarını o bloğun medyan değeri ile değiştirme işlemidir. Aynı işlem günlük ürün satış miktarı içinde tekrarlanır.

Yapılan bu işlemler sonucunda şekil 6.16’daki box plot gösterimi şu şekilde değişecektir. (Şekil 6.20)

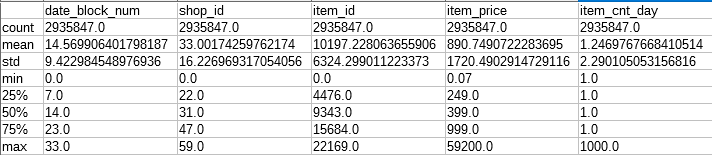
Şekil 6.20. Günlük satılan ürün miktarı fiyatlarının yapılan aykırı veri temizleme işlemi sonrası box plot yöntemi ile gösterimi



Şekil 6.21. Ürün fiyatlarının yapılan aykırı veri temizleme işlemi sonrası box plot yöntemi ile gösterimi

Şekil 6.20’da ve şekil 6.21’de de görüldüğü üzere veri kümemizi denklem 6.1 ve denklem 6.2’de belirlediğimiz aralıklar ile sınırlandırdık ve aykırı olarak kabul ettiğimiz verilerde düzenlemelere gitmiş olduk. Bu veri dağılımını da box plot yöntemleri ile gösterdik.

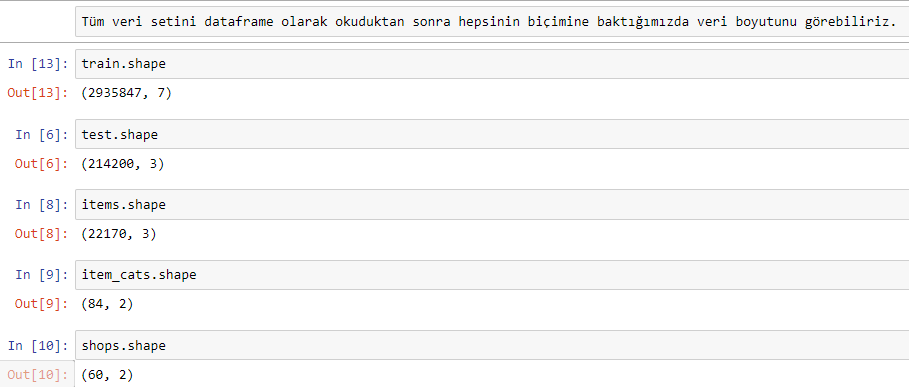
Yapmış olduğumuz verideki hatalı verileri temizleme işlemlerinden sonra eğitim kümemizin dağılım özelliklerine şekil 6.14’te yapmış olduğumuz gibi tekrar bakalım.



Şekil 6.22. Eğitim verimizde hatalı verileri temizledikten sonraki verilerin dağılımı

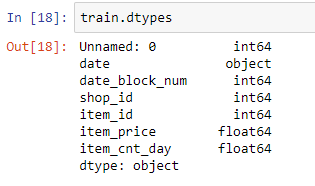
Python’da tekrar “describe” fonksiyonunu çalıştırdığımızda şekil 6.22 deki çıktıya ulaşıyoruz. Burada şekil 6.14 ile şekil 6.22’deki sonuçları karşılaştırdığımızda ortalamanın, standart sapmanın, minimum ve maksimum değerin değiştiğini görebiliyoruz. Bu işlemler sonucu verimizi hatalı verilerden arındırmış olduk.

Veri setin de yapmış olduğumuz ön işleme adımlarından sonra son değerlere daha detaylı olarak şu şekilde bulabiliriz. (Şekil 6.23)



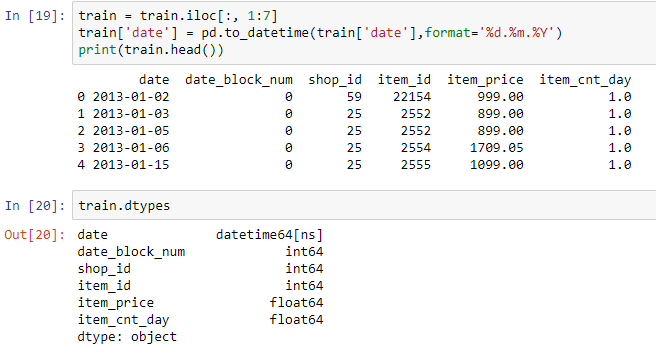
Şekil 6.23 Ön işleme sonrası veri setinin boyutu

Python’da model oluştururken veri özelliklerinin (feature) ayrık veriler olması gerekiyor. Bizde eğitim kümemizin durumunu görmek için Şekil 6.24’te bu özelliklerin tiplerini analiz ediyoruz.



Şekil 6.24. Train kümesi sütunlarının tipleri

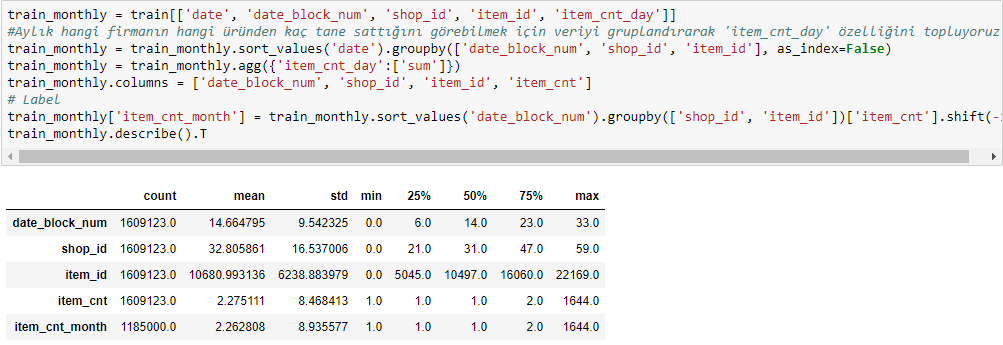
Şekil 6.24’te gördüğümüz üzere tarih (date) özelliğinin tipi object olarak karşımıza çıkıyor. Bu veri setine zaman serisi analizi yapabilmemiz için öncelikle bu özelliği tarih tipine çevirmemiz gerekiyor.



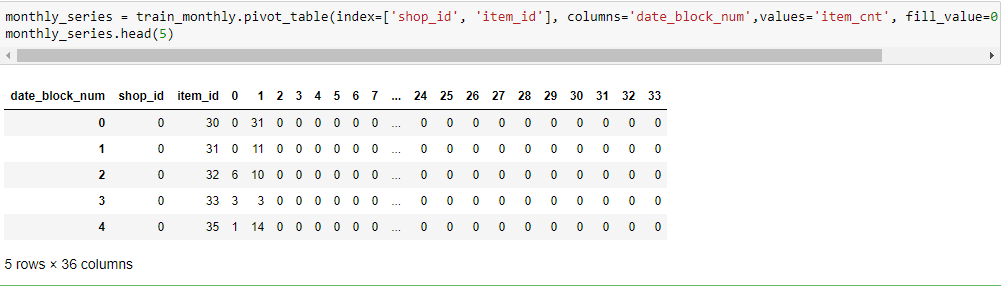
Şekil 6.25. Date tipini değiştirme

## Veriyi Etiketli Veri Haline Getirme

Veriye LSTM, SVR vb. gibi modeller uygulayabilmek için öncelikle veriyi etiketli veri haline getirmemiz gerekiyor. Etiket olarak ürünün aylık satış tutarını etiket olarak ürünün aylık satış miktarını hazırlıyoruz. Bu işlemler aşağıda Şekil 6.26 da gösterilmiştir.



Şekil 6.26. Zaman serisi eğitim kümesini etiketli hale getirme



Şekil 6.27. Etiketli zaman serisi

Hazır olan bu veri seti üzerinde model oluşturma ve performans ölçümleri Bölüm 9’da gerçekleştirilmiştir.

# LİTERATÜRDE BAHSEDİLEN YÖNTEMLER

## Regresyon Analizi

Aralarında sebep-sonuç ilişkisi bulunan iki veya daha fazla değişken kullanarak çıkarım veya tahmin üretme üzerine tanımlanır. Analiz tekniklerinde iki değişken arasındaki ilişki modeline **basit regresyon modeli,** ikiden fazla değişkenlerin aralarındaki ilişki modeline **çoklu regresyon** **modeli** denir. [46]

### Basit Lineer Regresyon Modeli

Denklem 6.1 bağıntı modeline sahip, bir bağımlı bir bağımsız değişken olmak üzere 2 değişkene sahip modeldir.

Y=α + βX + ε (7.1)

**Y**; bağımlı (sonuç) değişken

**X**; bağımsız (sebep) değişken

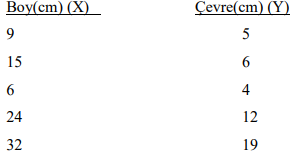
**α**; X=0 olduğunda Y’nin aldığı değer

**ß;** regresyon katsayısı

**ε**; tesadüfi hata terimi

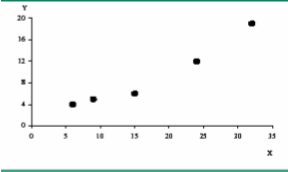
**Parametrelerin (Katsayıların) Tahmini**

Bir regresyon modeli oluşturulurken en küçük katsayılar yöntemi sıkça kullanılan yöntemlerden biridir.

