

T.C.
SAKARYA ÜNİVERSİTESİ
BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ

KAPAK SAYFASI

Bu sayfanın biçimini oynamayız. Yani satır silmeyiniz, eklemeyiniz ve yazı fontu değiştirmeyiniz.

Biçimi bozduğunuz takdirde kitapçık kapağıyla uyumsuzluk sorunu olacaktır.

Böyle bir sorun ile karşılaşmamak için kapak sayfanızı danışman hocanıza ya da kırtasiyeye önceden kontrol ettirebilirsiniz.

Burayı çıktı almadan önce siliniz.

BSM 401 BİLGİSAYAR MÜHENDİSLİĞİ TASARIMI

Bilgisayar Mühendisliği Tasarımı Başlığı

Bu kısım 3 satırdan oluşmaktadır. Başlık tek satır ise diğer 2satır boş bırakılmalıdır, ancak, boş satır silinmemelidir.

Burayı çıktı almadan önce siliniz.

**DÜNYA TİCARET VERİLERİNİN MODELLENMESİ
VE TAHMİNLENMESİ**

G171210070 – Fatih GÜLŞEN

Öğrenci No, Adı, Soyadı

Bu kısım 3 satırdan oluşmaktadır. Tek öğrenci var ise diğer 2satır boş bırakılmalıdır, ancak satır silinmemelidir.

Burayı çıktı almadan önce siliniz.

Bölüm : BİLGİSAYAR MÜHENDİSLİĞİ
Danışman : Doç. Dr. Nilüfer YURTAY

2020-2021 Güz Dönemi

ÖNSÖZ

İÇİNDEKİLER

ÖNSÖZ	İİ
İÇİNDEKİLER	İİİ
SİMGELER VE KISALTMALAR LİSTESİ.....	V
ŞEKİLLER LİSTESİ	VI
TABLolar LİSTESİ	Vİİ
ÖZET.....	Vİİİ

BÖLÜM 1. GİRİŞ.....	1
1.1. Makine Öğrenmesi	1
1.1.1. Denetimli öğrenme	1
1.1.2. Denetimsiz öğrenme	2
1.1.3. Pekiştirmeli öğrenme	2
1.2. Makine Öğrenmesinin Adımları.....	2
1.2.1. Problemin tanımı	4
1.2.2. Verinin hazırlanması	4
1.2.3. Değerlendirme	4
1.2.4. Verinin özellikleri	4
1.2.5. Modelleme	4
1.2.6. Test.....	5

BÖLÜM 2. PROJEDE KULLANILAN TEKNOLOJİLER VE YÖNTEMLER ..	6
2.1. Python dili	6
2.1.1. Pandas kütüphanesi	6
2.1.2. Scikit-Learn kütüphanesi	6
2.1.3. Matplotlib kütüphanesi	7
2.2. Makine Öğrenmesi Algoritmaları	7
2.2.1. Doğrusal regresyon.....	7
2.2.2. Karar ağaçları	9
2.2.3. Rastgele orman	11
2.2.4. XGBoost	13
2.3. Değerlendirme Ölçütleri.....	14
2.3.1. Determinasyon katsayısı (R^2)	14
2.3.2. Ortalama mutlak hata (MAE)	14
2.3.3. Ortalama kare hata (MSE)	15
2.3.4. Kök ortalama kare hata (RMSE)	15
2.3.5. Ortalama mutlak yüzdesel hata (MAPE)	16

BÖLÜM 3.	GELİŞTİRİLEN YAZILIM.....	17
3.1.	Veri Toplama	17
3.2.	Programa Giriş ve Nasıl Çalışır	19
3.2.1.	Veri görselleştirme	21
3.2.2.	Veri ön işleme	21
3.2.3.	Makine öğrenmesi ve başarı hesaplaması	22
3.2.4.	Zaman hesaplama	25
3.2.5.	Program akışı.....	27
BÖLÜM 4.	SONUÇ VE DEĞERLENDİRME	27
KAYNAKLAR.....		28
EKLER.....		30
ÖZGEÇMİŞ		31
BSM 401 BİLGİSAYAR MÜHENDİSLİĞİ TASARIMI		
DEĞERLENDİRME VE SÖZLÜ SINAV TUTANAĞI.....		32

SİMGELER VE KISALTMALAR LİSTESİ

ERP	: Enterprise Resource Planning (Kurumsal Kaynak Planlaması)
LTD.ŞTİ.	: Limited Şirketi
R^2	: Determinasyon Katsayısı
MAE	: Mean absolute error (Ortalama mutlak hata)
MSE	: Mean squared error (Ortalama kare hata)
RMSE	: Root mean square error (Kök ortalama kare hata)
MAPE	: Mean absolute percentage error (Ortalama mutlak yüzde hata)
BDS	: Berkeley Software Distribution
XGBoost	: Extreme Gradient Boosting
ITC	: International Trade Centre
TXT	: Text
NaN	: Not a Number

ŞEKİLLER LİSTESİ

Şekil 1.1.1	Makine Öğrenmesi Adımları.....	3
Şekil 2.2.1.1	Doğrusal Regresyon.....	8
Şekil 2.2.1.2	Polinomal Regresyon.....	9
Şekil 2.2.2.1	Karar Ağacı Yapısı.....	10
Şekil 2.2.2.2	Karar Ağacı Örneği.....	10
Şekil 2.2.2.3	Karar Ağacı Tahmin Çizelgesi.....	11
Şekil 2.2.3.1	Rastgele Orman Yapısı.....	12
Şekil 2.2.3.2	Rastgele Orman.....	12
Şekil 2.2.4.1	XGBoost.....	13
Şekil 3.2.1	Program Genel Akış Diyagramı.....	19
Şekil 3.2.1.1	Örnek Görselleştirme Kod Parçası.....	20
Şekil 3.2.2.1	Örnek Veri Ön İşleme Kod Parçası.....	21
Şekil 3.2.3.1	Örnek Değerlendirme Kod Parçası.....	22
Şekil 3.2.3.2	Örnek Toplu Değerlendirme Fonksiyonu.....	22
Şekil 3.2.3.3	Lineer Regresyon Örnek Kod Parçası.....	23
Şekil 3.2.3.4	Örnek Model Görselleştirme ve Başarı Hesaplama.....	23
Şekil 3.2.3.5	Lineer Regresyon Skor Tablosu.....	24

TABLÖLAR LİSTESİ

Tablo 1.1.	Örnek Veri Tablosu.....	17
Tablo 3.2.4.1	Makine Öğrenmesi Süre.....	24
Tablo 3.2.4.2	Toplam Süre Değerlendirme.....	25

ÖZET

Türkçe Özet Sayfası Hakkında

“ÖZET” başlığından sonra Anahtar Kelimeler yazılmalı, yazımda bir satır aralığı kullanılmalıdır. Özet bir sayfayı aşmamalıdır. Özet’te BSM 401 tasarım çalışmasının amacı, kapsamı, kullanılan yöntem(ler) ve varılan sonuç(lar) açık ve öz olarak belirtilmelidir. En az 3 anahtar kelime kullanınız.

Burayı çıktı almadan önce siliniz.

BÖLÜM 1. GİRİŞ

Devletler, kurumlar ya da şahıslar tarafından anlamlı sonuçlar elde etmek için toplanan ticaret verileri ülkelerin ekonomik durumları hakkında bilgiler vermektedir. Şirketler ya da ülkeler ticaret politikalarını dünya çapında elde edilen veriler doğrultusunda şekillendirmektedir.

Verilere sahip olan şirketler ya da şahıslar makine öğrenmesi kullanarak gelecek yılların tahmini ticaret verileri hakkında bilgi sahibi olabilirler. Ayrıca makine öğrenmesi sayesinde şirketler ya da şahıslar belirlenen ürün ile ilgili gelecek yıllara ait ithalat ve ihracat verilerini üzerinden aksiyon alabilir ve bu doğrultuda üretim planlaması ve ticaret planlaması gibi birçok farklı alanda çalışmalar gerçekleştirebilir.

1.1. Makine Öğrenmesi

Bir problemi ve o probleme ait verileri modelleyen bilgisayar algoritmasının genel adı makine öğrenmesidir [1]. Makine öğrenmesi sayesinde büyük veriler çok daha kısa sürede analiz edilebilir. Büyük veriler arasında korelasyonlar ve gizli ilişkiler mevcut olabilir. Teknolojinin bu kadar hayatımızda olmadığı dönemlerde yapılan veri analizlerinde bu korelasyonlar ve gizli ilişkilerin ayırt edilmesi oldukça zordu ancak makine öğrenmesi sayesinde bu işlemler çok daha kısa sürelerde ve daha doğru bir biçimde gerçekleştirilebilir [2]. Makine öğrenmesinde farklı öğrenme yaklaşımları bulunmaktadır.

1.1.1. Denetimli öğrenme

Denetimli öğrenme, etiketlenmiş veri setleri üzerinde kullanılan bir makine öğrenmesi alt dalıdır. Denetimli öğrenme bir dizi giriş ve hedeften oluşmaktadır. Giriş öznitelik vektörü yani özelliklerden oluşan bir vektör iken, hedef ise çıktı olarak ne istediğimizi belirtir [2].

1.1.2. Denetimsiz öğrenme

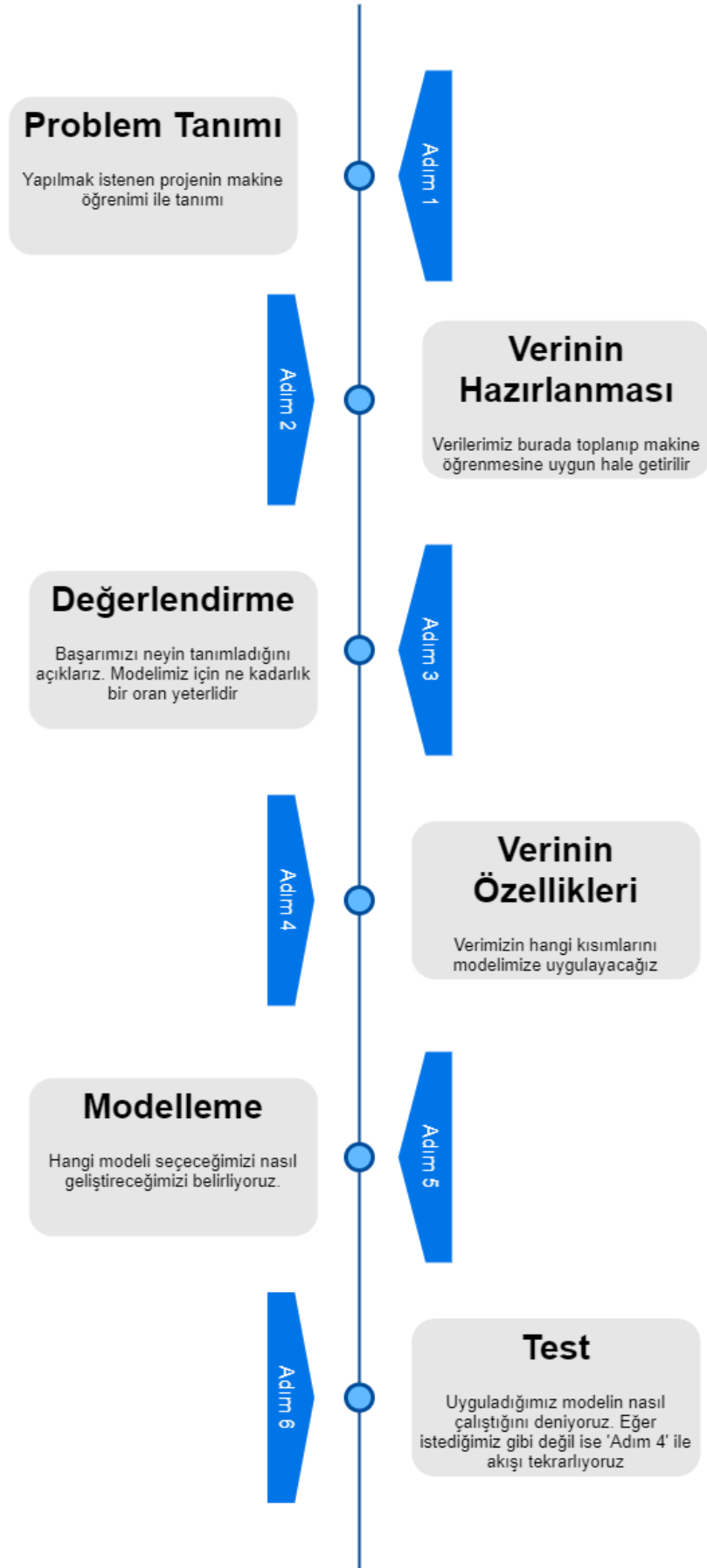
Denetimsiz öğrenme, denetimli öğrenmenin aksine hedef verileri olmadan giriş verileri arasındaki örüntüler ve ilişkiler tanımlanır. Bu sistemde hedef gösterilmeksizin sadece giriş verileri üzerinden makine öğrenmesi gerçekleştirilir [3].

1.1.3. Pekiştirmeli öğrenme

Pekiştirmeli öğrenme, diğer makine öğrenmesi yöntemlerinin aksine sadece giriş verileri verilir. Hedef verileri ise, verilmez ancak tahminlenmesi için doğru veya yanlış sinyallerinin gönderilmesi ile gerçekleşir. Bu doğru veya yanlış sinyaller gönderilmesiyle kümülatif ödül kavramı en üst düzeye çıkartılmaya uğraşılır. Böylelikle deneme yanılma yoluyla hedef verilere yaklaşılması sağlanmış olur.

1.2. Makine Öğrenmesinin Adımları

Makine öğrenmesi için literatürde yer alan belli başlı adımlar mevcuttur. Bu adımlar veri setlerine göre değişebilmektedir.



Şekil 1.1.1 Makine Öğrenmesi Adımları

1.2.1. Problemin tanımı

İş probleminin bir makine öğrenmesi problemine dökülmesi gerekmektedir. Probleme uygun makine öğrenmesi çeşidi (denetimli, denetimsiz, pekiştirmeli öğrenme) seçilmeli ve sınıflandırma, regresyon veya öneri mi yapılacağı tekrardan makine öğrenmesi problemi olarak yazılmalıdır [4], [5].

1.2.2. Verinin hazırlanması

Toplanması gereken veriler belirlenen probleme uygun olmalıdır. Eğer veriler mevcutsa bu veriler seçilen makine öğrenmesi çeşidine göre tekrar elden geçirilir. Örneğin, denetimli öğrenme için hedef değişkenlerinin tahmin edilebilmesi için giriş öznitelik vektöründen yararlanılmalıdır [4], [5].

1.2.3. Değerlendirme

İş problemi makine öğrenmesi problemi olarak tanımlanıp veriler hazırlandıktan sonra ihtiyaç olan doğruluk oranı belirlenmelidir.

Örneğin, %90 doğruluk oranı kredi ödeme durumu için uygun görülebilir ancak kanser teşhisinden kabul edilebilir sınırın altında kalabilir [4], [5].

1.2.4. Verinin özellikleri

Tüm veriler aynı özelliklere sahip değildir. Veriler kategorik özelliklere (cinsiyet vb.) ve türetilmiş verilerden (metin, resim vb.) oluşabilir. Bu veriler modelleme yapılmadan önce makine öğrenmesi için tamamen sayılara dönüştürülmelidir [4], [5].

1.2.5. Modelleme

Veriler özellikleri çıkarıldıktan sonra makine öğrenmesi modelleri (regresyon, karar ağaçları vb.) ile modellenir. Her modelin artı ve eksi yönleri bulunmaktadır. Modelleme yapılırken bunlar göz ardı edilmemelidir [4], [5].

1.2.6. Test

Oluşturulan model test edilir. Eğer bu modelin başarısı belirlenen başarı kriterinden düşük ise “verinin özellikleri” aşamasına tekrar geri dönülür ve başarı kriterine yakın değerler elde edilene kadar bu adımlar tekrarlanır [4], [5].

BÖLÜM 2. PROJEDE KULLANILAN TEKNOLOJİLER VE YÖNTEMLER

2.1. Python dili

Python, nesne yönelimli ve yüksek seviyeli bir programlama dilidir. Neredeyse bütün işletim sistemlerinde çalışabilir (Linux, Unix, Windows, Macintosh OS). Python ile veri analizi, ağ programlama, kullanıcı arabirimi programlama, web programlama, veri tabanı yazılımı gibi birçok alanda yazılım geliştirilebilir [6].

2.1.1. Pandas kütüphanesi

Pandas, python ile yazılmış bir kütüphane olup çoğunlukla veri işleme, veri analizi ve makine öğrenmesinde kullanılır. Yazılım tamamen ücretsiz ve BSD lisansına sahiptir [7].

Kütüphanenin başlıca özellikleri:

- Farklı türlerdeki verileri (Hafızada veya dosya şeklinde) okuyabilmeyi ve yazmayı sağlar
- Veri filtrelemesi yapabilir
- Veri setlerini birleştirebilir ve kırabilir
- Zaman serisi özellikleri barındırır. (Zaman aralığı oluşturabilir, tarih öteleme yapılabilir.)

2.1.2. Scikit-Learn kütüphanesi

Scikit-Learn, python ile yazılmış bir makine öğrenmesi kütüphanesidir. Kullanım alanları sınıflandırma, regresyon ve kümeleme şeklindedir. Daha hızlı çalışabilmesi için Numpy kütüphanesinden yararlanarak hesaplamalarını yapar. Birçok makine öğrenmesi algoritmasını hazır bir şekilde kendi bünyesinde barındırır [8].

2.1.3. Matplotlib kütüphanesi

Matplotlib, python ile yazılmış bir görselleştirme kütüphanesidir. Birçok şekilde bulunan matematiksel verileri görselleştirmeyi sağlar. Python ile kullanılan en yaygın görselleştirme kütüphanelerinden biridir. Oluşturulan görseller dışarı aktarılabilir bu sayede verilerin görselleri her an erişilebilir olur [9].

2.2. Makine Öğrenmesi Algoritmaları

Makine öğrenmesi algoritmaları, insanların karışık veri setlerini(kümelerini) analiz etmesine, keşfetmesine ve bunlardan anlamlı ilişkiler bulmasına yardımcı olan adımları hızlandıran belli başlı kod parçacıklarıdır. Bu kod parçacıkları istatistiksel yöntemleri kullanarak hesaplama yaparlar.

Makine öğrenesi algoritmaları, veri setinin büyük kısmını temsil eden eğitim verisi temel alarak işlemlerini optimize eder. Eğitim verisi büyüdükçe bu algoritmalar çok daha iyi sonuçlar verir.

Her algoritma farklı şekilde veriyi ele alır. En yaygın kullanılan yöntemler, kategori tahmini, aykırı verilerin tespiti ve değer tahmini gibidir [10].

2.2.1. Doğrusal regresyon

Doğrusal regresyon, iki değişken arasındaki ilişkiyi saptamaya çalışan bir istatistiksel yöntemdir. Amacı bir diziye en uygun düz çizgiyi (regresyon çizgisi) çizmektir.

Bağımlı değişken (y) ile bağımsız değişkenler (x) arasında ilişki kurar [11].