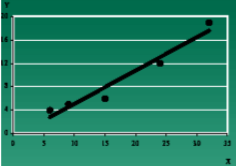


Şekil 7.1. En küçük kareler yöntemi için kullanılacak veri tablosu [47]

Şekil 7.1’de veriler X ve Y değişkeni altında verilmiş olup, en küçük katsayılar yönteminde kullanılacaktır.



Şekil 7.2. Şekil 7.1’deki verilerin dizilme örneği. [48]



Şekil 7.3. 7.2'deki dizilen verilerin üzerine regresyon doğrusu oluşturma örneği. [48]

Şekil 7.3’de grafiğe yerleştirilmiş olan verileri temsil eden regresyon doğrusu oluşturulmuştur. Regresyon doğrusu ile noktalar arasındaki farklar fark hata (ε) olarak isimlendirilir. En küçük katsayılar yöntemi fark hata (ε) değerinin en aza indirgenmesi sağlar. [49]

**Tek Değişkenli ve Çok Değişkenli Regresyon Analizi**

Bir bağımlı değişken ve bir bağımsız değişken arasında oluşturulan ilişkiye tek değişkenli regresyon analizi, bir bağımlı biden fazla bağımsız değişken arasında oluşturulan ilişkiye çok değişkenli regresyon analizi denir.

**Tek Değişkenli Regresyon Analizi**

Biri bağımlı ve biri bağımsız değişkenler arasındaki ilişkidir. Bu modele tek değişkenli regresyon modeli denmektedir. Bağımlı ve bağımsız değişkenler arasındaki ilişki doğrusal denklemler ile tanımlanmaktadır.

Regresyon analizi sonuçlarının yorumlanmasında birçok hata yapılabilmektedir. En çok yapılan hata; X bağımsız değişkeninin Y bağımlı değişkenine sebep olduğudur. Ancak X bağımsız değişkeni ile Y bağımlı değişkeni arasında doğrusal bir ilişki olması her zaman bağımsız değişkenin bağımlı değişkenin sebebi olduğu anlamına gelmez.

### Lojistik Regresyon Analizi

Bağımsız değişkenler içeren veri kümesinin analizini yapmak için kullanılan bir istatistiksel analiz yöntemidir. Lojistik regresyonda, lineer regresyon için geçerli kıstasların hiçbirinin aramadığı için kullanıcılar tarafından çokça tercih edilmektedir. Lojistik regresyon ile doğrusal regresyon arasındaki en büyük fark; bağımlı değişkenin lojistik regresyon modelinde kategorik olmasıdır. [50]

(7.2)

Odds kavramı, bir olayın olma olasılığı ile olmama olasılığının bölümünde elde edilmektedir. P(x) fonksiyonunu olayın olma olasılığı olarak tanımlarsak, olayın olmama olasılığı 1-p(x) olmaktadır. [51,52]

Odds p(x) / (1- p (x)) (7.3)

olarak tanımlanmaktadır.

logit(p) = b0 + b1X1 + b2X2 + …… + bKXK  (7.4)

Denklem 7.4’de verilen denklem p’nin ortaya çıkma olasılığını göstermektedir.

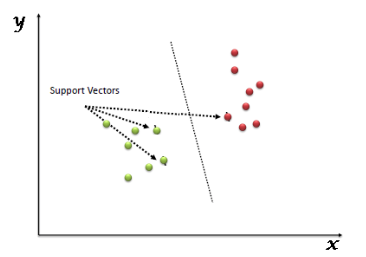
Lojistik regresyon analizinin merkezinde, bir olayın logit oranını tahmin eden görev bulunur. Matematiksel olarak, lojistik regresyon aşağıdaki gibi tanımlanmış çoklu doğrusal regresyon fonksiyonunu tahmin eder. Bu fonksiyon denklem 7.4’te gösterilmiştir. [53]

## Destek Vektör Makineleri (SVM)

Destek vektör makineleri; istatistiksek öğrenmeye dayalı gözetimli algoritmadır. Destek vektör makineleri, farklı sınıflardaki verileri birbirlerinden ayırmak için kullanılır. Bunun yapılabilmesi için karar sınırları belirlenir.

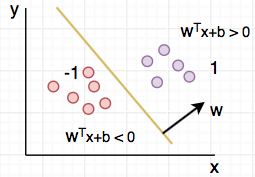
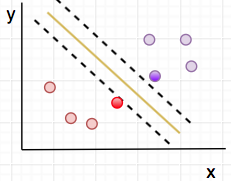
### Doğrusal Destek Vektör Makineleri

Doğrusal destek vektör makinelerinin amacı; 2 farklı sınıfın, belirlenen fonksiyonlar yardımıyla birbirinden ayrılması amaçlanır. Veri setini 2 ye ayıran doğruya karar doğrusu adı verilir.



Şekil 7.4. Karar doğrusunu gösterimi [54]

Karar çizgisine en yakın noktalar, destek noktası olarak kabul edilir. Yeni bir veri eklendiğinde, veri hangi kısma eklenecekse dengeleme ekleme sonrası tekrar yapılır. [54]

Şekil 7.6’da veri girişi gösterilmektedir. Şekil 6.7’de gösterilen kesikli çizgiler sınır düzlemlerini ifade eder. Sınır düzlemlerini tam ortasından geçen, sarı ile gösterilen düzleme hiper düzlem denmektedir. [55]

Şekil 7.5. Sınır düzlemlerini gösterme

Şekil 7.6. SVM için veri girişleri

### Doğrusal Olmayan Destek Vektör Makineleri

Doğrusal olmayan destek vektör makineleri bir veri kümesi içinde verileri hizaladıktan sonra hiper düzlem çizemez. Bu nedenle makine kernel trick kullanır. Kernel trick, kümedeki çekirdek numarasıdır.

En çok kullanılan çekirdek yöntemleri:

* Polynomial Kernel
* Gaussian RBF (Radial Basis Function) Kernel

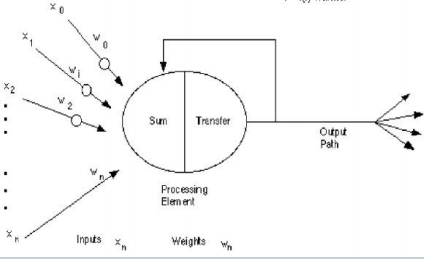
## Yapay Sinir Ağları Yöntemi

Yapay sinir ağları insan beyninden esinlenilerek ortaya çıkan bir yöntemdir. Öğrenme sürecinin matematiksel olarak modellenmesine dayanır. Beyin içindeki nöronlar ile ilgili çalışmalar yapay sinir ağları metodunun ilk çalışmalarıdır. Yapay sinir ağlarının çalışma yapısı ve öğrenmesi nöronlar ile benzerlik göstermektedir. [56]

### Yapay Sinir Ağları Yapısı

Hücre gibi düşünürsek, bilgi ilk olarak dış ortamdan alınır ve hücreye bağlanır. Devamını yöntemsel olarak düşünürsek, aktivasyon fonksiyonlarının çalışmasıyla net çıktı verileri elde edilir. Yapay sinir ağları, kendisine yeni veriler ve örnekler gösterildiğinde bu ağırlık değerleri güncellenir ve değişir. Hataya toleranslı bir yapıya sahiptir. Bir hücrenin bozulması veya yanlış çalışması durumunda ağ çalışmaya devam eder.  
Yapay sinir ağları ile;

* Regresyon
* Tahmin Yürütme
* Karar Verme
* Zaman Serisi Analizi
* Sınıflandırma
* Kontrol Benzetimi
* Optimizasyon
* Doğrusal olmayan sinyal işleme
* Doğrusal olmayan sistem modelleme gibi işlemler yapılabilmektedir.



Şekil 7.7. Yapay sinir ağları örnek modeli [57]

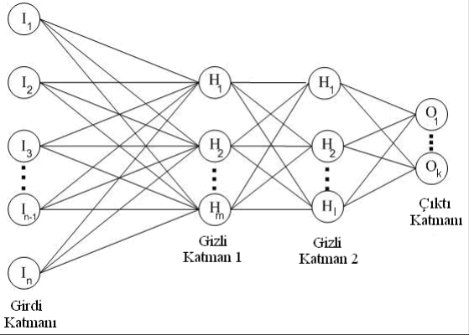
Fonksiyonun işlevi, hücreye gelen veriyi işleyerek çıktının ne olacağını belirlemesidir. Bu fonksiyona aktivasyon fonksiyonu denir.