Denklemini şu şekildedir:

$$y = \beta_0 + \beta_1 x + \epsilon \quad (2.2.1.1)$$

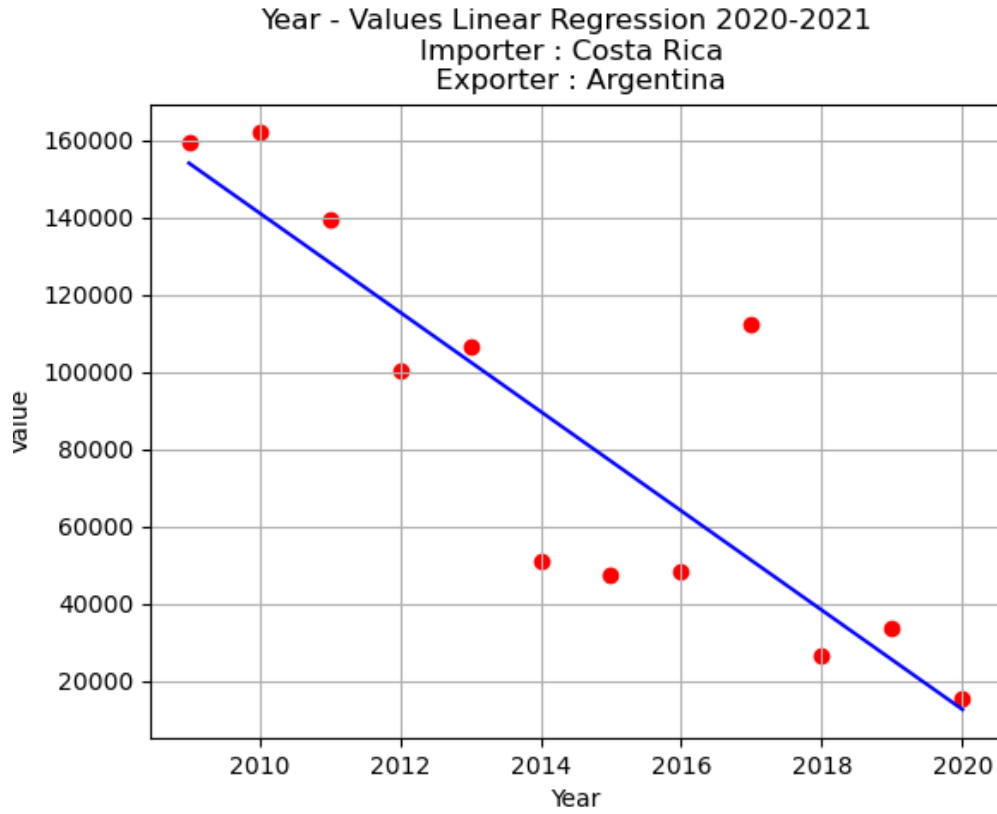
y: Bağımlı değişken

x: Bağımsız değişken

β_0 : Doğrunun y-eksenini kestiği nokta ve regresyon sabiti

β_1 : Doğrunun eğimi ve regresyon katsayısı

ϵ : Rastgele hata değeri



Şekil 2.2.1.1 Doğrusal Regresyon

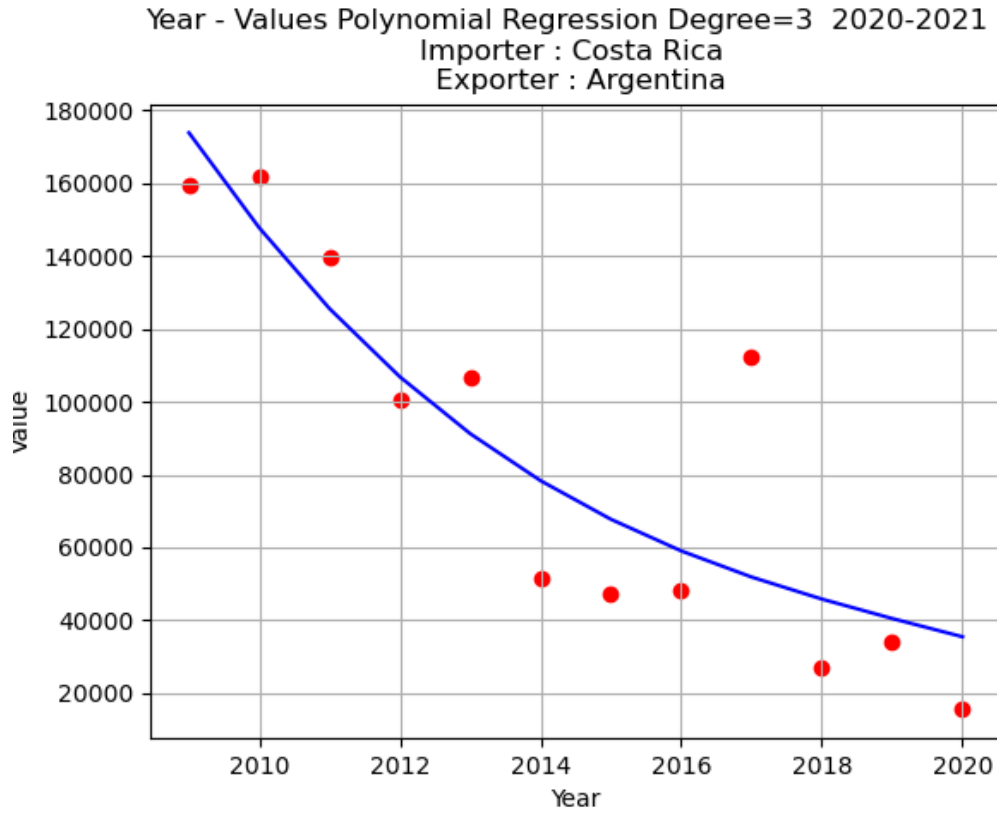
Şekil 2.2.1.1 de görünen kırmızı noktalar gerçek veriler, mavi çizgi regresyon çizgisidir. Verilere en uygun olan regresyon çizgisi çizilmiştir.

Eğer veriler düz bir çizgi ile temsil edilemiyorsa polinomal regresyon kullanılması önerilir.

Polinomal denklem şu şekildedir:

$$y = \beta_0 + \beta_1 x^1 + \beta_2 x^2 + \beta_3 x^3 + \dots + \beta_n x^n + \epsilon \quad (2.2.1.2)$$

Bu denklemde ‘n’ polinomal derecesi olarak adlandırılır. β_0 , β_1 gibi regresyon sabitleri doğrusal olduğundan bu denklemde doğrusal regresyon içerisine girmektedir [11].



Şekil 2.2.1.2 Polinomal Regresyon

Şekil 2.2.1.2 de görünen mavi çizgi 3. dereceden bir polinomal regresyona aittir.

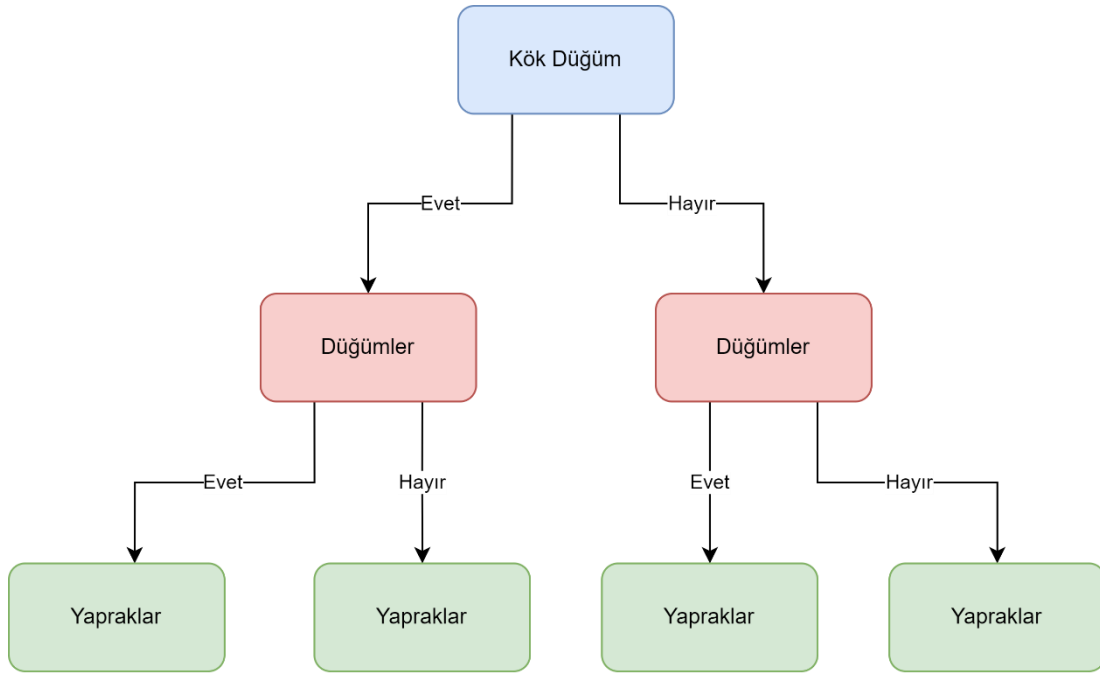
2.2.2. Karar ağaçları

Karar ağaçları, sınıflandırma ve tahminleme için sıkça kullanılan bir yöntemdir. En çok tercih edilme nedeni kolay anlaşılabilir ve yorumlanabilir olmasıdır. Bir adet kök düğümden oluşur ve alt düğümlere ayrılır. En alt düğümlere yaprak denir.

Ağaç tek bir düğüm ile başlar dallar oluşurken eğer seçilen düğüm aynı sınıfa ait ise yaprak oluşur [12]. Şekil 2.2.2.1 de ağacın yapısı gösterilmiştir.

İki farklı sınıfa ayrılır:

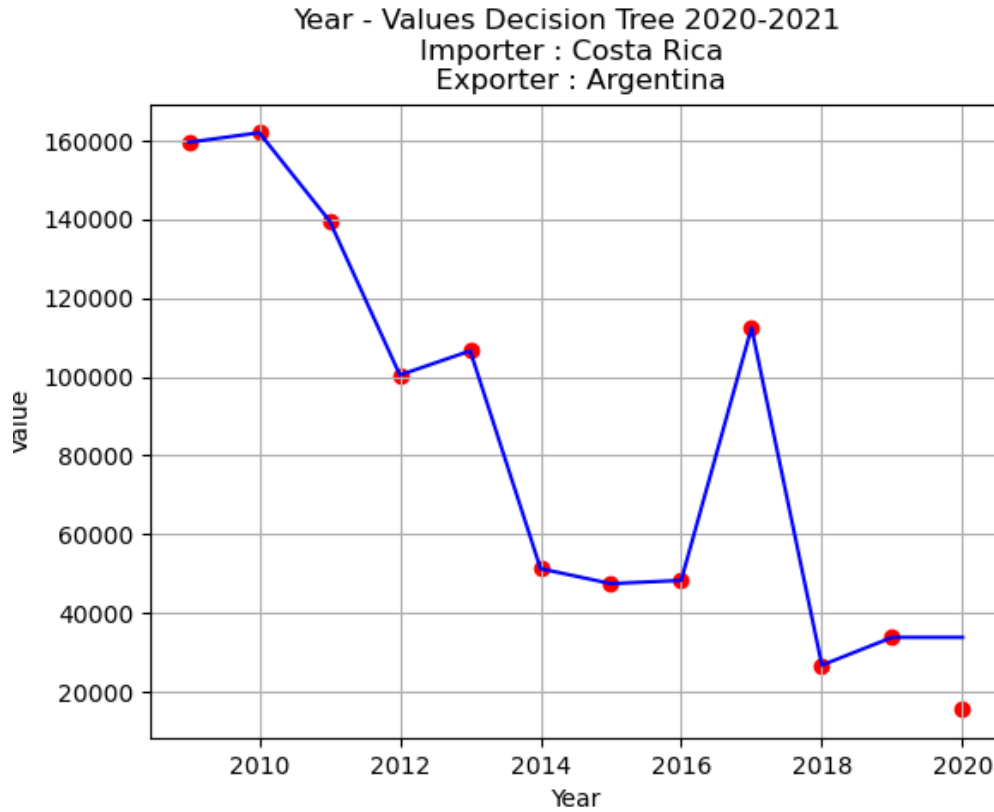
- Sınıflandırma Ağaçları: Hedef kümeyi ayrılan sınıflandır birine yerleştirmeye çalışır [13].
- Regresyon Ağaçları: Sonucu bir sınıf yerine sayısal bir değer olur. Şekil 2.2.2.1 de gösterilen bir regresyon ağacı yapısıdır [13].



Şekil 2.2.2.1 Karar Ağacı Yapısı



Şekil 2.2.2.2 Karar Ağacı Örneği

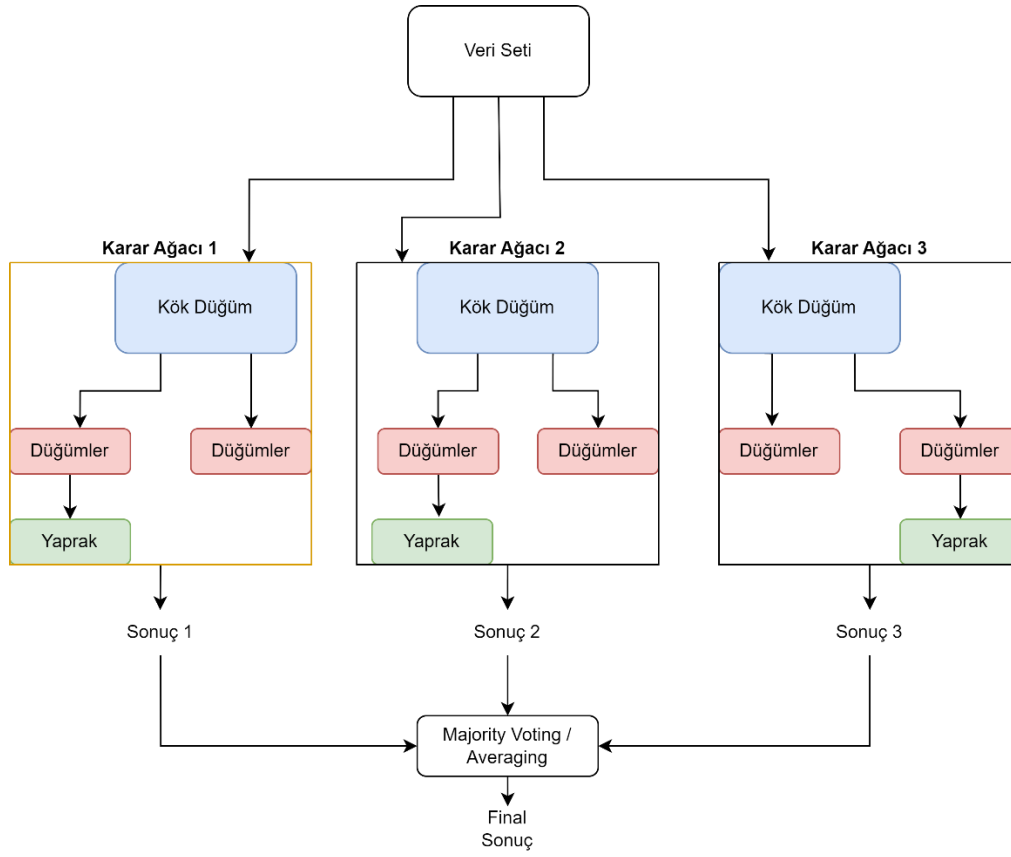


Şekil 2.2.2.3 Karar Ağaçları Tahmin Çizelgesi

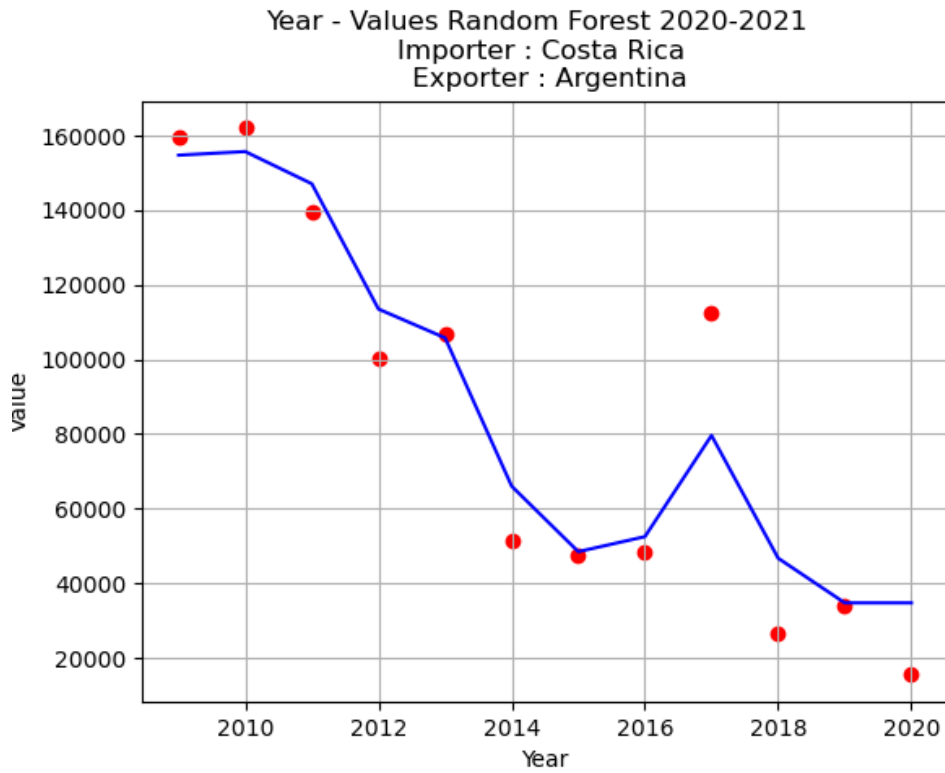
Şekil 2.2.2.2 de gösterilen ağacın Şekil 2.2.2.3 de sonuç tablosu gösterilmiştir. 2020 verilerini hiç görmediği için 2019 yılı ile aynı tahminde bulunmuştur. Hiç görmediği verilerde özellikle veri sayısı az ise çok başarılı değildir. Sınıflandırma problemlerine daha yatkın olduğu söylenebilir bu tablolar incelendiğinde.

2.2.3. Rastgele orman

Rastgele orman, birden fazla karar ağacı ile çalışır. Bunları birleştirerek daha istikrarlı ve daha doğru tahminler yapmayı hedefler. Hem sınıflandırma hem de regresyon problemlerinde kullanılabilir. Eğitim süresi çok uzun sürmez ancak tahminleme süresi karar ağaçlarına göre uzun sürer. Bunun nedeni birden fazla ağaç olması ve bunları bir takım işleme sokmasından dolayı tahminleme süresi karar ağaçlarına göre uzun sürmektedir [14]. Şekil 2.2.3.1 de örnek bir rastgele orman yapısı gösterilmiştir.