### Yapay Sinir Ağlarının Türleri

Yapay sinir ağları yöntemi, ileri beslemeli yapay sinir ağları ve geri beslemeli yapay sinir ağları olmak üzere ikiye ayrılır**.** [57]

**İleri beslemeli yapay sinir ağları**

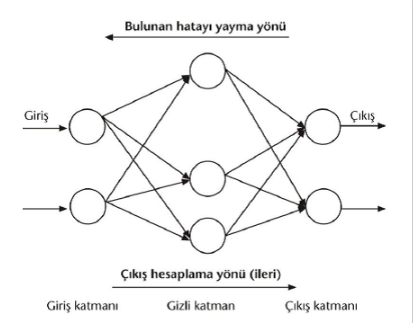
İleri beslemeli yapay sinir ağları modelinde 3 adet temel katman vardır. Bunlar girdi, gizli ve çıkış olarak sıralanmaktadır. Girdi katmanında veri alınır, gizli katmanı kendi içinde işlemi yapar. İşlem sonucunu öğrenir ve kendini sonuca göre eğitir. Ardından sonuç katmanında çıkış değerleri görülmektedir. [58,59]



Şekil 7.8. İleri beslemeli YSA [60]

**Geri beslemeli yapay sinir ağları**

Geri beslemeli ağlardan çıkan sonuç başka bir nörona girdi olarak verilebilir. Geri beslemeli yapay sinir ağları yöntemi doğrusal olmayan bir çizgi izler. Çıktının sonucundan sonra veri hata yayma yönünde ilerler. [61]

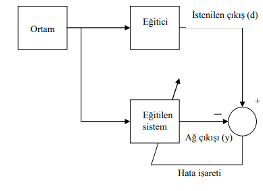


Şekil 7.9. Geri beslemeli YSA [61]

### Yapay Sinir Ağları Öğrenme Şekillerine Göre Sınıflandırma

**Eğiticili Öğrenme**

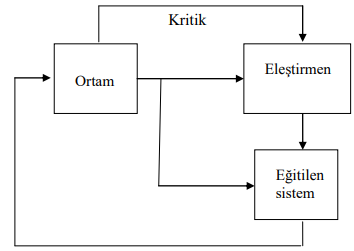
Çıktı verisi verilir. Ardından verilen çıktı verisi ile istenilen çıktı verisi arasında kıyaslama yapılır. Bu kıyaslamanın sonucuna göre nöronlar arası bağlantı ağırlığı yeniden ayarlanır ve güncellenir. Algoritmanın düzgün ve kusursuz çalışması için eğiticiye ihtiyaç vardır. Bu eğiticiler, bilgisi olmayan eğitilen sisteme bilgiyi aktarır. İşlem yapılan sisteme eğitilen sistem denir. [62]



Şekil 7.10. Eğiticili öğrenme blok diyagramı [62]

**Eğiticisiz Öğrenme**

Eğiticisiz öğrenmede, öğrenme işleminin sonucu önceden bilinmez. Öğrenme işlemi içinde bazı değişken ve özellikler yeniden düzenlenir. Bu yaklaşımın amacı, birbirlerine benzeyen değişken ve özellikleri belirli bir kritere göre gruplandırmaktır. [63]



Şekil 7.11. Eğiticisiz öğrenme blok diyagramı

**Özdenetimli Öğrenme**

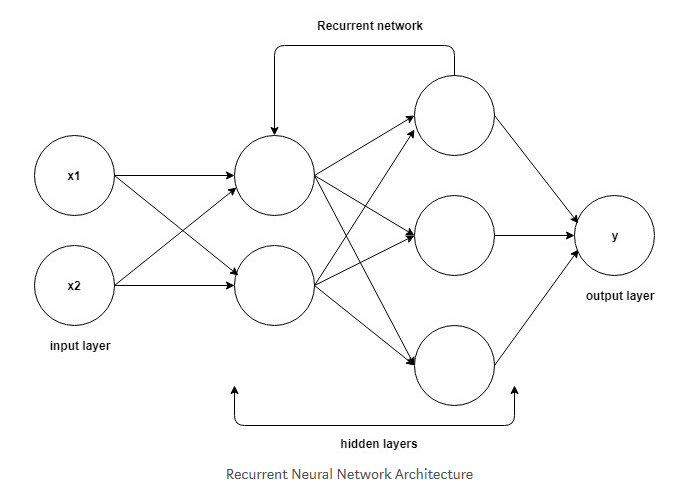
Özdenetimli öğrenmede, herhangi belirli bir çıkış değeri yoktur. Bu öğrenme yöntemindeki amaç, birbirlerine benzer veya yakın değişkenleri düzenli bir şekilde ve belirli değer aralıkları arasında gruplamaktır. [64]



Şekil 7.12. Özdenetimli Öğrenme blok diyagramı

## Tekrarlamalı Sinir Ağları (RNN)

RNN’ler birer sinir ağı yapılarıdır. Bu ağlarda çıkış verileri bir sonraki adımın giriş verileri olmaktadır yani bir önceki adımın çıkış değerleri, bir sonraki adımın çıkış değerlerini etkilemektedir. RNN’ler ile feedforward (ileri beslemeli) yapılar arasındaki farklardan biri de budur. Tekrarlamalı ağlar birer bellek yapılarına sahiptir. RNN’ler çıktı üretirken aynı veriyi gizli katman adı altındaki katmana da atar. Bu sayede veriyi belleğe kaydetmiş ve bir sonraki girdide kullanmak için bekletmektedir. [83]

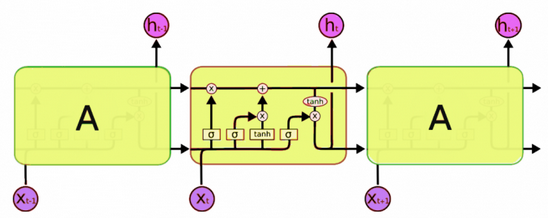


Şekil 7.13. RNN örneği [84]

Şekil 7.13’de görüldüğü üzere “input” yani giriş değeri gelir ardından çıkış değeri gerekli işlemler sonucu oluşturulur ve çıkış değeri gizli katmana tekrar döndürülür ve bir sonraki adım için tutulur.

## Uzun Kısa Süreli Bellek (LSTM)

Uzun dönem işlevleri öğrenebilen bir RNN türüdür. Uzun kısa süreli bellek, daha önceden yapılan işlemler sonucu ortaya çıkan değerleri rastgele zaman aralıklarında hatırlayan bir mimaridir. Daha önce yapılan işlemler hatırlanabilmesi nöronların arasından ileri ve geri beslemeye izin verilmesiyle yapılmaktadır. [65] Kısa bir problem çözümünde, problemi çözmek RNN ile mümkün olsa da karşılaşılan uzun ve karmaşık bir problemde RNN yetersiz kalabilmektedir. LSTM ağları RNN ağlarına nazaran bilgiyi öğrendikten sonra daha uzun süre kendi belleği içerisinde tutmaktadır. Bu sayede karmaşık problemlerin üstesinden gelebilmekte ve uzun vadede oluşabilecek olan ihtiyaç sorunlarının da önüne geçilmiş bulunulmaktadır. [73]



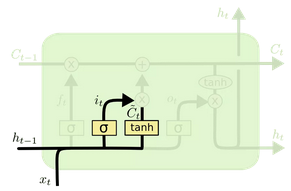
Şekil 7.14. LSTM örneği

LSTM ağlarını RNN ağlarında ayıran bir özellik de’ RNN tek katmana sahipken, LSTM yapısının 4 adet katmanı vardır. Bu katmanlara kapı ismi verilmektedir. Kapının işlevi sadece veriyi dışarı çıkarmak değil, veriyi dışarıdan almak da vardır. Kapılar verileri nerede ve nasıl kullanılacağını belirler.[66,67]

LSTM yapısının içerisinde 4 adet kapı bulunmaktadır.

* Giriş Kapısı Katmanı: Girişi hücreye ölçeklendirir (yaz)
* Çıkış Kapısı Katmanı: Çıktıyı hücreye ölçeklendirir (okunur)
* Unutturma Geçiş Kapısı Katmanı: Eski hücre değerini ölçeklendir (sıfırla)
* Sigmoid Katman: Her bileşenden ne kadarının geçeceğini belirleyen katmandır.

LSTM’e gelen bilginin hangi kısımlarının kullanılacağı belirlenir. Bu belirlemeleri LSTM içerisindeki kapılar yapmaktadır. Giriş kapısına giren veririn hangi kısımlarının kullanılıp hangi kısımlarının atılacağı belirlenir. Belirleme sonrası Tahn tabakası adındaki tabaka eksilen verileri doldurmak amacıyla vektörler oluşturur.

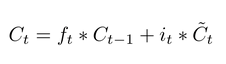


Şekil 7.15. Tahn tabakası [73]

Şekil 7.15’de de görüldüğü üzere giriş kapıları ve tanh tabakası görülmektedir. Gelen bilgiler giriş kapıları ve tanh tabakasından geçtikten sonra bir sonraki hücre durumuna geçer.

* Unutturma geçiş kapısı: “f”
* Hücre durumu: “C’’
* Tanh: Hiperbolik tanjant aktivasyon fonksiyonu
* Girdi: “I”
* Çıktı: “O”
* Gizli state “H”
* Hafıza state “C” olarak ifade edilmektedir.