Şekil 2.2.3.1 Rastgele Orman Yapısı

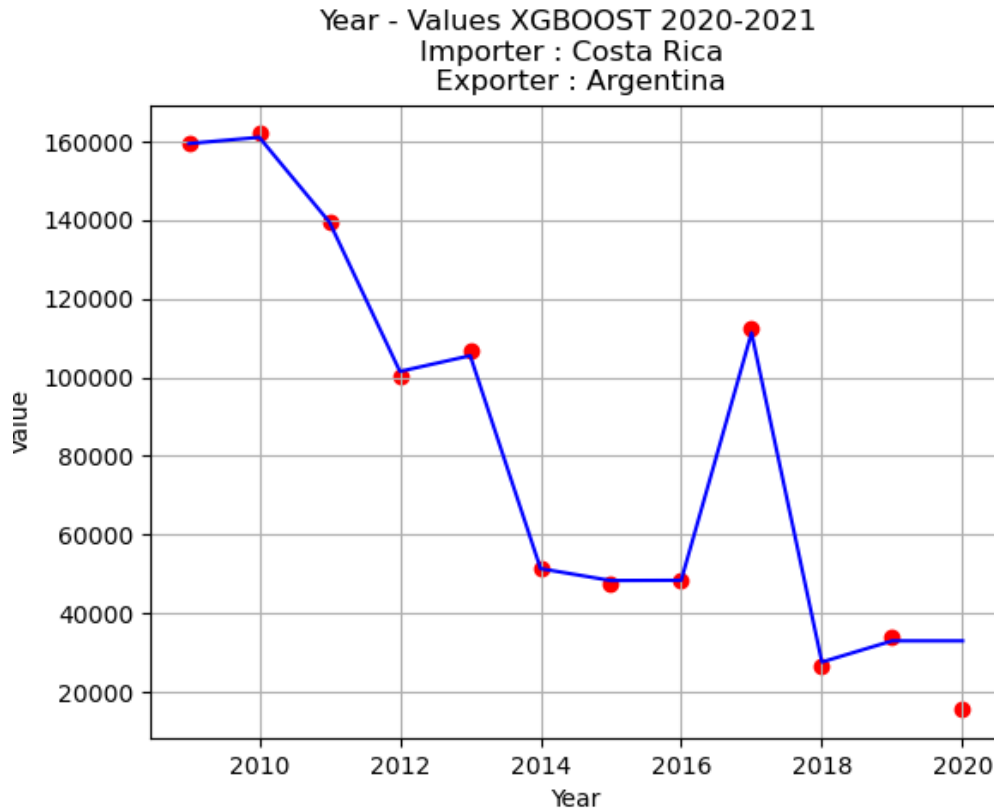


Şekil 2.2.3.2 Rastgele Orman

Şekil 2.2.3.2 de görüldüğü gibi 2020 verilerinde karar ağaçları ile aynı şekilde önceki yıl yani 2019 ile aynı değeri almıştır. Ancak önceki yıllarda karar ağaçları gibi nokta atışı yapmak yerine daha ortalama değerler dönmüştür.

2.2.4. XGBoost

XGBoost, karar ağaçlarını tahminleyicileri olarak kullanan bir gradyan artırma (gradyan artırma birkaç modelden gelen tahminleri bir modelde birleştirme yoludur) algoritmasıdır. Bunun yanı sıra hız ve performans için özel olarak tasarlanmış bir algoritmadır [15].



Şekiller 2.2.4.1 XGBoost

Şekil 2.2.4.1 de gösterildiği gibi XGBoost algoritması karar ağaçları ve gradyan artırma yöntemleri ile çalıştığı için hiç eğitimde kullanılmayan verilerde eğitimde kullanılan en yakın veriyi tercih etmiştir. Bu grafikte de bu 2019 yılını 2020 yılı tahmini olarak ele almıştır.

2.3. Değerlendirme Ölçütleri

Regresyon analizi yaparken veya tahminleme yaparken istatistikçiler tarafından birkaç değerlendirme ölçütü geliştirilmiştir. En sık kullanılan birkaç değerlendirme ölçütü sırasıyla verilmiştir.

2.3.1. Determinasyon katsayısı (R^2)

Determinasyon katsayısı, deneysel verilerin doğrusal veriye ne kadar uyumlu olduğunun hesaplanmasıdır. R^2 0 ile 1 arasında değerler alır. R^2 değerinin 1 eşit olması deneysel verilerin doğrusal veriye tamamen uyumlu olduğunu gösterir. Örneğin $R^2 = 0.65$ ise verilerin %65 uyumlu iken geri kalan %35’lik kısma doğruya uymadığını gösterir. Ne kadar çok veri var ise R^2 ’nin gösterdiği değere güven o kadar artar. R^2 değeri, açıklanabilen varyasyonun toplam varyasyona oranı olarak tanımlanır. En küçük kareler yöntemi ile hesaplanabilir [16]. Pay atıkların kareler toplamı olarak adlandırılırken, payda ortalamaya uzaklığın kareler toplamı olarak adlandırılır.

$$R^2 = 1 - \frac{\sum_{i=0}^n (\gamma_i - \hat{\gamma}_i)^2}{\sum_{i=0}^n (\gamma_i - \tilde{\gamma})^2} \quad (2.3.1)$$

γ_i : Gerçek değer

$\hat{\gamma}_i$: Tahmin edilen değer

$\tilde{\gamma}$: Ortalama değer

2.3.2. Ortalama mutlak hata (MAE)

Mutlak hata, her tahmin edilen değer ile gerçek değer arasındaki fark ile bulunur. Ortalama mutlak hata ise mutlak hataların toplanması ile ortaya çıkan değeri toplam gözlem sayısına bölünmesi sonucu çıkan değerdir. MAE değerimiz 0’a ne kadar yakın ise gerçek sonuca o kadar yakın olduğumuzu temsil eder.

Değer 0'dan uzaklaştıkça hata değerimiz artmaktadır. Hata değerleri katlanmadığından gerçek verilerden ne kadar uzaklaşılsa uzaklaşılsın eşit şekilde ceza uygular [17], [18].

$$MAE = \frac{1}{n} \sum_{i=0}^n |\hat{\gamma}_i - \gamma_i| \quad (2.3.2)$$

$\hat{\gamma}_i$: Tahmin edilen değer

γ_i : Gerçek değer

n : Toplam gözlem sayısı

2.3.3. Ortalama kare hata (MSE)

Ortalama kare hata, her bir tahminlenen veri noktası için karesel hata değerinin ortalaması ile bulunur. Değerlerin karesi alındığından dolayı değerler daima 0'dan büyük olarak bulunur. Çıkan hatanın karesi alındığı için aradaki hata değeri büyüdükçe ceza katlanarak artar. MSE değeri MAE gibi 0 değerine ne kadar yakın olursa tahminlerin gerçek değerlere o kadar yakın olduğunu belirtir [17], [18].

$$MSE = \frac{1}{n} \sum_{i=0}^n (\gamma_i - \hat{\gamma}_i)^2 \quad (2.3.3)$$

$\hat{\gamma}_i$: Tahmin edilen değer

γ_i : Gerçek değer

n : Toplam gözlem sayısı

2.3.4. Kök ortalama kare hata (RMSE)

Kök ortalama kare hata, adından da anlaşılacağı gibi MSE değerinin karekökü alınarak hesaplanır. RMSE ve MSE, birkaç aykırı verinin bulunduğu veri kümelerinde kullanması tercih edilen metriklerdir [17], [18].

$$RMSE = \sqrt{MSE} \quad (2.3.4)$$

2.3.5. Ortalama mutlak yüzdesel hata (MAPE)

Ortalama mutlak yüzdesel hata, regresyon analizi ve zaman serilerinde genellikle kullanılan bir yöntemdir. Gerçek değerler arasında 0 değerleri mevcut ise bu yöntem kullanılamaz. MAPE değeri ne kadar düşük ise tahminlenen değerler gerçek değerlere o yakın olduğu belirtir. Oluşturulan model için bir başarı kriteri olarak belirlenebilir. Örneğin MAPE değeri %10 ise çok başarı %20 ise yeterli olarak sınıflandırabiliriz [18], [19].

$$MAPE = \frac{1}{n} \sum_{i=0}^n \left| \frac{\gamma_i - \hat{\gamma}_i}{\gamma_i} \right| \quad (2.3.5)$$

$\hat{\gamma}_i$: Tahmin edilen değer

γ_i : Gerçek değer

n : Toplam gözlem sayısı

BÖLÜM 3. GELİŞTİRİLEN YAZILIM

Bu tasarım çalışması için hazırlanan yazılımda belli başlı adımlar gerçekleştirilmiştir.

Bu adımlar şu şekildedir:

- Veri setinin hazırlanması
- Kullanılacak programlama dilinin belirlenmesi
- Veri seti hakkında ön bilgilendirmelerin yapılması ve incelenmesi
- Veri setinden örneklem alınıp makine öğrenmesi algoritmalarının uygulanması
- Makine öğrenmesi algoritmalarının parametrelerinin iyileştirilmesi
- Bütün veri setine iyileştirilmiş parametreler ile makine öğrenmesinin uygulanması
- Tahminlenen verilerin değerlendirme ölçütleri ile değerlendirilmesi ve tahmin değerlerinin çıktı alınması

3.1. Veri Toplama

Yazılım geliştirmesi başlanmadan önce veri setlerini hazırlamak için ITC kurumuna ait Trademap'ten ücretsiz şekilde veriler edinilmiştir [20]. Bu kuruluştan veri edinmeden önce birkaç ülke belirlenmiştir. Bu ülkeler sırayla Türkiye, Amerika Birleşik Devletleri, Çin, Arjantin, Almanya ve Senegal'dir. Ülke belirlenmesinin sebebi verileri ülke bazlı sunuyor olmalarıdır.

Bu ülkelere ait ithalatçı ve ihracatçı ülkelerin ticaret hacimleri ayrıyeten ithalat ve ihracat yapılan ürünlerin (GTİP bazında) ticaret hacimleri edinilmiştir. Bu veriler tamamen ayrı olarak sunulduğundan bütün ülke verileri tek veri setinde birleştirilmiştir. Bir veri setine ait örneklem Tablo 1.1 de gösterildiği gibidir.