Daha önceden bahsetmiş olduğumuz giriş verisinde kullanılmayan diğer bir deyişle unutulmaya çalışan veriler çıkarılarak f ile çarpılır. Ardından I(girdi)\*C(Hafıza) işleminin sonucuna eklenir. Bu değer, unutulan verilerin yerine gelecek yani güncellenecek değerlere benzer bir değerdir.



Şekil 7.16. Yeni girdi ekleme [75]

Şekil 7.16’da görüldüğü üzere yeni bilgi ekleme işlemi yapılmaktadır. Yeni bilgiye ulaşmak için f (unutturma geçidi) değeri ile önceki değer çarpılır. Ardından C’ (hücre durumu) değeri ile I (girdi) değerleri çarpılarak elde edilen değerler toplanır. En sonda elde etmek istediğimiz değer, işlemlerden geçmiş bir değer olacaktır. Bu değere ulaşmak için sigmoid katmanları çalıştırmak gerekmektedir. Gerekli işlemler yapıldıktan sonra elde edilen veri, gereksiz ve kullanılmayan kısımları çıkarılmış gerekli veridir.

Python’da Keras ile LSTM modeli oluşturulurken ki LSTM parametreleri [76]:

* Units: Çıktı alanının boyutunu verir, pozitif bir integer değerdir.
* Recurrent\_activation: Tekrarlayan adım için kullanılacak aktivasyon foksiyonu belirtir.
* Use\_bias: Katmanda bias vektörü kullanılıp kullanılmadığını belirtir.
* Kernel\_initializer: Girdilerin doğrusal dönüşümü için kullanılır.
* Recurrent\_initializer: Tekrarlanan durumdaki başlangıç durumunu belirtir.
* Bias\_initializer: Bias vektörünün başlangıcını belirtir.
* Unit\_forget\_bias: Eğer ‘True’ değeri döndürürse, unutturma geçiş kapısına 1 ekler ve devam eder.
* Kernel\_regualizer: Kernel ağırlıklarını düzenleyen fonksiyonu belirtir.
* Recurrent\_regulizer: Tekrarlayan kernel ağırlık matrisindeki düzenleyici fonksiyonu belirtir.
* Bias\_regularizer: Bias vektörünü uygulanan fonksiyonu belirtir.
* Activity\_regularizer: Katman çıktısına uygulanan düzenleyici fonksiyonu belirtir.
* Kernel\_constraint: Kernel ağırlık matrisi üzerindeki kısıtlayıcı fonksiyonu belirtir.
* Recurrent\_constraint: Tekrarlayan kernel ağırlık matrisindeki kısıtlayıcı fonksiyonu belirtir.
* Bias\_constraint: Bias vektörüne uygulanan kısıtlayıcı fonksiyonu belirtir.
* Dropout: Girişlerin doğrusal dönüşümü için düşecek birimlerin bölünmesidir.
* Recurrent\_dropout: Tekrarlayan durumun lineer dönüşümü için düşecek birimin bölünmesidir. 0 ve 1 arasında değerler alır.
* Recurrent\_dropuots: Tekrarlanan durumlarda düşülecek olan birimlerin bölümünmesidir. 0 ve 1 değerlerini alır.
* İmplementation: 1 veya 2 modundan oluşmaktadır. 1 modu, daha fazla sayıda küçük işlemler yapar. 2 modu ise daha az sayıda büyük işlemler yapmaktadır. Mod 1 ve 2 arasında işlem farklılıklarından dolayı performans farkı da ortaya çıkabilmektedir.
* Return\_sequences: Ortaya çıkan son çıktının çıktı sırasına veya listenin son sırasına atılacağını belirler.,
* Return\_state: Çıktının bir sonraki işlem için döndürülüp döndürülmeyeceğini belirler.
* Go\_backwards: ‘True’ ve ‘False’ değerlerini alır. True değeri aldığında giriş sırasını terine çevirir ve döndürülmüş sırayı döndürür. False değeri etkisizdir (Varsayılan).
* Stateful: ‘True’ ve ‘False’ değerlerini alır. True değeri döndüğünde son durum için çıkan değer bir sonraki durumun başlangıç değeri olarak alınır. False değeri etkisizdir(Varsayılan).
* Unroll: ‘True’ ve ‘False’ değerlerini alır. True değeri döndürüldüğünde ağlar arasındaki bağlantı kaldırılır. Ancak bu durum uzun sıralar için uygun değildir.

## Zaman Serisi Analizi Yöntemleri

Zaman serisi analiz yöntemlerinde 4 ana analiz modeli vardır.

### Otoregresif (AR) Modelleri

AR modelleri, bir zaman serisinin herhangi bir dönemindeki değerin, aynı zaman serisi içinde başka bir döneme ait olan gözlem verilerini ve hata teriminin doğrusal olarak bileşimini anlatan bir modeldir. AR modelleri geçmiş dönem veri sayısına göre ifadelendirilir. ”1. dereceden”,”2. dereceden”, ”n. dereceden” gibi geçmiş dönem veri sayısına göre isimlendirilir.[10, 11]

x1 = φ1xt-1 + φ2xt-2 + … + φpxt-p  + at  (7.5)

xt, xt-1, xt-2, ….., xt-p değerleri gözlem değerleridir.

φ1, φ2,…., φP modelin parametreleridir, p modelin derecesini gösterir.

at, hata değişkenidir.

AR modelin çoklu regresyon modelinden temel farkı; çoklu regresyon modeli gibi bağımlı değişkeni kullanıp bağımsız değişkeni bulmak gibi bir yaklaşım değildir. AR modeli çoklu regresyon modeline kıyasla belirli bir x değerini daha önceki x değerleri ile açıklamaya çalışır. (bknz. Bölüm 7.1)

Birinci dereceden AR modelinin yazılımı AR (1) şeklindedir. Derece arttıkça parantez içindeki değer doğrusal olarak artmaktadır. Örneğin; 2.dereceden model ise AR (2), 5.dereceden model ise AR (5) olarak gösterilmektedir. [11]

AR (1) modelinin bir zaman serisinin t anındaki değeri aşağıdaki gibi açıklanmaktadır. xt t-1 değeri gözlem değeri, xt-1 ve at  hata terimi ile açıklanır.

x1= φ1xt-1 + at

### Hareketli Ortalama (MA) Modelleri

Hareketli ortalama yöntemi; Bir serinin geçmiş hata terimi, şimdiki hata terimini etkiliyorsa, bu durum hareketli ortalama sürecini tanımlamaktadır.

xt = e t – a1et-1 - …. ,t = 1,2,…n, (7.6)

E(xt) = μ (sürecin ortalaması)

Şeklinde ifade edilir. Şimdi ki değerler geçmiş hata terimlerinden etkilenmektedir. [11]

### Otoregresif hareketli ortalama (ARMA) Modelleri

Otoregresif hareketli ortalama modeli AR ve MA modelinin bazı özelliklerini bir arada kullanılması zaman modeline büyük rahatlık sağlamaktadır.

Yt = m+a1yt-1…..apyt-p+ut-but-1-…..-bqut-q (7.7)

m = Değişken

Yt = Sabit terim

ut = Varsayılan hatalar

Bize ARMA (p, d, q) sürecini göstermektedir. Denklemdeki m değişkeni yt nin sabit terimini, ut ise varsayılan hataları göstermektedir. [11]

### Box-Jenkins Modeli (ARIMA)

Otoregresif bütünleşik hareketli ortalama modeli (ARIMA) modeli, ARMA modeline benzer olarak AR ve MA modellerinin bazı özelliklerine sahiptir. ARIMA’nın ARMA’dan farkı (intergreted) isimli seri trendini içermesidir. AR modelinin parametresi p, MA modelinin parametresini q ve anlık işlem yapmayan bir seri ile karşılaşmamız da d’yi ifade eder ise ARIMA (p, d, p) modeli ortaya çıkmış olur. Aynı zamanda Box-Jenkins modeli olarak da bilinmektedir. [68, 69]

* Özdeşleme: p,q,d değerlerini bulmaktadır.
* Tahmin: Veriler belirlenen modelin üzeride işe yarayıp yaramayacağı tahmin edilir.
* Tanısal denetim: Verilerin modele tahmin sonrasıda uygun olup olmadığını inceler.
* Yordam: Denetimden geçen verinin yeterli olup olmadığı karar verilir.

wt = φ1wt-1+ φ1wt-1+……..+at – φ1at-1-φ2at-2-…….-φqat-q  Denklem7.8

wt = Farkı alınmış serisi,

d = Fark alma derecesi, şeklinde ifade edilir.

**Çift Üstel Düzeltme (Holt) Metodu**

Çift üstel düzetme, zaman serilerinde kullanılan bir yöntemdir. Doğrusal trendin izlenmesi için yaratılmış bir yöntemdir. Eşitlik kurulması için α (seri için) ve ß (trend için) değişkenlerinin kullanılması gerekmektedir. Eşitlik aşağıdaki gibi kurulmuştur. [70]

St = αDt + (1- α)(St\*1 + Gt-1) (7.9)

Gt = ß(St – St-1) + (1- ß)Gt-1  (7.10)

Denklem 7.9, t anındaki kesişme değer tahmini,

Denklem 7.10, t anındaki eğim tahmini şeklinde hesaplanır.

Genellikle ß ≤ α ilişkisi mevcuttur.

### Vektör Otoregresyon (VAR) Analizi

Vektör otoregresyon modeli, tek değişkenli AR modellerini genelleştiren bir analizdir. Çoklu zaman serileri arasındaki bağımlılıkları gösteren analiz çeşididir. Bu model değişkenlerin gecikmelerini inceler. Bu gecikme değerlerini alarak diğer tüm değişkenlerin her biri için ayrı ayrı denklem çıkarır ve bunlar ile ilgilenir. Var modelinin gösterimi Yt serisini aldığımızı varsayarsak aşağıdaki gibidir.

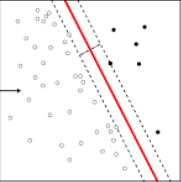


Şekil 7.17. Yt serisinin VAR gösterimi [77]

Şekil 2.1.’de görüldüğü üzere Yt serisinin VAR gösterimi yukarıdaki gibidir. Denklemde 𝜀1𝑡 olarak gösterilen değişken hata değişkenidir. X değişkenin her bir gecikmeli değeri Y değerini doğrudan etkilemekte ve değerini değiştirebilmektedir. Bu yapı ekonomi alanında sıkça kullanılan ve karşılaşılan bir analiz çeşididir. [78]

## SVR (Support Vector Regression)

Verilerin regresyon işlemleri ve sınıflandırması için kullanılan modellerden birisidir. Bu regresyon uygulanırken veri kümemizin işaretlendiği düzlem üzerinde, verilerin çoğunluklu olduğu yerin ortasından bir doğru çekilir. Doğrular arasına hesaplanan +/-E değeri doğruya eklenerek bir aralık oluşturulur. Bu aralık değeri dışında kalan değerler göz ardı edilmektedir ve hesaplamalara katılmamaktadırlar. Verilerin ortasına çekilen doğru doğrusal olmak zorunda değildir. Ancak kullanılan farklı metotlarda bu doğru eğrisel olarak da çizilebilmektedir.



Şekil 7.18. SVR örneği [79]

Şekil 7.18’da görüldüğü üzere, çekilen doğrunun üzerine +/-E değerleri eklenerek sınırlar belirlenmiştir. Belirlenen sınırlara çok uzak olan değerler görmezden gelinerek hesaplamalara katılmamışlardır.

**Sklearn kütüphanesi ile örnek SVR modeli oluşturma [80]:**

SVR(C=1.0, cache\_size=200, coef0=0.0, degree=3, epsilon=0.2, gamma='scale', kernel='rbf', max\_iter=-1, shrinking=True, tol=0.001, verbose=False)

* C: Hata parametresidir.
* cache\_size : Megabyte cinsinden çekirdek önbellek boyutunu belirtir.
* coef0: Çekirdek fonksiyonunda bağımsız terim. Sadece "poli" ve "sigmoid" lerde önemlidir.
* degree: Polinom çekirdek fonksiyonunun derecesi ("poli"). Diğer tüm çekirdekler tarafından yok sayılır.
* epsilon: Eğitme kaybı işlevinde, hiçbir epsilon mesafesi içinde gerçek değerden öngörülen noktalarla hiçbir cezanın ilişkilendirilmediği epsilon tüpünü belirtir.
* Gamma: ‘Rbf’, ‘poly’, ‘sigmoid’ için çekirdek sayısını belirtir.
* kernel: Algoritmada kullanılacak çekirdek türünü belirtir. “linear”, “poli”, “rbf”, “sigmoid”, “precomputed” veya “callable” olmalıdır. Hiçbiri verilmezse, “rbf” kullanılılır. Eğer “callable” ise, çekirdek matrisini önceden hesaplamak için kullanılır.
* max\_iter: Çözücü içindeki yinelemelerde sınır koyar. Sınırsız olması için “-1” değeri verilir.
* shrinking: Küçülen sezgisel değerin kullanılıp kullanılmayacağını belirler.
* tol: Durma kriteri için toleransı belirler.
* verbose: Ayrıntılı çıkışı etkinleştirir. Bu ayarın libsvm'deki işlem başına çalışma zamanı ayarından, etkinleştirildiğinde, çok iş parçacıklı bir bağlamda düzgün çalışmayabilir.

**Support Vector Machine (SVM) ve Support Vector Regression (SVR) Karşılaştırması**

Support Vector Machine (SVM) sınıflandırma problemi için kullanılan bir yöntemdir. Support Vector Regression (SVR) ise regresyon problemi için kullanılan bir yöntemdir. Her ikisi için de algoritmalar neredeyse aynıdır ancak farklı çıktı vermektedirler. Örneğin, Support Vector Machine (SVM) size 0 veya 1 değer verir. Buna karşılık SVR’lar sürekli değerler vermektedirler. Test verileri ise sırayla SVM ve SVR ile test edilmektedirler.

## Ridge Regresyon Analizi

Ridge regresyon çoklu bağlantılı durumlarda kullanılmaktadır. Bu yöntem kullanılacak olan değişkenlerin modele girilmesini sağlamaktadır. Ridge Regresyon modeli, En küçük katsayılar (EKK) modelinde ortaya çıkan varyans değerlerinden daha küçük varyans değerleri elde edilmesinden dolayı tercih edilebilmektedir. Ancak bazı durumlarda En küçük katsayılar(EKK) metodu kullanılması daha doğru sonuçlar verebilmektedir. Ridge regresyonda ilk adım, değişkenleri çıkartarak ve standart sapmalarına bölerek değişkenleri (hem bağımlı hem de bağımsız) standart hale getirmektir. Standarlaştırma işlemi karmaşıklığı azaltmak ve daha basit hale getirmek için kullanılır.



Şekil 7.19. . Ridge regresyon değişken katsayılar matrisi formülü [78]

Şekil 3.1.’de görülen formül ridge regresyon analizinin değişken katsayılar matrisi formülüdür. Burada ‘k’ değeri yanlılık değeri veya Ridge parametresi olarak da bilinmektedir. ‘X’ değeri açıklayıcı değişken matrisini, ‘Y’ değeri ise yanıt değişkeni vektörünü ifade etmektedir. Bu yöntem daha çok modeldeki gereksiz değerleri atarak, modeli basitleştirmek için kullanılmaktadır.

Sklearn kütüphanesinde kullanılan Ridge parametreleri ve açıklamaları:

* Alpha: Problemin tahmin oranının artmasına yardımcı olur. Bir dizi başarılı bir şekilde ilerletildiğinde, cezaların hedeflere özgü olarak çıktığı düşünülür.
* Fit\_intercept: Modeli için girilen girdiye göre işlem yapılır. True girdisi hesaplamarda kesişimler yapılacağını gösterir. False girdisi ise herhangi bir kesişim yapılmayacağını belirtmektedir.
* Normalize: Bu parametre True değeri döndürdüğünde X değerini normalleştirir. False değeri döndürüldüğünde ise parametreler görmezden gelinir.
* İntercept\_: Karar fonksiyonu için bağımsız bir terimdir. ‘fit\_intercept ’ değeri false olarak döndürüldüğünde otomatik olarak ‘0.0’ değerini almaktadır.
* Coef\_= Ağırlık vektörlerini belirtmektedir.
* N\_iter\_: Yineleme sayısını belirtmektedir. Bütün hedefleri için geçerlidir.
* Solver: ‘auto’ girdisi çözümleyiciyi otomatik olarak kendi seçmektedir. ‘svd’ girdisi Ridge katsayılarını hesaplamak için kullanılmaktadır. ‘cholesky’ girdisi kapalı formda çözüm elde etmek için kullanılır. ‘sparse\_cg’ girdisi büyük ölçekli veriler için tercih edilebilmektedir. ‘lsqr’ hızlı işlemler için tercih edilir.

# MODEL PERFORMANS ÖLÇÜTLERİ

Model oluşturma işlemi sonucunda modelin doğruluğunu ölçmemiz gerekir. Bu işlemleri gerçekleştirebilmek için literatürde tanımlı bazı standart kurallar vardır. Bu çalışmada sadece gözetimli öğrenme (bknz. Bölüm 2.2.1) için performans ölçütüne yer verilecektir.

## Hata Matrisi (Confusion Matrix)

Gözetimli öğrenme modeli performans değerlendirme ölçütü olarak en sık kullanılan yöntem Hata matrisi (kontenjans tablosu) oluşturmaktır. [34]

Tablo 8.1. Hata Matrisi (Confusion Matrix) [34]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | Gerçek Değerler | | |
| Pozitif | Negatif | Toplam |
| Tahmini Değerler | Pozitif | Doğru Pozitif (TP) | Yanlış Pozitif (FP) | tPoz |
| Negatif | Yanlış Negatif (FN) | Doğru Negatif (TN) | tNeg |
| Toplam | poz | neg | m |

Tablo 8.1’i örnek üzerinden açıklamak gerekirse bir kişinin kanser hastası olup olmadığını tahmin eden bir modelimiz olsun bu modelin doğruluğu şu şekilde ölçülür;

* **TP:** Modelin kanser olarak etiketlediği hastalardan kaç tanesi gerçekten kanser hastası.
* **FP:** Modelin kanser hastası olarak etiketlediği kişilerden kaç tanesi aslında kanser hastası değil.
* **FN:** Modelin kanser hastası değildir dediği hastalardan kaç tanesi aslında kanser hastası.
* **TN:** Modelin kanser hastası değildir olarak etiketlediği hastaların kaç tanesi gerçekten kanser hastası değil.

Bu değerlere göre modelin doğruluğunu (accuracy), hata oranını (error rate), duyarlılığını (recall), belirleyiciliğini (specificity), kesinliğini (precision) ve F-skoru’ nu bulabiliriz. Bu değerlerin nasıl hesaplanacağı aşağıda açıklanmıştır.

**Doğruluk (ACC):** Modelin doğruluk oranını vermektedir.

(8.1)

**Hata Oranı (ERR):** Modelin hata oranını vermektedir.

ERR = 1-ACC (8.2)

**Duyarlılık (Recall/TPR):**

(8.3)

**Belirleyicilik (SPC):**

(8.4)

**Kesinlik (Precision):**

(8.5)

**F-skor (F-measure):**

(8.6)

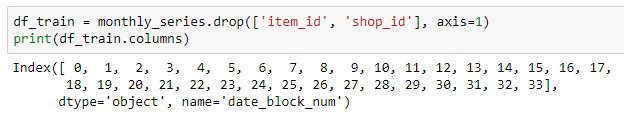
# VERİ SETİ ÜZERİNDE MODEL PERFORMANS ÖLÇÜMLERİ

Bu bölümde Bölüm 6 da hazırladığımız veri üzerinde farklı sınıflandırma algoritmaları gerçekleyerek doğruluk değerlerini farklı metrikler kullanarak hesaplanmıştır. Burada kullanılan modellerde Bölüm7 altında anlatılmıştır.

## LSTM Modeli

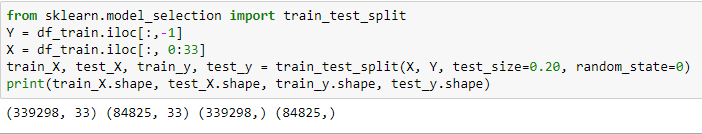
Model oluşturulurken yol haritası olarak Jason Brownlee’nin “Time Series Forecesting with LSTM” modelinden faydalanılmıştır.

Daha önce hazırladığımız veriden artık işimize yaramayacak olan ‘item\_id’ ve ‘shop\_id’ özelliklerini data framemimizden siliyoruz.



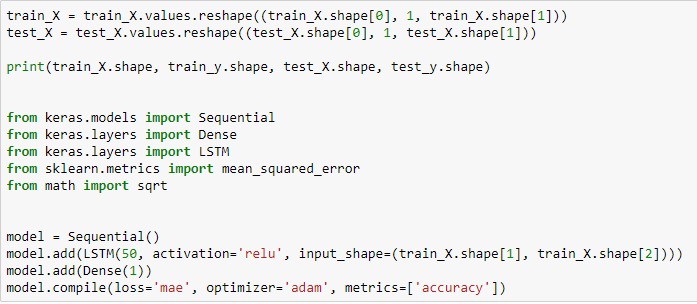
Şekil 9.1. Verisetinden gereksiz sütünları silme

Kalan ‘df\_train’ isimli verimizi modelin performansını değerlendirebilmek için (etiketi belli veriler gerektiğinden) eğitim ve test olarak bölmemiz gerekiyor. Bu işlemleri Python’da gerçekleştirmemizi sağlayan kodlar ve çıktıları Şekil 9.2’deki görselde gösterilmiştir.



Şekil 9.2. Eğitim kümesini test ve eğitim kümesi olarak ayırma

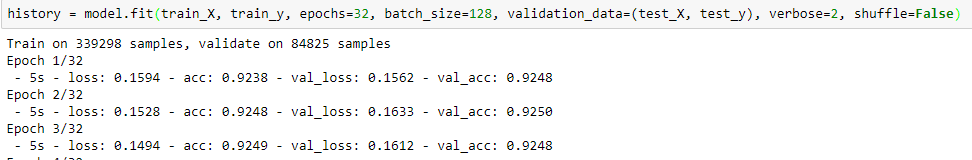
Eğitim verimizi %80 eğitim, %20 test olarak ayırdıktan sonra ayrılan veri boyutlarını Şekil 9.2 de görebiliyoruz.



Şekil 9.3. LSTM model oluşturma

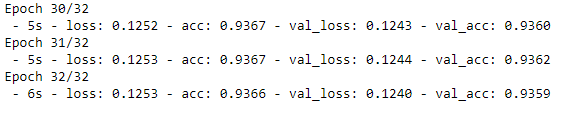
LSTM modeli oluşturabilmek için veriyi tekrar şekillendirmemiz gerekiyor (data points, time-steps, features).

Modelimizi oluştururken parametreler olarak, gizli katmanında 50 nöron, aktivasyon fonksiyonu olarak ‘relu’, toplam 3 katmandan oluşacak şekilde modellenmiştir.

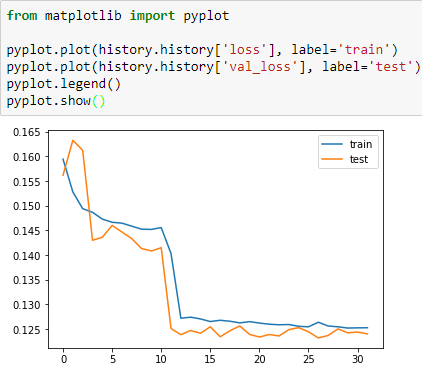


Şekil 9.4. LSTM ile veriyi eğitme

Şekil 9.4’te model fit edilmiştir. Burada ‘EPOCH’ sayısı 32 seçilmiştir. 32. EPOCH sonrası modelin loss ve doğruluk değeri Şekil 9.5’te gösterilmiştir.



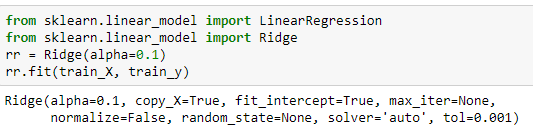
Şekil 9.5. LSTM modeli doğruluk değeri



Şekil 9.6. LSTM modei eğitim ve test kümeleri loss grafiği

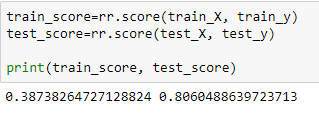
## Ridge Yöntemi

Modelimizi Ridge yöntemi ile oluşturduğunuzda sonuçlar aşağıdaki gibi olmaktadır. Bu yöntem uygulanırken [81] kaynağı yol haritası olarak seçilmiştir.



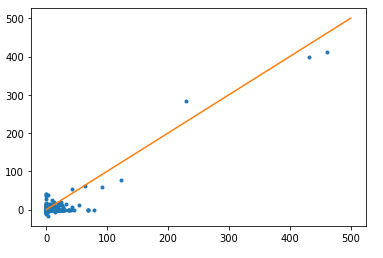
Şekil 9.7. Ridge yöntemi ile modeli fit etme

Şekil 9.7’de model Ridge yöntemine göre fit edilmiştir. Buradaki ‘train\_X’ ve ‘train\_Y’ dataframeleri yukarıda bulunan (Bölüm 9.1) LSTM modelinde oluşturulanla aynıdır.



Şekil 9.8. Ridge modeli test etme

Oluşturulan model ile Şekil 9.8’deki gibi tahmin yaptığımızda doğruluk değerleri test ve eğitim kümeleri için yukarıdaki gibi bulunmuştur. Şekil 9.9’da ise regresyon doğrusunu veri seti için çizimi gösterilmiştir.



Şekil 9.9. Ridge regreston doğrusu

# SONUÇ

Bu çalışma kapsamında 60 civarı firmanın geçmiş günlük satış verilerinden faydalanarak gelecekte yapacağı satış miktarı tahmin edilmeye çalışılmıştır. Öncelikle konu hakkında gerekli olan temel araştırmalar yapılıp tez kapsamına eklenmiştir. Daha sonra model oluşumu için ilgili literatürler taranarak yöntem seçimi yapılmaya çalışılmıştır. Literatürde birçok farklı yöntem ile model oluşturulabileceği görülmüştür. Bunlardan LSTM modeli, VAR, SVR, Ridge, Lineer Regresyon yöntemlerinden bazıları denenmiş olup Bölüm 9 altında detaylı olarak açıklanmıştır. Henüz modellenemeyenler ise yakın zamanda eklenerek tez güncellenecektir.

Şimdiye kadar çalışmada gelinen noktada yöntem araştırmaları yapılmıştır. Çalışma ortamına ve programlama diline karar verilmiştir. Bahsi geçen kavramlar hakkında bilgi sahibi olunarak çalışma için bir rota çizilmiştir. Ayrıca veri kümelerinin dağılımlarını incelenerek veri ön işleme adımlarından olan veri temizleme ve hataları giderme işlemi yapılmıştır. Veri sınıflandırma işlemine hazır hale getirilmiş ve LSTM, Ridge, VAR yöntemleri denenerek sonuçlar Bölüm 9 altında anlatılmıştır.

# KAYNAKÇA

1. Burak Işıklı, Veri Madenciliği (Data Mining) Nedir ve Nerede Kullanılır, [https://burakisikli.wordpress.com/2009/02/15/veri-madenciligidata-mining-nedir-ve-nerelerde-kullanilir-1/ (Ziyaret](https://burakisikli.wordpress.com/2009/02/15/veri-madenciligidata-mining-nedir-ve-nerelerde-kullanilir-1/%20(Ziyaret) Tarihi: 14 Ocak 2018).
2. Balaban M. Erdal, Kartal Elif, Veri Madenciliği ve Makine Öğrenmesi Temel Algoritmaları ve R Dili ile Uygulamaları, İkinci Baskı, Çağlayan Yayınevi, İstanbul, 14-23, 2018.
3. Yıldız Egemen, Makine Öğrenmesi Nedir, <https://www.technopat.net/2018/10/12/makine-ogrenmesi-nedir/> (Ziyaret Tarihi: 14 Ocak 2018)
4. Uzun Erdinç, İnternet Tabanlı Bilgi Erişimi Destekli Bir Otomatik Öğrenme Sistemi, Trakya Üniversitesi, Trakya, 2007
5. Şeker Sadi Evren, Zaman Serisi Analizi, YBS Ansiklopedisi, Cilt 2, Sayı 4, Aralık 2015
6. Sakarya Üniversitesi, <http://content.lms.sabis.sakarya.edu.tr/Uploads/49911/39167/7-14.hafta.pdf>, (Ziyaret Tarihi: 15 Ocak 2018)
7. Arıduru Oktay, Mali İstatistik Raporlama ve Analiz Şubesi, Aralık 2013
8. Eğilmez Mahfi, Konjonktür, mahfiegilmez.com, 20 Aralık 2014
9. Özcan Sami, Aykanat Zafer, Tam Yağlı Katı Yoğurt Üretiminde Süt Yağ Oranının İstatistiksel Kalite Kontrol Grafikleri ile İncelenmesi, Ardahan Üniversitesi, Karadeniz, 2017
10. Duru Özlem, Zaman Serileri Analizinde ARIMA Modelleri ve Bir Uygulama, İstanbul Üniversitesi Sosyal Bilimler Enstitüsü, İstanbul, 2007
11. Emeç Hamdi, Durağan ARIMA Modelleri, Ekonometri Semineri, Dokuz Eylül Üniversitesi, İzmir, 2014
12. Yapay Sinir Ağları Yöntemi İle Otomobil Satış Tahmini Meltem Karaatlı, Özlem Ceyda Helvacıoğlu, Nuri Ömürbek, Gönül Tokgöz Uluslararası Yönetim İktisat ve İşletme Dergisi, Cilt 8, Sayı 17, 2012
13. Yumurta tavukçuluğunda gelirin Ridge Regresyon analizi ile tahmini, Aytaç AKÇAY, Savaş SARIÖZKAN Ankara Üni. Vet Fak Derg, 62, 69-74, 2015
14. Türkiye muz üretim ve ithalatının box-jenkins ve delphi yöntemleri ile tahmini, Osman Uysal, O. Sedat Subaşı, Baran Yaşar Süleyman Demirel Üniversitesi Ziraat Fakültesi Tarım Ekonomi Bölümü
15. Yapay sinir ağları ile Türkiye elektrik enerjisi tüketiminin 2010 yılına kadar tahmini Coşkun Hamzaçebi ve Fevzi Kutay Gazi Üniv. Müh. Mim. Fak. Der. Cilt 19, No 3, 227-233, 2004
16. Regresyon analizi ve yapay sinir ağı yöntemleri ile uzun dönem yük tahmini, Fatih Aksel, İTÜ yüksek lisans tezi, yıl:2000
17. Hitit Üniversitesi Sosyal Bilimler Enstitüsü Dergisi - Yıl 11, Sayı 1, Haziran 2018
18. Türkiye’de otomobil talebinin tahmini, C. Emre Alper, Ayse Mumcu ISS/EC‐2005‐01 Boğaziçi Üniversitesi Bitirme Tezi Yıl:2005
19. Yemekhane için Yapay Zekâ Teknikleri Kullanımı ile Günlük Talep Tahmini, Fatih KILIÇ, Murat Reis AKKAYA, Nuran MEMİLİ, Avrupa Bilim ve Teknoloji Dergisi Sayı 13, S. 65-71, Ağustos 2018
20. YAPAY SİNİR AĞLARI VE DESTEK VEKTÖR MAKİNELERİ YÖNTEMLERİYLE BORSA ENDEKSİ TAHMİNİ, Emre YAKUT, Bekir ELMAS, Selahattin YAVUZ, Süleyman Demirel Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi Y.2014, C.19, S.1, s.139-157.
21. Yapay sinir ağları kullanarak konaklama işletmelerinde doluluk oranı tahmini: Türkiye’deki konaklama işletmeleri üzenine bir deneme
22. Murat ÇUHADAR-Cengiz KAYACAN Anatolia: Turizm Araştırmaları Dergisi, Cilt 16, Sayı 1, Bahar: 24-30, 2005
23. Kredi riski tahmininde yapay sinir ağları ve lojistik regresyon analizi karşılaştırılması, Hüseyin Budak, Semra ERPOLAT ,AJIT‐e: Online Academic Journal of Information Technology Fall/ 2012 – Vol: 3 Num: 9
24. İMKB şirketlerinin hisse senedi getirir başarılarının lojistik regresyon tekniği ile analizi, İlhan EGE,Ali BAYRAKDAROĞLU ZKU Journal of Social Sciences, Volume 5, Number 10, 2009, pp. 139–158
25. Türk Turizm Sektöründe Talep Tahmini Üzerine Bir Uygulama, Mehmet SOYSAL, Mine ÖMÜRGÖNÜLŞEN, Anatolia: Turizm Araştırmaları Dergisi, Cilt 21, Sayı 1, Bahar: 128-136, 2010.
26. İMKB ulusal-100 endeksinin yönünün yapay sinir ağları hata geriye yayma yöntemi ile tahmin edilmesi, Ali İhsan DİLER İMKB Dergisi Cilt:7 Sayı:25-26 Ocak-Haziran 2003.
27. Lojistik Regresyon Analizi: Tıp Verileri Üzerine Bir Uygulama, Hüdaverdi Bircan, Kocaeli Üniversitesi Sosyal Bilimler Enstitüsü Dergisi, 2004 / 2 : 185-208
28. Farklı Yöntemler Kullanılarak Geliştirilen Trafik Kaza Tahmin Modelleri ve Analizi, Ali Payıdar Akgüngör, Erdem Doğan, Int.J.Eng.Research & Development,Vol.2,No.1,January 2010
29. Bulanık mantık ve istatistiksel analiz yöntemleri ile revibrasyon uygulanmış betonlara basınç dayanımı tahmini, Serkan SUBAŞI, Ahmet BEYCİOĞLU, Mustafa ÇULLU, SDU International Journal of Technologic Sciences Vol. 2, No 3, Eylül 2010
30. YAPAY SİNİR AĞLARI İLE EN BÜYÜK YER İVMESİNİN TAHMİN EDİLMESİ, Hamza GÜLLÜ, Murat PALA, Recep İYİSAN, Altıncı Ulusal Deprem Mühendisliği Konferansı, 16-20 Ekim 2007, İstanbul
31. SANDY Robert, Statistics for Business and Economics, Mc-Graw-Hill C.USA, 1990, ss693-694
32. Bilinmiyor, Predict Future Sales, <https://www.kaggle.com/c/competitive-data-science-predict-future-sales/data> (Ziyaret Tarihi: 15.12.2018)
33. Jiawei Han, Micheline Kamber, Data Mining: Consepts and Concepts and Techniques, Illinois University, Urbana, 2006
34. Balaban M. Erdal, Kartal Elif, Veri Madenciliği ve Makine Öğrenmesi Temel Algoritmaları ve R Dili ile Uygulamaları, İkinci Baskı, Çağlayan Yayınevi, İstanbul, 30-36, 2018.
35. Balaban M. Erdal, Kartal Elif, Veri Madenciliği ve Makine Öğrenmesi Temel Algoritmaları ve R Dili ile Uygulamaları, İkinci Baskı, Çağlayan Yayınevi, İstanbul, 41-45, 2018.
36. Şeker Evren Sadi, WEKA, <http://bilgisayarkavramlari.sadievrenseker.com/2009/06/01/weka/> (Ziyaret Tarihi: 18.01.2019)
37. <https://www.cs.waikato.ac.nz/~ml/weka/gui_explorer.html> (Ziyaret Tarihi: 18.01.2019)
38. <https://tr.libreoffice.org/kesfet/calc/> (Ziyaret Tarihi: 18.01.2019)
39. <http://wiki.libreoffice.org.tr/Fonksiyonlar%C4%B1n_Tan%C4%B1m%C4%B1> (Ziyaret Tarihi: 18.01.2019)
40. <https://commons.wikimedia.org/wiki/File:LibreOffice_Calc_3.3.png> (Ziyaret Tarihi: 18.01.2019)
41. <https://teknolojiprojeleri.com/programlar/python-nedir-ne-ise-yarar-nerelerde-kullanilir> (Ziyaret Tarihi: 18.01.2019)
42. <https://teknolojiprojeleri.com/programlar/python-nedir-ne-ise-yarar-nerelerde-kullanilir> (Ziyaret Tarihi: 18.01.2019)
43. https://www.python.tc/numpy-numpy-egitim-serisi-giris/ (Ziyaret Tarihleri: 20.01.2019)
44. <https://veribilimcisi.com/2017/09/06/seaborna-hizli-bir-baslangic/> (Ziyaret Tarihleri: 20.01.2019)
45. http://www.datascience.istanbul/2017/05/21/python-pandas-ile-temel-islemler-1/ (Ziyaret Tarihleri: 20.01.2019)
46. <https://www.sistemlinux.org/2018/05/spyder-nedir-nasil-kurulur.html> (Ziyaret Tarihleri: 20.01.2019)
47. [w3.balikesir.edu.tr/~bsentuna/wp-content/uploads/2013/03/Regresyon-Analizi.pdf](http://w3.balikesir.edu.tr/~bsentuna/wp-content/uploads/2013/03/Regresyon-Analizi.pdf)(Ziyaret tarihleri: 20.01.2019)
48. <http://mustafaakca.com/regresyon-analizi-nedir/> **,**( Ziyaret tarihi:20.01.2019)
49. Şentuna Barış**,** Regresyon Analizi, Balıkesir Üniversitesi, Balıkesir, [/w3.balikesir.edu.tr/~bsentuna/wp-content/uploads/2013/03/Regresyon-Analizi.pdf](http://w3.balikesir.edu.tr/~bsentuna/wp-content/uploads/2013/03/Regresyon-Analizi.pdf) (Ziyaret tarihi:20.01.2019)
50. Murat Akyıldız, Regresyon analizi, [http://www.istatistik.gen.tr/p=83](http://www.istatistik.gen.tr/?p=83(Ziyaret) ,(Ziyaret tarihi:20.01.2019)
51. <https://veribilimcisi.com/2017/07/18/lojistik-regresyon/>, (Ziyaret tarihi: 20.01.2019)
52. Akın Çağrı Emre, Logistic Regression (Classification), <http://cagriemreakin.com/veri-bilimi/logistic-regression-classification-8.html>,( Ziyaret tarihi: 20.01.2019)
53. Bircan Hüdaverdi, Lojistik regresyon analizi: Tıp verileri üzerine bir uygulama, Kocaeli üniversitesi sosyal bilimler enstitüsü dergisi, 2004
54. Çokluk Ömay , Lojistik regresyon analizi: kavram ve uygulama, Doktora tezi, Ankara Üniversitesi, Eğitim bilimleri fakültesi, Ankara ,2010
55. Ulgen Kaan, Makine Öğrenimi Bölüm-4 (Destek Vektör Makineleri) [,https://medium.com/@k.ulgen90/makine-%C3%B6%C4%9Frenimi-b%C3%B6l%C3%BCm-4-destek-vekt%C3%B6r-makineleri-2f8010824054](mailto:,https://medium.com/@k.ulgen90/makine-%C3%B6%C4%9Frenimi-b%C3%B6l%C3%BCm-4-destek-vekt%C3%B6r-makineleri-2f8010824054) ,(Ziyaret tarihi : 20.01.2019)
56. <https://veribilimcisi.com/2017/07/19/destek-vektor-makineleri-support-vector-machine/> (Ziyaret tarihi: 20.01.2019)
57. Kabalcı Ersan, Yapay Sinir Ağları, <https://ekblc.files.wordpress.com/2014/02/ysa.pdf>, (Ziyaret tarihi: 20.01.2019)
58. Ceylan Gülden , Yapay sinir ağları ile kısa dönem yük tahmini,yüksek lisans tezi, İstanbul Teknik Üniversitesi , Fen Bilimleri Enstitüsü, İstanbul,2004
59. <https://veribilimcisi.com/2017/09/26/ileri-beslemeli-sinir-aglari-feedforward-neural-network/>[(Ziyaret](https://veribilimcisi.com/2017/09/26/ileri-beslemeli-sinir-aglari-feedforward-neural-network/(Ziyaret) tarihi : 20.01.2019)
60. Sadi Evren Seker, İleri beslemeli ağlar, <http://bilgisayarkavramlari.sadievrenseker.com/2008/11/02/ileri-beslemeli-aglar-feedforward-neural-networks/> (Ziyaret tarihi: 20.01.2019)
61. Utku Kubilay Çınar, Yapay sinir ağları ve r programıyla uygulama, <https://www.veribilimi.co/yapay-sinir-aglari/> (Ziyaret tarihi: 20.01.2019)
62. Sengoz Nilgün, Yapay sinir ağları, <http://www.derinogrenme.com/2017/03/04/yapay-sinir-aglari/> (Ziyaret tarihi : 20.01.2019)
63. Ceylan Gülcen , Yapay sinir ağları ile kısa dönem yük tahmini, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul,
64. <https://veribilimcisi.com/2017/09/26/tekrarlayan-sinir-aglari-recurrent-neural-network/>, (Ziyaret tarihi : 20.01.2019)
65. Christopher Olah, Understanding LSTM networks, <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>, (Ziyaret tarihi: 20.01.2019)
66. Ergüder Hamza, Reccurrent neural network nedir?, https://medium.com/@hamzaerguder/recurrent-neural-network-nedir-bdd3d0839120,(Ziyaret tarihi : 20.01.2019)
67. Hüdaverdi Bircan, Box-Jenkins modelleri ile aylık döviz kuru tahmini üzerine bir uygulama, Kocaeli Üniversitesi, Sosyal Bilimler Enstitüsü, Kocaeli, 2003
68. SANDY,Robert,Statistics for Business and Economics,Mc-Graw—Hill C.USA,1990,ss693-694
69. [https://docplayer.biz.tr/49924705-Cift-ustel-duzeltme-holt-metodu-ile.html, (Ziyaret](https://docplayer.biz.tr/49924705-Cift-ustel-duzeltme-holt-metodu-ile.html,%20(Ziyaret) tarihi: 19.01.2019)
70. <https://medium.com/deep-learning-turkiye/google-colab-ile-%C3%BCcretsiz-gpu-kullan%C4%B1m%C4%B1-30fdb7dd822e>, (Ziyaret tarihi: 28.05.2019)
71. Fuat Beşer, Deep Learning Development with Google Colab, Tensorflow,<https://www.kdnuggets.com/2018/02/google-colab-free-gpu-tutorial-tensorflow-keras-pytorch.html,(Ziyaret> tarihi: 28.05.2019)
72. <https://www.teknotechnic.com/gpu-nedir-tanimi-ve-gpu-hakkinda-temel-bilgiler/>, (Ziyaret tarihi: 27.05.2019)
73. Christopher Olah, Understanding LSTM Networks, <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>, (Ziyaret tarihi: 21.05.2019)
74. <https://stats.stackexchange.com/questions/276388/update-of-the-cell-state-functions-in-lstm-rnns-interplay-of-sigmoid-and-tanh>, (Ziyaret tarihi: 21.05.2019)
75. İshak Dölek, LSTM, <https://devhunteryz.wordpress.com/2018/07/14/uzun-kisa-sureli-bellek-long-short-term-memory/> (Ziyaret tarihi: 29.05.2019)
76. <https://keras.io/layers/recurrent/>, (Ziyaret tarihi: 20.05.2019)
77. Vektör Otoregres(VAR) Modeli ile İklimsel Değişkenlerin İstatiksel Analizi, Hayriye Esra Akyüz , Kırıkkale Üniversitesi Uluslararası Mühendislik Araştırma ve Geliştirme Dergisi, Haziran 2018, Cilt:10, Sayı:2
78. <http://www.wikizero.biz/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvVmVrdMO2cl9vdG9yZWdyZXN5b24>, (Ziyaret tarihi:20.05.2019)
79. http://www.datascience.istanbul/2017/06/27/support-vector-machine-svm-ile-siniflandirma-python-ornek-uygulamasi/ , (Ziyaret tarihi: 20.05.2019)
80. <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html>, (Ziyaret tarihi: 21.05.2019)
81. <https://towardsdatascience.com/ridge-and-lasso-regression-a-complete-guide-with-python-scikit-learn-e20e34bcbf0b>