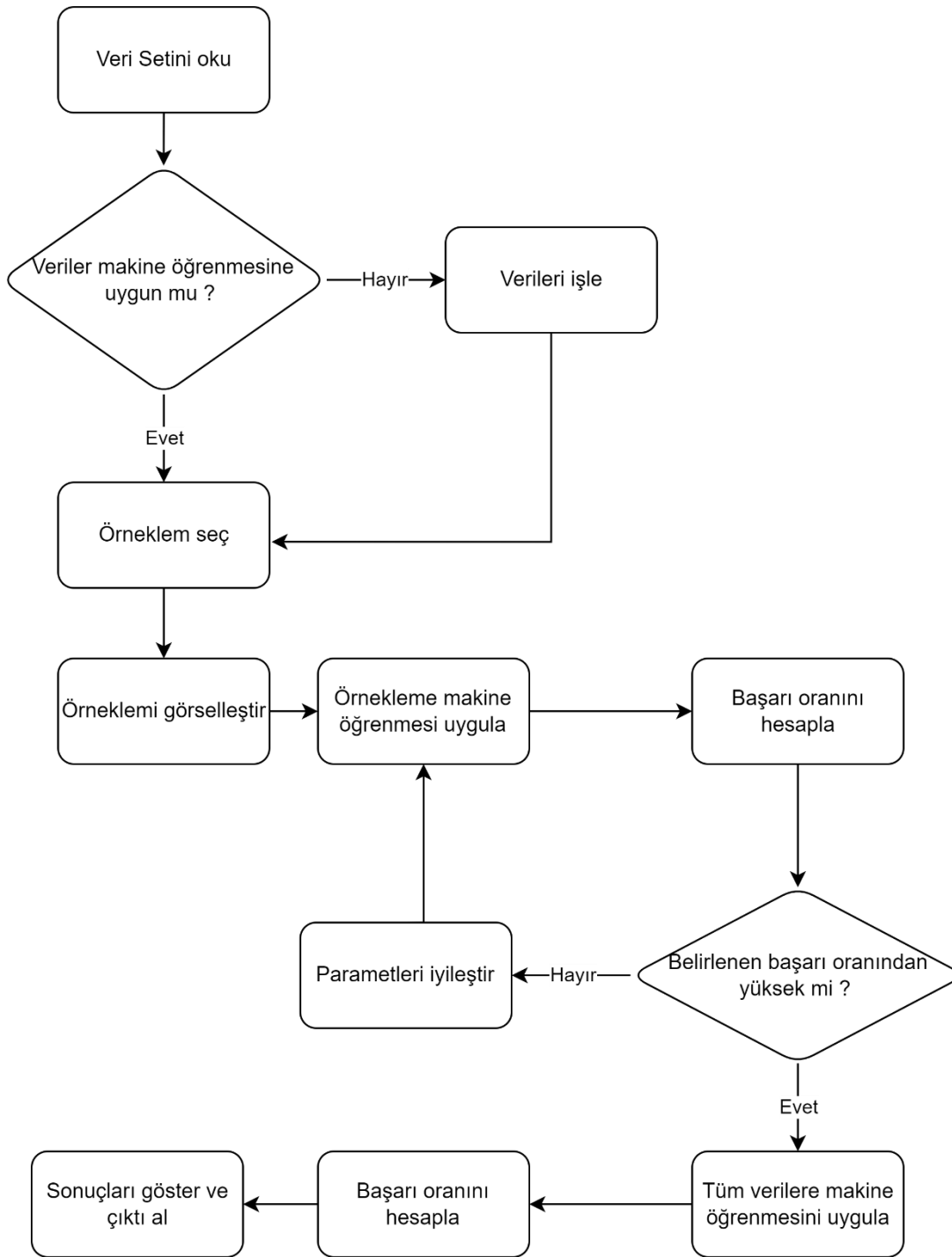
Tablo 1.1 Örnek Veri Tablosu

Exporter	Argentina	China	Germany	Senegal	Turkey	USA
Importers	Netherlands	Mexico	Denmark	Japan	Poland	United Kingdom
Exported value in 2009	2392143	17872653	18334498	3965	1504280	48328089
Exported value in 2010	2369770	23975906	20339168	10395	1758252	56009021
Exported value in 2011	2548736	27517963	19075405	10574	1853700	54853629
Exported value in 2012	2204215	28966300	20732965	4206	2137802	47339021
Exported value in 2013	1912958	32255386	21514739	6406	2484489	53808147
Exported value in 2015	1205855	32447349	19206751	7138	2651896	55197491
Exported value in 2016	1170977	35954952	20691180	5535	3070758	56262428
Exported value in 2017	1374942	44129450	23572859	12321	3346149	66454485
Exported value in 2018	1523674	46377886	21488112	16321	3449492	69066568
Exported value in 2019	1606701	44850418	21070133	10733	3474883	58429779
Exported value in 2020	1209206	12298991	17846943	3182	1322218	45707871

3.2. Programa Giriş ve Nasıl Çalışır

Veri setinin tahminlenmesi ve görselleştirilmesi için tamamen Python programlama dili kullanılmıştır. Ham veriler 'txt' formatındadır ve '\t' ile biçimlendirilmiştir. Bu şekilde olmasından dolayı Python programlama dilinin Pandas kütüphanesi ile kolayca bu veriler okunabilmektedir. Dosya sayısını azaltmak için bütün ham veriler aynı türde olacak şekilde birleştirilmiştir. Bunun nedeni bütün veriye aynı anda makine öğrenmesini uygulanmak istenmesindendir.

Şekil 3.2.1'de programın genel olarak tasarımının nasıl yapıldığına dair bilgi verilmiştir. Şekil 3.2.1'de gösterildiği gibi veriler hazırlanıp ilk ön işleme girer. Ön işlemde kasıt sütun isimlerinin düzenlenmesi, aykırı verilerin atılması ve bilinmeyen (NaN) değerlerin silinmesi veya o değerlerin belli başlı işlemlerle doldurulması. Bu yazılımda kullanılan verilerde bilinmeyen değer bulundugu satır tamamen silinmiştir. Bunun nedeni satır sayısı ne kadar fazla olsa da sütun yani yıl sayısı az olduğundan veri eksikliği tahmin değerlerinde fazlasıyla sapmaya neden olabilir. Ön işlemler yapıldıktan sonra bir örneklem seçilir. Bunun nedeni ise sonuçları hızlı görmek ve görselleştirme yapılırken daha kolay şekilde grafiklerin okunabilmesinin sağlanması. Görselleştirme verileri daha iyi anlaşılmasını sağlar. Veri ne kadar iyi çözümlenir ise makine öğrenmesi kullanılırken o kadar iyi sonuçlar alınır. Görselleştirme bu yüzden en önemli adımlardan biridir. En son makine öğrenmesi uygulanır. Örneklem üzerinde en iyi parametreler elde edilene kadar tekrar tekrar veriyi makine öğrenmesi uygulanır. Örneklem seçilmesinin nedeni buradadır. Makine öğrenmesi (özellikle yüksek miktarda veri setlerinde) uzun süreler aldığından bütün veride tekrar tekrar denenmesi vakit kaybına neden olmaktadır. Uygun parametreler bulunduğu taktirde bütün veriye makine öğrenmesi uygulanabilir. Buradaki parametrelerden kasıt polinomal regresyonun derecesi, karar ağaçlarının derinlikleri gibi değişkenlerdir. Başarı değerlendirmeleri Başlık 2.3'te belirtilen matematiksel formüller ile hesaplanır. Oradaki formüllerin kullanılma nedeni verileri sırasıyla kontrol etmeden çıkan sonuca göre parametrelerin daha kolay ayarlanmasıdır. Başarı ölçütleri genellikle Başlık 2.3'te belirtilen formüller ile belirlenir.



Şekil 3.2.1 Program Genel Akış Diyagramı

3.2.1. Veri görselleştirme

Python ile veri görselleştirme yaparken genellikle Matplotlib ve Seaborn gibi kütüphanelere başvurulur. Bunun nedeni birkaç kod bloğu ile hazır şablonlar üzerinden kolaylıkla istenilen görselin elde edilmesindendir.

```
def bar_plot(df_catplot: pd.DataFrame, country: str, value: int, _x: str, _y: str, _col: str):
    chart = sns.catplot(x=_x, y=_y, col=_col,
                        data=df_catplot.loc[
                            ((df_catplot[_col] == country) & (df_catplot[_y] > value)
                            )], kind='bar', legend=True, legend_out=True)
    chart.set_xticklabels(rotation=90, horizontalalignment='center', fontsize=7)
    plt.tight_layout()
    plt.show()

def pie_plot(_all_data: pd.DataFrame, _x: str, _y: str, _title: str):
    colors = sns.color_palette('pastel')
    _all_data = _all_data.groupby([_x]).sum()
    _all_data.plot(kind='pie', y=_y, x=_x, autopct='%1.2f%%', colors=colors,
                  startangle=270, fontsize=17, shadow=True, explode=(0.1, 0.1, 0.1, 0.1, 0.1, 0.1),
                  legend=False,
                  title=_title)
    plt.tight_layout()
    plt.show()
```

Şekil 3.2.1.1 Örnek Görselleştirme Kod Parçası

Şekil 3.2.1.1 de görüldüğü gibi kısa bir fonksiyon yazılarak hazır kütüphaneden yararlanarak görselleştirme yapılabilir.

‘bar_plot’ fonksiyonunda ‘Seaborn’ kütüphanesinin ‘catplot’ fonksiyonundan yararlanılmışken ‘pie_plot’ fonksiyonunda ‘Pandas’ kütüphanesinin kendi içinde bulunan ‘plot’ fonksiyonundan yararlanılmıştır.

3.2.2. Veri ön işleme

Python ile veri ön işleme yapılırken genellikle yazılımcılar kendi fonksiyonları yazarlar. Bunun nedeni verilere göre bazı kısımların değişmesidir. Hazır kütüphane ile yapılan kısımları da bulunmaktadır. ‘sci-kit learn’ ve ‘Pandas’ gibi kütüphanelerden yararlanılabilir. Bu ön işlemenin başlıca nedeni makine öğrenmesi uygulanmadan önce belli başlı işlemlerin yapılması gerekir.

Örneğin regresyon analizine metin değeri giriş yapılamaz giriş değerleri tamamen sayısal olmalıdır. Metin değerine örnek olarak cinsiyet verilebilir. Cinsiyet erkek, kadın olarak kullanılması yerine 0 ve 1 kullanılır. Bu literatüre ‘one-hot encoder’ olarak geçmiştir.

Bunun gibi işlemleri uygularken dikkatli olunmalıdır. Bu işlemlerin en çok karşılaşılan sorunu kukla değişken tuzağıdır. Kukla değişken tuzağına örnek olarak, 4 mevsim için 3 adet değişken olması yeterlidir. Eğer 4 değişken kullanılır ise 1 adet değişken kukla değişken tuzağına düşmüş demektir. Şekil 3.2.2.1 de örnek birkaç fonksiyona yer verilmiştir.

```
def one_hot_encoder(dataframe, categorical_cols, drop_first=False):
    dataframe = pd.get_dummies(dataframe, columns=categorical_cols, drop_first=drop_first)
    return dataframe

def label_encoder(dataframe, binary_col):
    labelencoder = preprocessing.LabelEncoder()
    dataframe[binary_col] = labelencoder.fit_transform(dataframe[binary_col])
    return dataframe

def outlier_thresholds(dataframe, col_name):
    quartile1 = dataframe[col_name].quantile(0.01)
    quartile3 = dataframe[col_name].quantile(0.99)
    interquartile_range = quartile3 - quartile1
    up_limit = quartile3 + 1.5 * interquartile_range
    low_limit = quartile1 - 1.5 * interquartile_range
    return low_limit, up_limit

def grab_outliers(dataframe, col_name, index=False):
    low, up = outlier_thresholds(dataframe, col_name)
    if dataframe[((dataframe[col_name] < low) | (dataframe[col_name] > up)).shape[0] > 10:
        print(dataframe[((dataframe[col_name] < low) | (dataframe[col_name] > up)).head())
    else:
        print(dataframe[((dataframe[col_name] < low) | (dataframe[col_name] > up))])

    if index:
        outlier_index = dataframe[((dataframe[col_name] < low) | (dataframe[col_name] > up)).index
        return outlier_index
```

Şekil 3.2.2.1 Örnek Veri Ön İşleme Kod Parçası

3.2.3. Makine öğrenmesi ve başarı hesaplaması

Makine öğrenmesi kısmında en yaygın olarak genellikle ‘sci-kit Learn’ kütüphanesi kullanılır. Burada genellikle hazır kütüphane kullanılmasının nedeni uzman seviyede matematik, istatistik ve mühendislik gerektiren bir alanın kısa kod parçaları ile zamandan ve iş gücünden tasarruf edilmesini sağlar.

Bu makine öğrenmesi algoritmaları Bölüm 2.3'te bahsedilen matematiksel formüller ile kolaylıkla değerlendirilebilir. Bu formüller bazıları hazır olarak sunulurken bazıları yazılımcı tarafından yazılması gerebilir.

```
def SMAPE(preds, target):
    n = len(preds)
    masked_arr = ~((preds == 0) & (target == 0))
    preds, target = preds[masked_arr], target[masked_arr]
    num = np.abs(preds - target)
    denom = np.abs(preds) + np.abs(target)
    smape_val = (200 * np.sum(num / denom)) / n
    return smape_val

def MAPE(Y_actual, Y_Predicted):
    mape = np.mean(np.abs((Y_actual - Y_Predicted) / Y_actual)) * 100
    return mape
```

Şekil 3.2.3.1 Örnek Değerlendirme Kod Parçası

```
def metrics_score(text, predictions, y_values):
    from sklearn.metrics import r2_score
    from sklearn.metrics import mean_squared_error
    from sklearn.metrics import mean_absolute_error
    print('\n----- Score Table -----')
    print(f'{text} R^2 Score: %.3f' % r2_score(y_values, predictions))
    print(f'{text} Root Mean Square Error (RMSE): %.3f' % np.sqrt(mean_squared_error(y_values, predictions)))
    print(f'{text} Mean Squared Error (MSE): %.3f' % mean_squared_error(y_values, predictions))
    print(f'{text} Mean Absolute Error (MAE): %.3f' % mean_absolute_error(y_values, predictions))
    print(f'{text} Mean Absolute Percentage Error (MAPE): %.3f' % MAPE(y_values, predictions))
    print(f'{text} Symmetric Mean Absolute Percentage Error (SMAPE): %.3f' % SMAPE(y_values, predictions))
    print('-----\n')
```

Şekil 3.2.3.2 Örnek Toplu Değerlendirme Fonksiyonu

Şekil 3.2.3.1'de gösterildiği gibi yazılımcı tarafından yazılmış 2 adet değerlendirme ölçütü olarak kullanılacak fonksiyon bulunmaktadır.

Şekil 3.2.3.2'de gösterilen fonksiyonda bütün değerlendirme metotları ile çıktı alınması sağlanmıştır. Bunun nedeni bütün metotların tek bir anda görüntülenip hep daha iyi anlaşılmasının sağlanması hem de başarı kriterinin daha kolay belirlenmesini sağlamaktır. Ayrıca Şekil 3.2.3.2'de görünen 4 adet metot 'sci-kit Learn' kütüphanesinden hazır olarak kullanılmakta iken 2 adet metot Şekil 3.2.3.1'de gösterilmiştir.

```
# Linear Regression
lin_reg = LinearRegression(n_jobs=-1)
lin_reg.fit(X_values, Y_values)

predictedLinear = lin_reg.predict(X_values)
models_sum(row, predictedLinear, X_values, Y_values, 'Linear Regression ', plot_metric)

predictedLinear2020 = lin_reg.predict(x_2020)
models_sum(row, predictedLinear2020, x_2020, y_2020, 'Linear Regression 2020-2021 ', plot_metric)
```

Şekil 3.2.3.3 Lineer Regresyon Örnek Kod Parçası

Şekil 3.2.3.3'te gösterilen kod parçasında bir makine öğrenmesi algoritmasının eğitimi ve tahminlenmesi sağlanmıştır. Lineer regresyon ve diğer algoritmalar uygulanır iken daha iyi hiper-parametreler kullanılarak doğruluk oranı artırılabilir.

```
def algo_scatter(row, predictions, x_values, y_values, text=''):
    plt.scatter(x_values, y_values, color='red')
    plt.grid(True)
    plt.plot(x_values, predictions, color='blue')
    plt.title(f'Year - Values {text} \nImporter : {row["Importer"]} \n Exporter : {row["Exporter"]}')
    plt.xlabel('Year')
    plt.ylabel('Value')
    plt.show()

def models_sum(row, predictions, x_values, y_values, text='', plot_metric=[True, True]):
    if plot_metric[0]:
        algo_scatter(row, predictions, x_values, y_values, text)
    if plot_metric[1]:
        metrics_score(text, predictions, y_values)
```

Şekil 3.2.3.4 Örnek Model Görselleştirme ve Başarı Hesaplama

Şekil 3.2.3.4'te gösterilen fonksiyonlar ile modeli kolayca sorgulanması sağlanmıştır.

Şekil 3.2.3.5'te gösterilen skor tablosunda lineer regresyonun değerlendirme ölçütlerinin çıktı değerleri gösterilmiştir. İlk tablosunda sadece 2019 (tamamı eğitim verisi) yılına kadar ikinci tabloda 2020 ve 2021 yılları da dahil edilmiştir. 2020 ve 2021 yılları eğitim verisinde hiç görmediğinden asıl dikkate alınması gereken değerler ikinci tabloda yer almaktadır.


```

----- Score Table -----
Linear Regression R^2 Score: 0.711
Linear Regression Root Mean Square Error (RMSE): 25929.160
Linear Regression Mean Squared Error (MSE): 672321360.691
Linear Regression Mean Absolute Error (MAE): 20134.496
Linear Regression Mean Absolute Percentage Error (MAPE): 30.518
Linear Regression Symmetric Mean Absolute Percentage Error (SMAPE): 28.360
-----

----- Score Table -----
Linear Regression 2020-2021 R^2 Score: 0.759
Linear Regression 2020-2021 Root Mean Square Error (RMSE): 24838.341
Linear Regression 2020-2021 Mean Squared Error (MSE): 616943204.907
Linear Regression 2020-2021 Mean Absolute Error (MAE): 18689.112
Linear Regression 2020-2021 Mean Absolute Percentage Error (MAPE): 29.476
Linear Regression 2020-2021 Symmetric Mean Absolute Percentage Error (SMAPE): 27.647

```

Şekil 3.2.3.5 Lineer Regresyon Skor Tablosu

3.2.4. Zaman hesaplama

Tablo 3.2.4.1 Makine Öğrenmesi Süre

Algoritma Türü	Ortalama Geçen Süre	Maksimum Geçen Süre	Minimum Geçen Süre	Toplam Geçen Süre
Lineer Regresyon	0.0007 s	0.0078 s	0.0006 s	0.6967 s
Polinomal Regresyon	0.0012 s	0.0019 s	0.0010 s	1.2196 s
Rastgele Orman	0.0476 s	0.0989 s	0.0365 s	50.4804 s
Karar Ağaçları	0.0009 s	0.0017 s	0.0007 s	1.0014 s
XGBoost	0.0191 s	0.2188 s	0.0163 s	20.2609 s

Tablo 3.2.4'te gösterilen süreler 1060 satır 14 sütundan oluşan veri setinden elde edilmiş değerlerdir. Geçen süre makine öğrenmesinin eğitimi, eğitim verisinin testi ve 2020-2021 yıllarının tahmini için geçen sürenin tamamıdır. Her satır ayrı veriler olarak değerlendirildiğinden ortalama geçen süreyi baz alarak değerlendirmek ve toplam süreyi değerlendirmek daha doğru olacaktır. Bu sürelerle görselleştirme ve tahmin değerlerinin ekrana çıktı olarak verilmesi dahil değildir. Bu tablo elde

edilirken test edildiğinde toplam süre 109,4572 s elde edilmiştir. Süreler arasında bu kadar fark olmasının nedeni bütün süreleri tek tek ekrana çıktı olarak gösterilmesidir.

Tablo 3.2.4.2 Toplam Süre Değerlendirme

Görselleştirme: açık Değerlendirme: kapalı	Görselleştirme: kapalı Değerlendirme: açık	Görselleştirme: açık Değerlendirme: açık	Görselleştirme: kapalı Değerlendirme: kapalı
1062.2566 s	137.3815 s	1057.0912 s	109,4572 s

Tablo 3.2.4.2’de okunan değerlere göre süreyi uzatan en büyük etken görselleştirme. Değerlendirme %25 etki eder iken görselleştirme %974 etkilemiştir. İkisi açık olduğu zaman değerlendirme görselleştirmenin yanında etkisi çok olmadığından bir etkisi gözlemlenememiştir. Hızlı sonuçlar istenildiğinde sadece değerlendirme veya ikisi de kapalı şekilde test edilmelidir. Sadece ihtiyaç halinde görselleştirme yapılması zaman tasarrufu açısından daha olumlu sonuçlar vermektedir.

Tablo 3.2.4.1 ve Tablo 3.2.4.2 de bulunan bütün süreler şu özelliklere sahip bilgisayarda test edilmiştir:

- Intel Core i7 10875H
- Nvidia RTX 2070 Super
- 16 x 1 GB 2933 MHz Ram
- Samsung 980 EVO Plus 512 GB Nvme2 SSD 3500Mb/s okuma, 3300Mb/s yazma

3.2.5. Makine öğrenmesi parametrelerinin iyileştirilmesi

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gub rgren, no sea takimata sanctus est Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut lab ore sit et dolore magna. diam nonumy eirmod tempor invidunt ut lab ore sit et dolore magna.

3.2.6. Program akışı

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gub rgren, no sea takimata sanctus est Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore sit et dolore magna. diam nonumy eirmod tempor invidunt ut labore sit et dolore magna.

BÖLÜM 4. SONUÇ VE DEĞERLENDİRME

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gub rgren, no sea takimata sanctus est Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore sit et dolore magna.

KAYNAKLAR

- [1] ATALAY, M., ÇELİK, E., Mehmet Akif Ersoy Üniversitesi Sosyal Bilimler Enstitüsü Dergisi, Büyük Veri Analizinde Yapay Zekâ ve Makine Öğrenmesi Uygulamaları, Cilt No: 99, Sayı :22, 2017 Aralık, s: 161.
- [2] Dr. KARAKUŞ C., Makine Öğrenmesi Temelleri Ders Notu, Erişim Adresi: [https://ckk.com.tr/ders/ML/ML_00 Makine Öğrenmesi Ders Notu.pdf](https://ckk.com.tr/ders/ML/ML_00_Makine_Ogrenmesi_Ders_Notu.pdf) , Erişim Tarihi: 05.11.2021
- [3] DEVECİ KOCAKOÇ, İ., PULAT, M., Manisa Celal Bayar Üniversitesi İ.İ.B.F., Türkiye’de Makine Öğrenmesi ve Karar Ağaçları Alanında Yayınlanmış Tezlerin Bibliyometrik Analizi, Cilt No: 28, Sayı: 2, Yıl: 2021, s: 289
- [4] Makine Öğrenim Projesi Oluşturmak İçin Takip Etmeniz Gereken 6 Adım, Erişim Adresi: <https://www.elektrikport.com/makale-detay/makine-ogrenim-projesi-olusturmak-icin-takip-etmeniz-gereken-6-adim/22513>, Erişim Tarihi: 18.11.2021
- [5] 7 Adımda Makine Öğrenmesi, Erişim Adresi: <https://sibersaldirilar.com/yapay-zeka/7-adimda-makine-ogrenmesi/>, Erişim Tarihi: 18.11.2021
- [6] Python (Programlama Dili), Erişim Adresi: [https://tr.wikipedia.org/wiki/Python_\(programlama_dili\)](https://tr.wikipedia.org/wiki/Python_(programlama_dili)) Erişim Tarihi: 22.11.2021
- [7] Pandas, Erişim Adresi: <https://tr.wikipedia.org/wiki/Pandas>, Erişim Tarihi: 22.11.2021
- [8] Scikit-Learn, Erişim Adresi: <https://en.wikipedia.org/wiki/Scikit-learn>, Erişim Tarihi: 22.11.2021
- [9] Matplotlib, Erişim Adresi: <https://en.wikipedia.org/wiki/Matplotlib>, Erişim Tarihi: 23.11.2021

- [10] Makine Öğrenmesi Algoritmaları, Erişim Adresi: <https://azure.microsoft.com/tr-tr/overview/machine-learning-algorithms/>, Erişim Tarihi: 23.11.2021
- [11] Regresyon Analizi, Erişim Adresi: https://tr.wikipedia.org/wiki/Regresyon_analizi, Erişim Tarihi: 27.11.2021
- [12] Karar Ağaçları ile İstatistiksel Sınıflandırmaya Bakış, Erişim Adresi: <https://www.matematiksel.org/karar-agaclari-ile-istatistiksel-siniflandirmaya-bir-bakis/>, Erişim Tarihi: 28.11.2021
- [13] Karar Ağacı Öğrenmesi, Erişim Adresi: <https://bilgisayarkavramlari.com/2012/04/11/karar-agaci-ogrenmesi-decision-tree-learning/>, Erişim Tarihi: 28.11.2021
- [14] Rastgele Orman Algoritması, Erişim Adresi: <https://devhunteryz.wordpress.com/2018/09/20/rastgele-ormanrandom-forest-algoritmasi/>, Erişim Tarihi: 28.11.2021
- [15] A Beginner's Guide Of XGBoost, Erişim Adresi: <https://towardsdatascience.com/a-beginners-guide-to-xgboost-87f5d4c30ed7>, Erişim Tarihi: 29.11.2021
- [16] Erişim Adresi: https://acikders.ankara.edu.tr/pluginfile.php/62029/mod_resource/content/0/4.%20hafta.pdf, Erişim Tarihi: 30.11.2021
- [17] Regresyon Analizi için Değerlendirme Ölçütleri, Erişim Adresi: <https://ichi.pro/tr/regresyon-analizi-icin-ortak-degerlendirme-olcutleri-82886198762157>, Erişim Tarihi: 30.11.2021
- [18] MSE, RMSE, MAE, MAPE ve Diğer Metrikler, Erişim Tarihi: <https://veribilimcisi.com/2017/07/14/mse-rmse-mae-mape-metrikleri-nedir/>, Erişim Tarihi: 30.11.2021
- [19] Mean Absolute Percentage Error, Erişim Adresi: https://en.wikipedia.org/wiki/Mean_absolute_percentage_error, Erişim Tarihi: 30.11.2021

[20] Eriřim Adresi: www.trademap.org, Eriřim Tarihi: 22.10.2021

EKLER

ÖZGEÇMİŞ

Fatih Gülşen, 07.01.1999 da Bursa’da doğdu. İlk ve orta eğitimini Osmangazi’de tamamladı. Lise eğitimini ise Nilüfer’de tamamladı. 2021 yılı itibarıyla Sakarya Üniversitesi, Bilgisayar ve Bilişim Bilimleri Fakültesinde eğitimine devam etmektedir. 2020 yılında Trade Atlas Bilgi Sistemleri Şirketinde yazılım stajını yaptı ve staj bitiminden itibaren 1 sene burada çalışmaya devam etti. Şu anda aktif olarak Mol Kimya Ltd. Şti.’nin ERP sisteminin tasarımı ve veri tabanı sistemi yöneticiliğini gerçekleştirmektedir. Bunun yanı sıra Kristal TEK Ltd. Şti.’nin ERP sistem tasarımı yapmıştır.

BSM 401 BİLGİSAYAR MÜHENDİSLİĞİ TASARIMI DEĞERLENDİRME VE SÖZLÜ SINAV TUTANAĞI

KONU: DÜNYA TİCARET VERİLERİNİN MODELLENMESİ VE
TAHMİNLENMESİ

ÖĞRENCİLER (Öğrenci No/AD/SOYAD): G171210070- Fatih GÜLŞEN

Değerlendirme Konusu	İstenenler	Not Aralığı	Not
Yazılı Çalışma			
Çalışma klavuzu uygun olarak hazırlanmış mı?	x	0-5	
Teknik Yönden			
Problemin tanımı yapılmış mı?	x	0-5	
Geliştirilecek yazılımın/donanımın mimarisini içeren blok şeması (yazılımlar için veri akış şeması (dfd) da olabilir) çizilerek açıklanmış mı?			
Blok şemadaki birimler arasındaki bilgi akışına ait model/gösterim var mı?			
Yazılımın gereksinim listesi oluşturulmuş mu?			
Kullanılan/kullanılması düşünülen araçlar/teknolojiler anlatılmış mı?			
Donanımların programlanması/konfigürasyonu için yazılım gereksinimleri belirtilmiş mi?			
UML ile modelleme yapılmış mı?			
Veritabanları kullanılmış ise kavramsal model çıkarılmış mı? (Varlık ilişki modeli, noSQL kavramsal modelleri v.b.)			
Projeye yönelik iş-zaman çizelgesi çıkarılarak maliyet analizi yapılmış mı?			
Donanım bileşenlerinin maliyet analizi (prototip-adetli seri üretim vb.) çıkarılmış mı?			
Donanım için gerekli enerji analizi (minimum-uyku-aktif-maksimum) yapılmış mı?			
Grup çalışmalarında grup üyelerinin görev tanımları verilmiş mi (iş-zaman çizelgesinde belirtilebilir)?			
Sürüm denetim sistemi (Version Control System; Git, Subversion v.s.) kullanılmış mı?			
Sistemin genel testi için uygulanan metotlar ve iyileştirme süreçlerinin dökümü verilmiş mi?			
Yazılımın sızma testi yapılmış mı?			
Performans testi yapılmış mı?			
Tasarımın uygulamasında ortaya çıkan uyumsuzluklar ve aksaklıklar belirtilerek çözüm yöntemleri tartışılmış mı?			
Yapılan işlerin zorluk derecesi?	x	0-25	
Sözlü Sınav			
Yapılan sunum başarılı mı?	x	0-5	
Soruları yanıtlama yetkinliği?	x	0-20	
Devam Durumu			
Öğrenci dönem içerisindeki raporlarını düzenli olarak hazırladı mı?	x	0-5	
Diğer Maddeler			
Toplam			

DANIŞMAN: DOÇ.DR. NİLÜFER YURTAY

DANIŞMAN İMZASI: