



AKDENİZ ÜNİVERSİTESİ  
SOSYAL BİLİMLER ENSTİTÜSÜ



Irmak DALDIR

HAVACILIK SEKTÖRÜNDE UÇUŞ GECİKMELERİNİN MAKİNE ÖĞRENMESİ  
TEMELLİ ANALİZİ VE TAHMİNİ

İşletme Ana Bilim Dalı  
Doktora Tezi

Antalya, 2021



AKDENİZ ÜNİVERSİTESİ  
SOSYAL BİLİMLER ENSTİTÜSÜ



Irmak DALDIR

HAVACILIK SEKTÖRÜNDE UÇUŞ GECİKMELERİNİN MAKİNE ÖĞRENMESİ  
TEMELLİ ANALİZİ VE TAHMİNİ

Danışman

Doç. Dr. Ömür TOSUN

İşletme Ana Bilim Dalı

Doktora Tezi

Antalya, 2021

**Akdeniz Üniversitesi**  
**Sosyal Bilimler Enstitüsü Müdürlüğüne,**

Irmak Daldır'ın bu çalışması, jürimiz tarafından İşletme Ana Bilim Dalı Doktora Programı tezi olarak kabul edilmiştir.

Başkan : Prof. Dr. Gökhan AKYÜZ (İmza)

Üye (Danışmanı) : Doç. Dr. Ömür TOSUN (İmza)

Üye : Dr. Öğr. Üyesi M. Serhan SEKRETER (İmza)

Üye : Dr. Öğr. Üyesi Hacer GÜNER GÖREN (İmza)

Üye : Dr. Öğr. Üyesi Ali Cem BAŞARIR (İmza)

Tez Başlığı: Havacılık Sektöründe Uçuş Gecikmelerinin Makine Öğrenmesi Temelli  
Analizi ve Tahmini

Onay : Yukarıdaki imzaların, adı geçen öğretim üyelerine ait olduğunu onaylarım.

Tez Savunma Tarihi : 18/02/2021

Mezuniyet Tarihi : 11/03/2021

(İmza)  
Prof. Dr. Suat KOLUKIRIK  
Müdür

## AKADEMİK BEYAN

Doktora Tezi olarak sunduđum ‘‘Havacılık Sektöründe Uçuş Gecikmelerinin Makine Öğrenmesi Temelli Analizi ve Tahmini’’ adlı bu çalışmanın, akademik kural ve etik değerlere uygun bir biçimde tarafımda yazıldığını, yararlandığım bütün eserlerin kaynakçada gösterildiğini ve çalışma içerisinde bu eserlere atıf yapıldığını belirtir; bunu şerefimle doğrularım.

İmza

**Irmak DALDIR**



**T.C.**  
**AKDENİZ ÜNİVERSİTESİ**  
**SOSYAL BİLİMLER ENSTİTÜSÜ**  
**TEZ ÇALIŞMASI ORJİNALLİK RAPORU**  
**BEYAN BELGESİ**



**SOSYAL BİLİMLER ENSTİTÜSÜ MÜDÜRLÜĞÜ'NE**

<b>ÖĞRENCİ BİLGİLERİ</b>	
<b>Adı-Soyadı</b>	<b>Irmak Daldır</b>
<b>Öğrenci Numarası</b>	<b>20155213010</b>
<b>Enstitü Ana Bilim Dalı</b>	<b>İşletme</b>
<b>Programı</b>	<b>Doktora</b>
<b>Programın Türü</b>	<b>( ) Tezli Yüksek Lisans ( X ) Doktora ( ) Tezsiz Yüksek Lisans</b>
<b>Danışmanın Unvanı, Adı-Soyadı</b>	<b>Doç. Dr. Ömür TOSUN</b>
<b>Tez Başlığı</b>	<b>Havacılık Sektöründe Uçuş Gecikmelerinin Makine Öğrenmesi Temelli Analizi ve Tahmini</b>
<b>Turnitin Ödev Numarası</b>	<b>1527594706</b>

Yukarıda başlığı belirtilen tez çalışmasının a) Kapak sayfası, b) Giriş, c) Ana Bölümler ve d) Sonuç kısımlarından oluşan toplam 128 sayfalık kısmına ilişkin olarak, 08/03/2021 tarihinde tarafımdan Turnitin adlı intihal tespit programından Sosyal Bilimler Enstitüsü Tez Çalışması Orijinallik Raporu Alınması ve Kullanılması Uygulama Esasları'nda belirlenen filtrelemeler uygulanarak alınmış olan ve ekte sunulan rapora göre, tezin/dönem projesinin benzerlik oranı;

alıntılar hariç % 2

alıntılar dahil % 3'tür.

Danışman tarafından uygun olan seçenek işaretlenmelidir:

- ( ) Benzerlik oranları belirlenen limitleri aşmıyor ise;  
Yukarıda yer alan beyanın ve ekte sunulan Tez Çalışması Orijinallik Raporu'nun doğruluğunu onaylarım.
- ( ) Benzerlik oranları belirlenen limitleri aşıyor, ancak tez/dönem projesi danışmanı intihal yapılmadığı kanısında ise;  
Yukarıda yer alan beyanın ve ekte sunulan Tez Çalışması Orijinallik Raporu'nun doğruluğunu onaylar ve Uygulama Esasları'nda öngörülen yüzdeler sınırlarının aşılmasına karşın, aşağıda belirtilen gerekçe ile intihal yapılmadığı kanısında olduğumu beyan ederim.

**Gerekçe:**

Benzerlik taraması yukarıda verilen ölçütlerin ışığı altında tarafımda yapılmıştır. İlgili tezin orijinallik raporunun uygun olduğunu beyan ederim.

08/03/2021

(imzası)  
Doç Dr. Ömür TOSUN

## İÇİNDEKİLER

ŞEKİLLER LİSTESİ .....	viii
TABLolar LİSTESİ .....	x
KISALTMALAR LİSTESİ .....	xii
ÖZET .....	xiii
SUMMARY .....	xiv
TEŞEKKÜR .....	xv
GİRİŞ .....	1

### BİRİNCİ BÖLÜM YAPAY ÖĞRENME VE TEKNİKLERİ

1.1. Yapay Zekâ .....	3
1.1.1. Yapay Zekânın Kullanım Alanları .....	4
1.1.2. Yapay Zekânın Geleceği Tartışmaları .....	5
1.2. Makine Öğrenmesi .....	6
1.3. Derin Öğrenme .....	7
1.4. Makine Öğrenmesi Teknikleri .....	9
1.4.1. Makine Öğrenmesi Aşamaları .....	9
1.4.2. Makine Öğrenmesinde Hata Tipleri .....	9
1.4.3. Makine Öğrenmesinde Aşırı Uyum .....	11
1.4.4. Makine Öğrenmesinde Yetersiz Uyum .....	12
1.4.5. Öğrenme Tipleri .....	13
1.4.5.1. Denetimli Öğrenme .....	13
1.4.5.2. Denetimsiz Öğrenme .....	15
1.4.5.3. Pekiştirmeli (Takviyeli) Öğrenme .....	16
1.4.6. Çalışmada Kullanılan Yöntemler .....	17
1.4.6.1. Yapay Sinir Ağları (Artificial Neural Network) .....	17
1.4.6.2. Karar Ağaçları .....	19
1.4.6.3. Topluluk Öğrenmesi .....	21
1.4.6.4. Rastgele Ormanlar .....	22
1.4.6.5. Gradyan Yükseltme (Gradient Boosting Machines) .....	24
1.4.6.6. EXtreme Gradient Boosting (XGBoost) .....	26
1.4.6.7. LightGBM .....	29
1.4.6.8. CatBoost Algoritması .....	30

### İKİNCİ BÖLÜM HAVAYOLU SEKTÖRÜ

2.1. Havayolu Ulaşımının Tarihçesi .....	33
2.1.1. Günümüzde Havayolu Ulaşımı .....	34

2.1.2.	Havacılıkta Güncel Sorunlar .....	35
2.1.2.1.	Uçuş Güvenliği .....	36
2.1.2.2.	Kırılgan Yapı .....	36
2.1.2.2.1.	Politik Etkiler .....	36
2.1.2.2.2.	Petrol Fiyatlarındaki Dalgalanmalar .....	37
2.1.2.2.3.	Hastalık .....	37
2.1.2.2.4.	Uçuşlarda Yaşanan Gecikmeler .....	38
2.1.2.3.	Çevresel Etkiler .....	39
2.2.	Literatür Taraması .....	39
2.2.1.	Makine Öğrenmesi ve Uçuş Gecikmeleri .....	40
2.2.2.	Uçuş Gecikmeleri Tahmininde Kullanılan Diğer Yöntemler .....	51
2.2.2.1.	Regresyon Modelleri .....	51
2.2.2.2.	Olasılık Modelleri .....	51
2.2.2.3.	Yöneylem Araştırması Modelleri .....	51
2.2.2.4.	Şebeke Temelli Modeller .....	52
2.2.3.	Literatürün Çalışmaya Katkısı .....	52

## ÜÇÜNCÜ BÖLÜM UÇUŞ GECİKMELERİNİN TAHMİNİ

3.1.	Metodoloji .....	54
3.1.1.	Veri Ön İşleme .....	55
3.1.2.	Dengesiz Veri Seti Yönetimi .....	57
3.1.2.1.	Veri Ön İşleme .....	57
3.1.2.1.1.	Yeniden Örnekleme .....	58
3.1.2.1.2.	Aktif Öğrenme .....	59
3.1.2.1.3.	Veri Uzayını Ağırlıklandırma .....	59
3.1.2.2.	Özel Amaçlı Öğrenme Yöntemleri .....	60
3.1.2.3.	İşlem Sonrası Tahmin .....	60
3.1.2.4.	Hibrit yöntemler .....	60
3.1.3.	Öğrenme .....	60
3.1.3.1.	Kılavuz Arama Yöntemi (Grid Search) .....	62
3.1.3.2.	Rastgele Arama .....	62
3.1.3.3.	Bayes Optimizasyonu .....	63
3.1.4.	Modellerin Performans Ölçüm Metrikleri .....	63
3.1.4.1.	Hata Matrisi .....	64
3.1.4.1.1.	Doğruluk (Accuracy) .....	65
3.1.4.1.2.	Duyarlılık (Recall) .....	65
3.1.4.1.3.	Kesinlik (Precision) .....	65
3.1.4.1.4.	F Değeri .....	66
3.1.4.1.5.	Cohen Kappa İstatistiği .....	66
3.1.4.1.6.	İşlem Karakteristik Eğrisi ve Eğri Altında Kalan Alan .....	67

3.1.5. Verilerin İncelenmesi ve Analizi .....	68
3.2. Araştırma Bulguları .....	74
3.2.1. Yapay Sinir Ağları Kullanılarak Elde Edilen Bulgular .....	74
3.2.2. Rastgele Ormanlar .....	76
3.2.3. XGBoost .....	78
3.2.4. LightGBM .....	80
3.2.5. CatBoost .....	82
3.2.6. Tüm Modellerin Performans Metrikleri Aracıđı ile Karşılaştırılması .....	84
<b>SONUÇ .....</b>	<b>91</b>
<b>KAYNAKÇA .....</b>	<b>94</b>
<b>ÖZGEÇMİŞ .....</b>	<b>112</b>



## ŞEKİLLER LİSTESİ

Şekil 1.1 Makine Öğrenmesi ile Derin Öğrenme Arasındaki Fark (Ciptadi, 2019: 1).....	8
Şekil 1.2 Makine Öğrenmesi Aşamaları.....	9
Şekil 1.3 Varyans ve Yanlılık Hata Türleri Gösterimi (Singh, 2018:1).....	10
Şekil 1.4 Gürültü Tipi Hata .....	11
Şekil 1.5 Makine Öğrenmesi Uyum Tipleri (Singh, 2018:1) .....	11
Şekil 1.6 Aşırı Öğrenmeye Karşı Erken Durdurma Yöntemi (Chen, 2020:1) .....	12
Şekil 1.7 Makine Öğrenmesi Öğrenme Türleri (Guru99, 2019:1) .....	13
Şekil 1.8 Makine Öğrenmesi Sürecinde Denetimli Öğrenme .....	14
Şekil 1.9 Denetimsiz Öğrenme Çalışma Prensibi.....	16
Şekil 1.10 Denetimsiz Öğrenme Çalışma Prensibi.....	16
Şekil 1.11 Memeli Sinir Hücresi (Arbib, 1989: 52) .....	17
Şekil 1.12 Yapay Sinir Düğümü.....	18
Şekil 1.13 Yapay Sinir Ağları.....	19
Şekil 1.14 Karar Ağacı Gelişimi.....	20
Şekil 1.15 Topluluk Öğrenme Yapısı .....	21
Şekil 1.16 Torbalama ve Yükseltme Yöntemleri .....	22
Şekil 1.17 Rastgele Orman Yöntemi .....	23
Şekil 1.18 Seviye Bazlı Ağaç Gelişimi .....	27
Şekil 1.19 Yaprak Bazlı Gelişim .....	29
Şekil 1.20 CatBoost Ağaç Gelişim Örneği.....	31
Şekil 2.1 Dünya Hava Kargo Trafiği (Mazareanu, 2020a).....	35
Şekil 2.2 Yıllara Göre Havayolu Kullanan Yolcu Sayısı (Mazareanu, 2020b).....	35
Şekil 3.1 Çalışmanın Metodolojisi .....	54
Şekil 3.2 K-Katmanlı Çapraz Doğrulama Veri Seti Örneği .....	55
Şekil 3.3 Dengesiz Veri Seti Yönetimi (Branco Vd., 2015:14) .....	57
Şekil 3.4 Mükemmele Yakın İşlem Karakteristik Eğrisi.....	67
Şekil 3.5 Yıllara Göre Uçuş Sıklığı .....	68
Şekil 3.6 Hafta Günlerine Göre Uçuş Sıklığı Ve Ortalama Geç Kalma Süresi.....	69
Şekil 3.7 Ay Günlerine Göre Yoğunluk Ve Ortalama Geç Kalma Süresi .....	69
Şekil 3.8 En Sık Kullanılan İlk On Havalimanındaki Gecikme Süreleri .....	70
Şekil 3.9 En Sık Kullan 10 Havaalanında Planlanan Ve Gerçekleşen Kalkış Arasındaki Süre Farkı.....	71

Şekil 3.10 Vaktinde Ve Geciken Uçuş Sıklıkları .....	71
Şekil 3.11 Planlanan Ve Gerçekleşen Uçuş Aşamaları .....	72
Şekil 3.12 Varsayılan Ayarlara Göre Model Performans Sonuçları Sütun Grafik Gösterimi..	86
Şekil 3.13 Varsayılan Ayarlara Göre Model Performans Sonuçları Çizgi Grafik Gösterimi ..	86
Şekil 3.14 Smote Yöntemi İle Oluşturulan Veri Seti İle Modellerin Test Performans Sonuçları Sütun Grafik Gösterimi .....	88
Şekil 3.15 Smote Yöntemi İle Oluşturulan Veri Seti İle Modellerin Test Performans Sonuçları Çizgi Grafik Gösterimi .....	88
Şekil 3.16 Bayes Hiper-Parametre Ayarları İle Modellerin Test Performans Sonuçları Sütun Grafığı.....	90
Şekil 3.17 Bayes Hiper-Parametre Ayarları İle Modellerin Test Performans Sonuçları Çizgi Grafığı.....	90

## TABLOLAR LİSTESİ

Tablo 1.1 Klasik Makine Öğrenmesi ve Derin Öğrenme .....	8
Tablo 2.1 Uçuş Gecikmesi Probleminde Kullanılan Makine Öğrenmesi Yöntemleri .....	49
Tablo 3.1 Kategorik Veri Dönüştürme Örneği .....	56
Tablo 3.2 Hata Matrisi .....	65
Tablo 3.3 Cohen Kappa Değeri Yorumlaması .....	67
Tablo 3.4 En Sık Gecikme Gerçekleşen Havalimanları .....	70
Tablo 3.5 Uçuşa Ait Değişkenler .....	72
Tablo 3.6 Hava Durumu Değişkenleri .....	73
Tablo 3.7 Ysa Varsayılan Parametrelere Göre Hata Matrisi Eğitim Sonuçları .....	75
Tablo 3.8 Ysa Varsayılan Parametrelere Göre Hata Matrisi Test Sonuçları .....	75
Tablo 3.9 Ysa Smote İle Hata Matrisi Eğitim Sonuçları .....	75
Tablo 3.10 Ysa Smote İle Hata Matrisi Test Sonuçları .....	75
Tablo 3.11 Çift Gizli Katman Ysa Varsayılan Parametrelere Göre Eğitim Sonuçları .....	76
Tablo 3.12 Çift Gizli Katman Ysa Varsayılan Parametrelere Göre Test Sonuçları .....	76
Tablo 3.13 Smote İle Çift Gizli Katman Ysa Varsayılan Parametrelere Göre Eğitim Hata Matrisi Sonuçları .....	76
Tablo 3.14 Smote İle Çift Gizli Katman Ysa Varsayılan Parametrelere Göre Test Hata Matrisi Sonuçları .....	76
Tablo 3.15 Rastgele Ormanlar Varsayılan Ayarlar İle Eğitim Aşaması Hata Matrisi Sonuçları .....	77
Tablo 3.16 Rastgele Ormanlar Varsayılan Ayarlar İle Test Aşaması Hata Matrisi Sonuçları ..	77
Tablo 3.17 Smote İle Rastgele Ormanlar Yöntemi Eğitim Aşaması Hata Matrisi Sonuçları ..	77
Tablo 3.18 Smote İle Rastgele Ormanlar Yöntemi Test Aşaması Hata Matrisi Sonuçları .....	77
Tablo 3.19 Rastgele Ormanlar Parametre Uzayı Ve Optimum Değerler .....	78
Tablo 3.20 Hiper-Parametre Ayarları İle Rastgele Orman Tekniği Eğitim Hata Matrisi Sonuçları .....	78
Tablo 3.21 Hiper-Parametre Ayarları İle Rastgele Orman Tekniği Test Hata Matrisi Sonuçları .....	78
Tablo 3.22 XGBoost Varsayılan Ayarlarla Eğitim Aşaması Hata Matrisi Sonuçları .....	79
Tablo 3.23 XGBoost Varsayılan Ayarlarla Test Aşaması Hata Matrisi Sonuçları .....	79
Tablo 3.24 XGBoost Ve Smote Kullanımıyla Eğitim Aşaması Hata Matrisi Sonuçları .....	79
Tablo 3.25 XGBoost Ve Smote Kullanımıyla Test Aşaması Hata Matrisi Sonuçları .....	79

Tablo 3.26 XGBoost Parametre Uzayı Ve Optimum Değerler .....	79
Tablo 3.27 Hiper-Parametre Ayarları İle XGBoost Eğitim Hata Matrisi.....	80
Tablo 3.28 Hiper-Parametre Ayarları İle XGBoost Test Hata Matrisi.....	80
Tablo 3.29 LightGBM Varsayılan Ayarlar İle Eğitim Hata Matrisi Sonuçları.....	80
Tablo 3.30 LightGBM Varsayılan Ayarlar İle Test Hata Matrisi Sonuçları.....	80
Tablo 3.31 LightGBM Ve Smote Kullanımıyla Eğitim Aşaması Hata Matrisi Sonuçları.....	81
Tablo 3.32 LightGBM Ve Smote Kullanımıyla Test Aşaması Hata Matrisi Sonuçları.....	81
Tablo 3.33 LightGBM Parametre Uzayı Ve Optimum Değerler .....	81
Tablo 3.34 Hiper-Parametre Ayarları İle LightGBM Eğitim Hata Matrisi.....	82
Tablo 3.35 Hiper-Parametre Ayarları İle LightGBM Test Hata Matrisi.....	82
Tablo 3.36 CatBoost Varsayılan Ayarlar İle Eğitim Hata Matrisi Sonuçları.....	82
Tablo 3.37 CatBoost Varsayılan Ayarlar İle Test Hata Matrisi Sonuçları.....	82
Tablo 3.38 CatBoost Ve Smote Kullanımıyla Eğitim Aşaması Hata Matrisi Sonuçları.....	83
Tablo 3.39 CatBoost Ve Smote Kullanımıyla Test Aşaması Hata Matrisi Sonuçları.....	83
Tablo 3.40 CatBoost Parametre Uzayı Ve Optimum Değerler .....	83
Tablo 3.41 Hiper-Parametre Ayarları İle CatBoost Eğitim Hata Matrisi.....	83
Tablo 3.42 Hiper-Parametre Ayarları İle CatBoost Test Hata Matrisi.....	83
Tablo 3.43 Varsayılan Ayarlara Göre Model Performans Sonuçları .....	84
Tablo 3.44 Smote Yöntemi İle Oluşturulan Veri Seti İle Model Performans Sonuçları.....	87
Tablo 3.45 Bayes Hiper-Parametre Ayarları İle Modellerin Performans Sonuçları .....	89

## KISALTMALAR LİSTESİ

AB	AdaBoost	KA	Karar Ağacı
BR	Bayes Ridge	K-EYK	K-en Yakın Komşular
ÇDR	Çoklu Doğrusal Regresyon	LR	Lojistik Regresyon
DİA	Derin İnanç Ağları	NB	Naive Bayes
DS	Decision Stump	OYGS	Otomatik Yer Gözlem Sistemleri
DVM	Destek Vektör Makineleri	RO	Rastgele Orman
EA	Ekstra Ağaçlar	RT	RepTree
EAKA	Eğri Altında Kalan Alan	TAYS	Trafik Akış Yönetim Sistemi
GA	Gradyan Artırma	TÇÇD	Tek Çıkışlı Çapraz Doğrulama
GOSS	Gradient-Based One-Side Sampling	TSA	Tekrarlayan Sinir Ağları
HSPÖ	Havacılık Sistemi Performans Ölçütleri	UKSB	Uzun Kısa Süreli Bellek
IATA	Uluslararası Hava Taşımacılığı Birliği		

## ÖZET

Dünyada artan hareketlilik havayolu ulaşımına olan talebi artırmıştır. Yolcu taşımacılığının yanı sıra yükte hafif pahada ağır olarak tanımlanabilecek malzeme ve ürünlerin taşımacılığı konusunda da havayolu taşımacılığının payı günden güne artmaktadır. Yıllar içerisinde hem yolcu hem de yük taşımacılığı konusunda artan talep nedeniyle uçuş gecikmeleri artmış ve bu durum paydaşlar üzerinde olumsuz etkisi yaratmıştır. Gecikmelerin son ana kadar öngörülememesi havaalanlarında yığılma, bazı yolcular için uzun süreli bekleme veya aktarmalı uçuşları kaçırma gibi birçok problemi beraberinde getirmektedir.

Havayolu ulaşımına olan talebin artmasının yanı sıra, teknolojik imkanların da artması bu alanda oluşan verilerin daha kolay depolanmasına ve oluşan bu büyük veri ile faydalı analizler yapılmasına imkan tanımaktadır. Bu analizlerin yapılabilmesi için kullanılan makine öğrenmesi yöntemleri son yıllarda önemli bir ivme kazanmıştır.

Bu çalışmada, uçak gecikmelerinin sebep olduğu problemleri azaltabilmek adına makine öğrenmesi yöntemleri ile detaylı uçuş ve hava durumu verileri birlikte değerlendirilerek uçak gecikmelerinin tahmin edilmesi amaçlanmıştır. Bu doğrultuda, sektörde faaliyet gösteren uluslararası bir havayolu şirketiyle irtibata geçilerek çalışmada kullanılacak olan uçuş verileri temin edilmiştir. Uçuş bilgileri kapsamında uçakların planlanan kalkış ve varış zamanları, uçakların piste çıkış/varış zamanları, hangi havalimanları arasında seyahat gerçekleştirdiği, seyahatlerin tam tarih bilgileri gibi konularda üç yıllık (2016-2017-2018) veriler toplanmıştır. Bu veriye ek olarak hava durumuna ilişkin veriler ise çevrimiçi kaynaklardan elde edilmiştir. Hava durumu için ise, kalkış ve varış havalimanlarının planlanan kalkış/varış saatlerindeki hava durumu verileri (sıcaklık, basınç, bulutluluk durumu, görüş mesafesi, olağanüstü hava durumları vb.) dikkate alınmıştır. Havayolu firmasının üç yıllık uçuş verileri ile uçak kalkış ve varış zamanındaki hava durumu eşleştirilerek uçuşların durumunu tahmin etmek üzere veri seti oluşturulmuştur. Oluşturulan veri seti, makine öğrenme yöntemleri içerisinde yer alan Yapay Sinir Ağları, Rastgele Ormanlar, XGBoost, LightGBM ve CatBoost ile incelenmiş ve analiz edilmiştir. Bu modellerin performansı ise hata matrisi temelli çeşitli birçok performans metriklerine göre değerlendirilmiş ve elde edilen bulgular birbirleri ile karşılaştırılarak yorumlanmıştır.

**Anahtar Kelimeler:** Uçuş Gecikmesi Tahmini, Makine Öğrenmesi, Sınıflandırma

## SUMMARY

### MACHINE LEARNING BASED ANALYSIS AND PREDICTION OF FLIGHT DELAYS IN AVIATION INDUSTRY

The demand for air transportation has risen as the world's mobility has increased. In addition to the demand for passenger transport, the share of airline transportation, particularly for light and valuable goods, is growing day by day. Increasing demand for both passenger and cargo over the years lead to flight delays, and which has a negative impact on many stakeholders. The fact that the delays cannot be predicted until the last moment conduce to many problems such as congestion at airports, long waiting periods for some passengers or missing connected flights.

In addition to the increase in demand for airline transportation, technological developments are enabled to store data easily and using this big data it has become possible to make useful analyzes. Machine learning methods have gained significant momentum in recent years in terms of making these analyzes.

In this study, it is aimed to predict aircraft delays by machine learning methods with using detailed flight and weather data together in order to reduce the problems caused by the delays. Accordingly, an international airline company which is operating in the aviation sector is contacted and flight data to be used in the study is obtained through them. Within the scope of flight information, three-year (2016-2017-2018) data are collected such as the planned departure and arrival times of the planes, the departure arrival times of the planes, the airports they travel between, and the exact date information of the trips. In addition to this data, weather data were obtained from online sources. For this, the weather data (temperature, pressure, cloudiness, visibility, extraordinary weather conditions, etc.) at the planned departure /arrival times of the departure and arrival airports are taken into consideration. A data set was created to estimate the status of the flights by matching the airline company's three-year flight data with the weather at the time of aircraft departure and arrival. The created data set was examined and analyzed with Artificial Neural Networks, Random Forests, Extreme Boosting, Light Gradient Boosting and Categorical Boosting, which are among machine learning methods. The performance of these models is evaluated according to various performance metrics based on error matrix and the findings obtained are compared with each other and interpreted.

**Key Words:** Flight Delay Prediction, Machine Learning, Classification

## TEŞEKKÜR

Çalışmamın her aşamasında bana destek olan, bilgi ve deneyimleri ile yol gösteren danışman hocam Sayın Doç. Dr. Ömür TOSUN'a yürekten teşekkür ederim.

Tez izleme komitelerinde yer alıp görüş ve önerileri ile çalışmaya çok büyük katkılar sağlayan Prof. Dr. Gökhan AKYÜZ'e ve Dr. Öğr. Üyesi Ali Cem BAŞARIR'a çok teşekkür ederim.

Tüm doktora süreci boyunca bana destek olan oda arkadaşlarım Dr. Öğr. Üyesi Beyhan AKAY'a ve Öğr. Gör. İsmail KARAYÜN'e destek ve sabırları için teşekkür ederim.

Beni hep destekleyen ve yanımda olan annem Zühal DÖNMEZ'e yürekten teşekkür ederim.

Son olarak 2211-E Doğrudan Yurt İçi Doktora Burs Programı ile doktora eğitimime katkıda bulunan TÜBİTAK'a teşekkür ederim.

Irmak DALDIR



## GİRİŞ

Makine öğrenmesi teknikleri günümüzde birçok farklı alanda uygulanmaya başlanmış ve elde edilen başarılı sonuçlar sayesinde daha fazla araştırmacının dikkatini çekmeye başlamıştır. 1950’li yıllara dayanan bir geçmişi bulunan makine öğrenmesi yardımıyla o dönemde her soruna özgü çözümler üretilmekte olup; her durum için ayrı ayrı hazırlanan bu çözümler sonuca ancak birçok zincir işlemden geçerek ulaşabilmesine neden olmuştur. Bu şekilde problemlerin çözüm süreci uzamış ve “kombinasyonel patlama” ismi verilen bu durum ile her önermenin doğruluğu araştırılırken sonuç alınamaması durumu ortaya çıkmıştır. Günümüz bilgisayar altyapısı bile her durumun incelenmesi konusunda yetersiz kalabilirken, o dönemde bu tip karar destek sistemlerinin kullanılıyor olması yetersiz işlemci hızı ve yetersiz donanım yüzünden tercih edilmez bir hale gelmiştir. Yapay zekânın kullanımı ile ilgili yaşanan bu olumsuzlukların sonrasında 1980’lerin başında Japonya öncülüğünde bir girişim olduysa da kurula dayalı programlar ile istenilen başarı elde edilememiştir. 1990’lardan sonra ise ortaya koyulan deneyime dayalı öğrenme fikri, teknolojik altyapı gelişimi ve veriye ulaşım kolaylığı gibi faktörlerin bir araya gelmesi ile günümüzde birçok farklı problemin çözümünde kullanılabilir hale gelmiştir. Bu gelişmelerin nihai sonucunun insan zekâsını geçerek bir süper zekâ haline gelmesi araştırmacılar tarafından olası bir gelecek olarak görülmektedir (Bostrom, 2020: 14). Günümüzde makine öğrenme yöntemleri sınırlı bir çerçevede olsa dahi, birçok problemin etkin bir şekilde çözüme ulaşmasını sağlamaktadır. Örneğin; e-posta adreslerini güvenli tutan sistemler, internet alışverişlerinde ürün önerisi yapan sistemler gibi alanlarda makine öğrenmesi karşımıza çıkmakta ve hayatı kolaylaştırmaktadır.

Bu çalışmada havayolu sektörünü ve kullanıcılarını doğrudan etkileyen uçakların planlanan uçuş saatinden geç kalkması durumunu makine öğrenmesi yöntemleri ile tahmin edilmesi amaçlanmaktadır. Literatüre bakıldığında bu alanda yapılan çalışmalar genellikle havalimanları baz alınarak gerçekleştirilmiştir. Bu çalışmada havalimanları yerine önceden belirlenmiş olan bir havayolu şirketinden alınan resmi veriler için farklı makine öğrenmesi teknikleri yardımıyla analizler yapılmış olması, çalışmanın özgün değerini oluşturmaktadır. Ayrıca çalışma kapsamında alanda temel olarak kabul edilen makine öğrenmesi yöntemlerinin yanı sıra, güncel yöntemler de kullanılarak karşılaştırmalı analizler yapılmış ve seçilen performans metriklerine göre oldukça yüksek bir başarı elde edilmiştir.

Çalışmanın birinci bölümünde yapay zekâ, makine öğrenmesi ve derin öğrenme ile ilgili kavramsal bilgiler verilmiş olup, sonrasında çalışma kapsamında dikkate alınan makine öğrenmesi teknikleri incelenmiştir. Yöntem seçiminde en eski yöntem olan Yapay Sinir Ağları,

benzer problemlerin çözümünde sıkça kullanılmış olan Rastgele Ormanlar birbirine rakip olarak gösterilen ve diğer yöntemlere göre göreceli olarak yeni olan Gradyan Artma temelli yöntemler olan XGBoost, LightGBM ve CatBoost makine öğrenmesi yöntemleri kullanılmıştır. İkinci bölümde ise havayolu sektörünün durumu, gecikmelerin etkileri ve literatür incelemesi yer almaktadır. Üçüncü bölümde ise makine öğrenme yöntemlerini daha verimli hale getirmek için kullanılan geliştiriciler, kullanılan veri seti tanıtımı ve uygulamaya yer verilmiştir. Sonuç bölümünde ise elde edilen tüm sonuçlar yorumlanarak, çalışmanın gelecekte geliştirilebileceği alanlar hakkında bilgi verilmiştir.

## BİRİNCİ BÖLÜM

### YAPAY ÖĞRENME VE TEKNİKLERİ

Sanayi devrimleri ile makinelerden destek almaya başlayan insan, buhar gücü ile çalışan sistemlerden seri üretime, sonrasında bilgisayarlar ile otomasyonun gücüne aşına olmuş durumdadır. Yakın zamana kadar makinelerden genellikle insan emeğinin tehlikeli, yetersiz veya pahalı olduğu durumlarda bir yardımcı gibi destek alınmıştır. Günümüzde ise makinelere “öğrenme” kavramının tanıtılmasıyla birçok farklı alanda makinelerden destek alınır hale gelmiştir. Öğrenmek, Türk Dil Kurumu’na göre bilgi edinmek demektir. Makineler için öğrenme ise bütün olarak anlam ifade etmeyen veri setleri içinden ihtiyaç duyulan anlamlı bilgileri elde ederek, gelecekte karşılaşılabilecek olay için deneyim sahibi olmaktır. Bu bölümde öğrenme kavramının doğuşunu sağlayan ve genel çevresini oluşturan yapay zekâ kavramı incelenmiş sonrasında makine öğrenmesine ve derin öğrenmeye yer verilmiştir. Son aşamada ise çalışmada kullanılan makine öğrenmesi yöntemlerinin çalışma prensipleri tanıtılmıştır.

#### 1.1. Yapay Zekâ

Bilgisayar ortamında bir işi gerçekleştirebilmek için algoritmalara ihtiyaç duyulmaktadır. Algoritma, “bilgisayarlara nelerin yapılması gerektiğini söyleyen talimatlar dizisidir” (Domingos, 2017: 27). Yapay zekâ ise bu kalıbın dışında yer alan her durum için algoritma gereksiniminden önce, insan gibi karşılaştığı durumlardan öğrenen duruma göre karar alabilen bir yapıdadır. “Yapay zekâ” terimini de ortaya koyan John McCarty’nin bu alanı “akıllı makine yapma bilimi ve mühendisliği” olarak tanımlamıştır (Peart, 2020: 1). Düşünebilen makine fikri insanlık için son derece eski bir fikir olsa da günümüz dünyasında gerçek anlamda yapay zekâ kavramından daha yeni bahsedebilmektedir (Goodfellow vd., 2016: 1). Yapay zekâ bilimsel olarak 1950 yıllarından itibaren kendinden söz ettirmektedir. Tanımı ise; insan gibi düşünebilme, akıl yürütme, anlam çıkarabilme, genelleme yapabilme, deneyimlere göre öğrenebilme yeteneğine sahip bir bilgisayar veya makine şeklindedir (Yılmaz, 2017: 7). Olayları idrak ederek çözümlenmek insan beyni için daha kolayken, sayısal işlemleri ve genelleme yapabilmek daha karmaşık ve zor olmaktadır (Elmas, 2007: 21). Tersini durum yani sayısal işlemleri başarı ile yerine getirebilmek makineler için daha kolayken olayları idrak ederek çözmek daha zorlayıcı olabilmektedir. Bu durum Turing, John von Neuman, Harvey ve Kyoto ödüllü bilgisayar mühendisi Donald Kunth tarafından şöyle değerlendirilmiştir: “Yapay Zekâ bu zamana kadar özünde ‘düşünmeyi’ gerektiren her şeyi başarmış, fakat insanların ve hayvanların ‘düşünmeden’ (kendisinden) yaptığı birçok şeyi yapmayı başaramamıştır” demiştir

(Nilsson, 2009: 24). Elbette insanlar yapay zekânın hâlâ gerçekleştiremediği birçok düşünme kabiliyetine sahiptir. Örneğin, hali hazırda nitelikli bir kitap yazabilen veya felsefe yapabilen bir yapay zekâ ortaya çıkmamıştır. Bir makine için de gördüğü bir nesneyi tanımlamak sayısal bir işlemi gerçekleştirmekten daha zordur. Bu duruma da gözetilerek yapay zekâ kavramından ayrıca “hesaba dayalı zekâ” (computational intelligence) olarak da isimlendirilmiştir (Poole vd., 1998: 2). Yapay zekâ ortamı algılayıp amaçlarını gerçekleştirme şansını en yüksek düzeye çıkaracak olan aksiyonu alan sistem şeklindedir (Nilsson N. J., 1998: 1). Diğer tanımlara göre daha eski olsa da güncelliğini koruyan bir diğer tanım ise: “Yapay zekâ an itibariyle insanların iyi olduğu konuların bilgisayarlar tarafından nasıl yapılabileceğinin çalışmakta olduğu alandır” (Rich, 1983: 1).

### 1.1.1. Yapay Zekânın Kullanım Alanları

Yapay zekânın güncel kullanımını incelediğimizde birçok büyük firmanın bu alanda yüksek yatırımları olduğunu görülmektedir. Bu duruma örnek olarak DHL, Amazon, Facebook ve Google’ın sağladığı hizmetler verilebilir. DHL, 2015 yılında 26 milyon dolar olan yapay zekâ yatırımını 2016 yılında 39 milyon dolara çıkarmıştır. Bu yatırımları ile yaptıkları ise, derin öğrenme temelli doğal dil işleme ile basit konularda müşterilere yardımcı olan bir sistem kurmaktır (Burt, 2018: 1). Yardımcı otomatik sesli yanıt sisteminin çözüm bulamadığı durumlarda müşteri yetkilisine yönlendirme yapılmaktadır. Ayrıca Resilience360 isimli bulut temelli platformları ise müşterilere iş süreçleri ile ilgili gönderi durumu, risk veya tedarik zincirinde oluşabilecek aksaklıkları izleme imkânı sunmak üzere geliştirilmiştir. Makine öğrenmesi ve doğal dil işleme süreci algoritmaları ile tedarik sorunlarını tespit etmek amaçlanmıştır. Tespitin yapılabilmesi için 8 milyon ileti 300.000 çevrimiçi ve sosyal medya kaynağı takip edilmektedir. DHL’in bir diğer yapay zekâ yatırımı olan Global Trade Barometer ise çeşitli ülkelerin erken-orta olarak belirlenmiş olan dönemlerde üretim hareketliliklerinin tahminini yapmaktır. Böylece deniz ve hava taşımacılığı konusunda küresel hareketlerinin nasıl olacağını üç ay öncesinden tahmin etmek amaçlanmaktadır. Bunun için kullanılan yapay zekâ algoritması ise 240 milyon farklı değişkene başvurmaktadır. Daha güncel yatırımları incelendiğinde 2019 yılında sadece Kuzey Amerika’daki depolarını nesnelere interneti sistemi ile güncellemek üzere 300 milyon dolarlık yatırım yer aldığı görülmektedir (Johnson, 2018: 1). Tüm bu sistemlerin yanı sıra günlük hayatta kullanımı olağan hale gelmiş yapay zekâ sistemlerine örnek olarak: Amazon’un 20 yıldan uzun süredir yapay zekâ teknolojileri ile müşterilerine ürün önerilerinde bulunması verilebilir (Smith ve Linden, 2017: 12). En güçlü sosyal medya ağlarından biri olan Facebook ise yüz tanıma aracı ile kullanıcıları fotoğrafları

kendileri etiketlemeksizin eklenen fotoğraftaki kişilerin kim olduklarını bularak kullanıcılarına hizmet sağlamaktadır (Narayanan, 2019: 1). Google çeviri sisteminin temelinde de yapay zekâ kullanılmaktadır (Wong, 2016: 1).

### 1.1.2. Yapay Zekânın Geleceği Tartışmaları

Yapay zekâ sunduğu avantajlar haricinde de adından çokça bahsedilen bir alan olma özelliğine sahiptir. İnsanlığın günümüze kadar olan mücadelesi hep yaşayan organizmalar, kendisi veya doğa ile olmuştur. İlk kez bilinci olmayan ve zekâsı küçümsenemez, rakibi olarak tanımlanabilecek yapay zekâyı kendisinin yaratmak üzere olduğu düşünülmektedir (Gürsakal, 2017: 2). Rekabetin ilk başlangıç noktası iş gücü alanında yaşanmış ve artarak devam etmektedir. Birçok sektörde ağır ve tehlikeli işlerde insan işgücü yerine makineler tercih edilmeye çoktan başlanmıştır. Örneğin, Oxford Economics'in (2019) raporuna göre son 20 yılda kullanılan makine sayısı üçe katlanarak 2,25 milyona çıkmış durumdadır, 2030 yılına kadar 20 milyona ulaşacağı tahmin edilmektedir. 2030 yılında ise Çin'de 14 milyon kişinin işinin makineler tarafından yapılacağı öngörülmektedir (Oxford Economics, 2019: 4). Bu durum genel olarak otomasyonun etkisidir. Buna ek olarak yapay zekâ ile birleşmiş robot ve makine gibi gelişmiş sistemlere de akıllı otomasyon ismi verilmektedir (PwC, 2018: 3). Sadece akıllı otomasyonların 2030 yılında dünya gayri safi hasılası içinde %14'lük bir payı olması beklenmektedir (PwC, 2017: 1). Yapay Zekâ hakkında önde gelen kişilerin farklı görüşleri olduğu görülmüştür. Örneğin, SpaceX ve Tesla gibi teknolojik alt yapısı kuvvetli şirketlerin üst düzey yöneticisi olan Elon Musk attığı tweet ile yapay zekâ konusundaki yarışların sonucunun Üçüncü Dünya Savaşı'na yol açacağını düşündüğünü beyan etmiştir (Musk, 2017: 1). Bu konuda Stephen Hawking de olumsuz beklentilerini dile getirmiştir. Teknolojinin insanlardan daha üstün bir aşamaya gelerek insanları ortadan kaldıracağını düşündüğünü ifade etmiştir (BBC, 2014: 1). Bill Gates ise 2015 yılında verdiği röportajda yapay zekâ konusunda kaygı duyulması gerektiğini söylemiştir (BBC, 2015: 1). Diğer tarafta da istikrar, güvenilirlik ve maliyet avantajları sağlayan yapay zekâ aynı zamanda problem çözümlerindeki belirsizliği ortadan kaldırmakta ve hızlı sonuç sağlamaktadır (Chowdhury ve Sadek, 2012: 6). Yadsınamaz faydalarından birkaçı; insan hatasını ortadan kaldırması, risk almaya çekinmemesi, sürekli çalışabilmesi, tekrarlayan işlerde de başarılı olması ve hızlı karar verebilmesi olarak özetlenmektedir. Ayrıca yeni iş modellerinin ortaya çıkmasında ise teknoloji ve yapay zekâdaki gelişmeler son derece önemli katkılar sağlamaktadır. Örneğin, tatil yeri kiralama uygulaması olan Airbnb tek bir odası olmadan dünya üzerinde birçok yerde aracılık yapmış ve 2017 yılında 96 milyon dolar kar elde etmiştir (Gmelich, 2019: 1). Carnegie Mellon Üniversitesi Makine

Öğrenmesi Bölüm Başkanı Prof. Manuela Veloso röportajında görüşlerini çok daha iyimser bir çerçeveden paylaşmıştır. Veloso'ya göre gelecekte kişileri otomatik asistanlarından ayırmanın mümkün olmayacağını aksi halde ikisinin de diğeri olmadan yapamayacağını öngörmektedir. Gelişmekte olan yapay zekânın gelecekte tam olarak nasıl bir sistem olarak işleyeceğini tahmin etmek mümkün olmasa da iyi bir aşamada olduğumuzu konu hakkında tamamen iyimser bir bakış açısını desteklediğini, fakirlik dahil günümüz sorunlarına yapay zekâ sayesinde çözüm bulunacağını düşündüğünü dile getirmiştir (Brandom, 2016: 1).

## 1.2. Makine Öğrenmesi

Makine, “enerji üreten, enerjiyi bir formdan başka bir forma dönüştüren, enerjiyi bir yerden başka bir yere ileten, enerjiyi kullanarak faydalı mekanik iş ortaya koyan teknik sistemlerdir” (İşler, 2018: 5). Makine öğrenmesinde ise asıl kast edilen makine; bilgisayarlardır. Günümüz bilgisayarları yapıları itibariyle birçok transistörden oluşmaktadır. Transistörler etkin olma durumunda 1, olmama durumunda 0 değerini alır. Transistörlerin yol haritası ise algoritmalarıdır. Algoritmalar sayesinde alınan komutlarla bilgisayarlar ne yapmaları gerektiğini bilmektedirler. Algoritmaların çalışabilmesi için kesin, belirsizlikten uzak olmaları, hangi durumda sonlanacağı belirlenmiş olması gerekmektedir aksi halde hiç çalışmamaları veya sonsuz döngüye girmeleri mümkündür. Bu kriterlere göre yazılmış algoritmalar, bilgisayarlara hangi durumda hangi adımları takip etmesi gerektiği konusunda yol haritası olan talimatlar dizisinden oluşmaktadır. Makine öğrenmesi yöntemleri de algoritmalar sayesinde çalışmaktadır; fakat ilk ölçümlerine göre öğrenme gerçekleştirir ve sonraki durumlarda verdiği sonuçlar değişmektedir. Diğer bir deyişle geçmiş deneyimlerine göre kendini geliştirebilme, bu durumda öğrenme, yapabilmektedir. Tanımsal olarak incelendiğinde makine öğrenmesi, Patgiri vd. (2020) göre büyük veri setleri üzerinde çalışabilen mantıksal bir analizdir. Bir başka tanıma göre ise makine öğrenmesi, bilgisayar programlama bilimi ve hatta sanatının, verilerden öğrenebilme gerçekleştirmek üzere kullanılması olarak da tanımlanmıştır (Geron, 2019: 4).

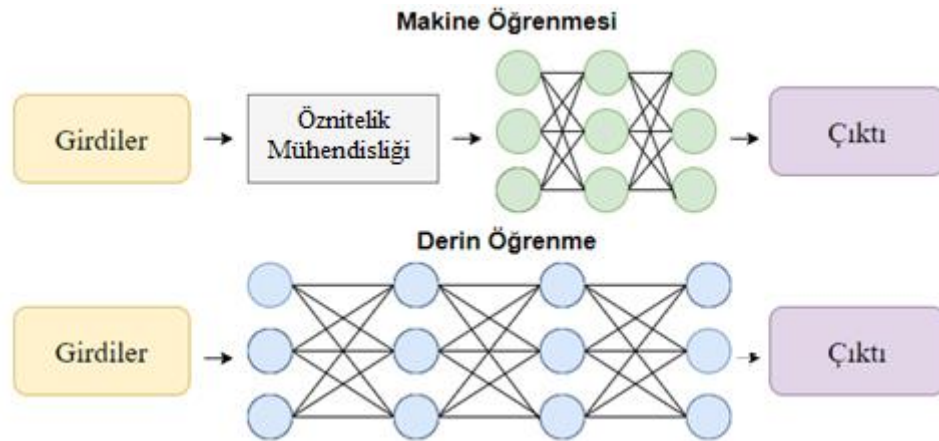
Yapay zekâ ile makine öğrenmesi terimleri sık sık birbirinin yerine kullanılmakta olsa da yapay zekâ aslında tanımında belirtildiği üzere seçim yapma yeteneğine sahip bir sistemdir. Asıl amacı başarıyı sağlamaktır. Makine öğrenmesi ise bu durumdan farklı olarak en faydalı sonucu bulmakta, diğer bir deyişle amaçlanan değeri ortaya koymaktadır. Belirlenen maksimize edilecek faydayı bulmakta kullanılsa dahi makine öğrenmesi yöntemleri kendi kendine aksiyon almamaktadır. Makine öğrenmesi yöntemleri bu özellikleri ile yapay zekânın alt başlığı olarak görülmektedir (Das vd., 2015: 31).

Makine öğrenmesinin temel olarak diğer algoritmalarından farkı ise programlayıcısının her durumu öngörmek zorunda kalmadığı ve bütün sonuçları vermediği bir ortamda makinenin geçmiş deneyimi veya örnek veriyi kullanarak yeni duruma uyum sağlayabilmesidir (Alpaydın, 2014: 3). Bir makine öğrenmesi algoritması belirlenmiş amacı yerine getirmek için önceden belirlenen performans metriklerini en iyileyecek olan parametre değerlerini bulmaktır (Alpaydın, 2017: 3). Bu durum makine öğrenmesinin farklı yönünü ortaya koymaktadır. Yapay zekânın gelişimi açısından ise makine öğrenmesi algoritmalarının önemi daha da ortaya çıkmaktadır (Ertel, 2011: 24).

Makine öğrenmesinin bir diğer avantajı da büyük veriler aracılığı ile kolaylıkla hipotezleri test etmeye imkân sağlamasıdır (Domingos, 2017: 41). Büyük verinin günümüz sistemleri ile daha kolay elde edilebilir ve yaratılabilir olması geleneksel yöntemlerle çok daha yavaş olan hipotez üretme ve test etme süreçlerini de değiştirmektedir. Geleneksel yöntemlerle matematiksel bir teoremin kanıtlanması için öncelikle yardımcı önermelerin titizlikle kanıtlanması gerekmekte ve bu da derin bir matematiksel altyapı gerektirmektedir. Gelişmekte olan makine öğrenmesi yöntemleri bu durumu da kolaylaştırmaktadır (Nilsson, 1998: 59).

### 1.3. Derin Öğrenme

Derin öğrenme Ertel'e (2017) göre 1995 yılından beri çalışılmaktadır, Nvidia'a (2016) göre ise 2010'lu yıllarda çalışılmaya başlanmıştır. Derin öğrenme teknikleri makine öğrenmesinin bir alt dalını oluşturmaktadır (Nvidia, 2016: 1). Klasik makine öğrenmesi yöntemleriyle karşılaştırmalı olarak incelendiğinde derin öğrenmenin en önemli farkı öznetelik mühendisliği için ek çalışma gerektirmemesidir (Skiena, 2017: 377). Şekil 1.1'de gösterildiği üzere ek düzenlemelere ihtiyaç duymaksızın daha fazla hücre ve katman (Ravi vd., 2017: 4) ile bu aşamaları atlamaktadır. Ek düzenleme aşamasında veri saflaştırılır bu duruma örnek olarak uç değerlerin veri setinden çıkarılması veya eksik verilerin düzenlenmesi gibi çalışmalar örnek verilebilmektedir. Yapılandırılmış veri, sütunlar altında belirli durumları niteleyen veri olarak da açıklanabilir; yapılandırılmamış veri ise ses, görüntü veya metin şeklindeki verilerden de oluşabilmektedir.



**Şekil 1.1 Makine Öğrenmesi ile Derin Öğrenme Arasındaki Fark (Ciptadi, 2019:1'den alınıp yazar tarafından düzenlenmiştir)**

Makine öğrenmesi yöntemleri ile derin öğrenme yöntemleri yaklaşım farklılıkları Tablo 1.1'de özetlenmiştir. Tabloya göre makine öğrenmesi daha çok kullanıcı müdahalesi gerektirmekteyken derin öğrenme yöntemleri soyut kavramlar üzerinden de öğrenmeyi gerçekleştirmektedir. Derin öğrenmede eldeki her veri işlenmeden girdi olarak kullanılır ve eğitim tüm değişkenler üzerinden gerçekleştirilir. Makine öğrenmesinde ise seçilen yönteme göre eğitim yapılmadan önce parametre değerleri belirlenir ve modülün öğrenmesi aşamalar halinde gerçekleşmektedir.

**Tablo 1.1 Klasik Makine Öğrenmesi ve Derin Öğrenme**

	Öğrenme	Model Kurulumu	Model Eğitimi	Donanım Gereksinimi
Klasik makine öğrenmesi	Uzman alan bilgisi gerektirir Değişkenler uzmanlar tarafından düzenlenir	Çıkarılmış değişkenler veriye dayalı model oluşturulmak için kullanılır Genellikle sığ yapılar üzerinden model kurulur	Her modül aşama aşama eğitilir	Normal bir donanımla da kullanılabilir
Derin Öğrenme	Değişkenler verinin soyut temsilleri üzerinden öğrenilir	Çok katmalı doğrusal olmayan uçtan uca hiyerarşik model kurulur	Parametreler birlikte eğitilir.	Yüksek grafik işlemcili donanım gerektirir

(Wang, vd., 2018: 148'den alınıp yazar tarafından düzenlenmiştir)

1990'lı yıllarda YSA modelleri iki veya daha fazla olduğu durumda derin sinir ağlar olarak kabul edilmekteydi. Fakat günümüzde yüzlerce gizli katmandan oluşan ağlara "derin" adı verilmektedir (Geron, 2019: 286). Derin öğrenme, genellikle kompleks yapısı sebebi ile bilgisayar görmesi, doğal dil işleme ve konuşma tanıma gibi alanlarda kullanılmaktadır. Bunun sebebi daha fazla öğrenebilme ve daha detaylı olarak analiz edebilme kabiliyetine sahip olmasıdır. Belli temellere sahip olsa dahi derin öğrenmenin özünde nasıl bu kadar başarılı olduğu konusu henüz netleştirilememiştir (Skiena, 2017: 378).

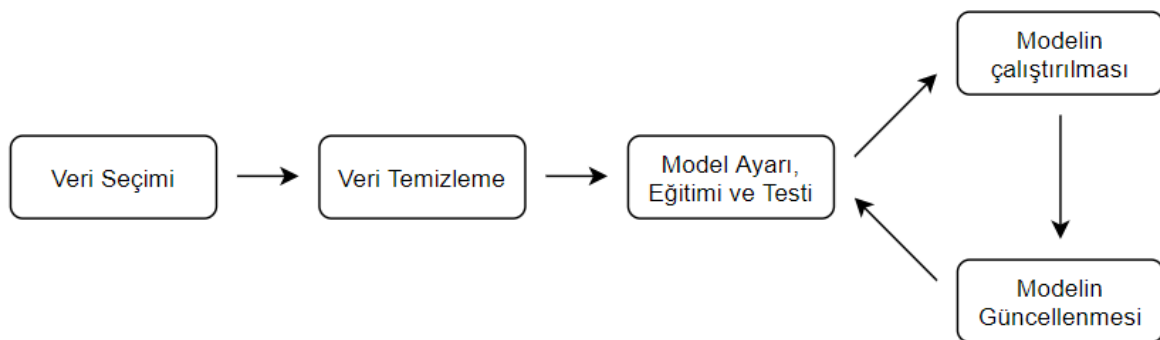


## 1.4. Makine Öğrenmesi Teknikleri

Çalışmada kullanılan yöntemlerin daha iyi ifade edilebilmesi amacıyla bu başlık altında makine öğrenmesi yöntemlerinin aşamaları, karşılaşılabilecek hata türleri, uyum sorunları ve öğrenmenin gerçekleşebilmesi için yaygın olarak kullanılmakta olan öğrenme tiplerine yer verilmiştir.

### 1.4.1. Makine Öğrenmesi Aşamaları

Makine öğrenmesi yönteminin uygulanması için takip edilebilecek aşamalar Şekil 1.2’de özetlenmiştir. İşlemler açısından makine öğrenmesi verilerin hazırlanması ile başlamaktadır. Bu aşamada verilerin doğruluğu, sayısı, seçimi gibi adımlar içerebileceği gibi makine öğrenmesi algoritmaları öğrenme yöntemlerine göre farklılık gösterebilmektedir. Eğitim verisi ana verinin içinden çekilen örnekler aracılığı ile oluşturulmaktadır. Genellikle veri setinin %70’i eğitim kalan %30’u ise test için kullanılmaktadır. Eğitim veri seti, algoritmanın öğrenme sırasında kullandığı maliyet fonksiyonunu minimize etmek için kullanılmaktadır (Zhang vd., 2020: 22). Modeller için seçilebilecek öğrenme çeşitlerine ve farklılıklarına devam eden bölümlerde yer verilmektedir. Seçilen eğitim tekniği ve parametre ayarları ile seçilen model çalıştırılır, gerekli görüldüğü durumlarda (model performansının iyileştirilebileceği zamanlarda) parametreler güncellenir ve model tekrar çalıştırılır. Model beklenen sonuçları verene veya daha fazla iyileşme sağlanamadığı ortaya çıkana kadar işlem tekrarlanarak devam ettirilir. Nihai seçim, belirlenmiş performans metriklerine göre en uygun sonucu veren model olacaktır. Model değerlendirme ve parametre ayarlama işlemleri çalışmanın 3. bölümünde daha detaylı olarak ele alınmaktadır.

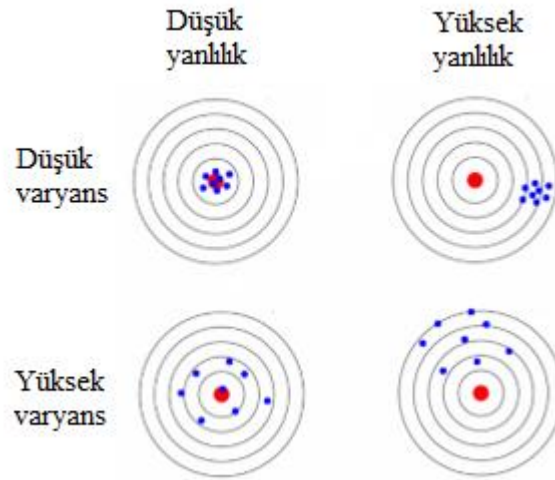


Şekil 1.2 Makine Öğrenmesi Aşamaları

### 1.4.2. Makine Öğrenmesinde Hata Tipleri

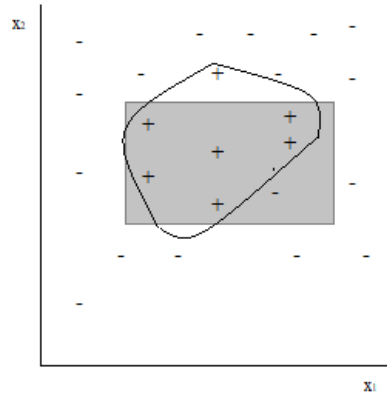
Yöntem fark etmeksizin hedef değer tahmini yapılan bir makine öğrenmesi yönteminde, gerçek değer ile tahmin edilen değer arasında oluşacak fark olarak da bilinen hata üç ana

sebebe dayanmaktadır. Bunlar yanlılık (bias), varyans ve gürültüdür. Eğitim seti yüksek performans gösterir, fakat test verisi veya gerçek veri üzerinde kötü bir performans gösteriyor ise o zaman modelde varyans hatası olduğu düşünülür. Gerçek değerler ile tahmin edilen değerler arasındaki dağınkılık varyanstır. Yanlılık hatası ise tahmin ile gerçek değerler arasında kalan uzaklıktır. Varyans ve yanlılık hataları Şekil 1.3'te örnek üzerinde gösterilmektedir. Araştırmacılar tarafından ulaşılmak istenen durum ise hedefte birleşen düşük varyans ve düşük yanlılık durumudur.



**Şekil 1.3 Varyans ve Yanlılık Hata Türleri Gösterimi (Singh, 2018:1'den alınıp yazar tarafından düzenlenmiştir)**

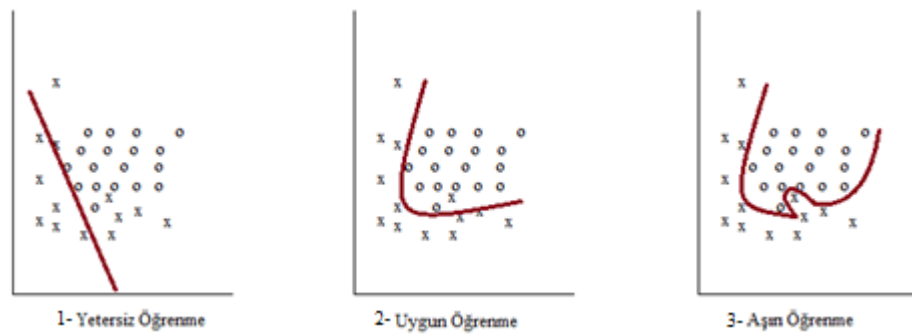
Gürültü ise veri setindeki hatalar veya aykırı değerler için kullanılan bir terimdir. Genellikle veri toplama aşamasında kullanılan cihazlar, aktarım veya ayrıştırma sırasında oluşan hatalar veya insan kaynaklı yanlışlıklar sebebi ile meydana gelmektedir. Şekil 1.4'te gürültü tipi hata gösterilmiştir. Normalde pozitif değerlerin kutu içinde bulunması gerekirken bir tane veri dışarıda olarak kaydedilmiştir, aynı şekilde kutu içinde de negatif değer tespit edilmiştir. Bu durumda öğrenme aşaması kare alanının dışında yeni çizilen alan ile gerçekleşeceği için ilerleyen aşamalarda gerçek veri veya test verisi ile tahmin yapacak olan algoritmanın yanlış tahmin yapması mümkündür.



Şekil 1.4 Gürültü Tipi Hata

### 1.4.3. Makine Öğrenmesinde Aşırı Uyum

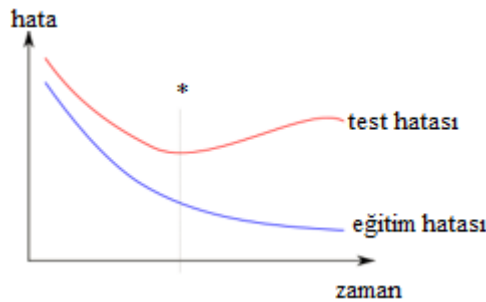
Aşırı öğrenme, modelin eğitim setinden öğrenme aşamasında başarı gösterdiği, fakat test aşamasında düşük performans gösterdiği durumlardır (Aksu ve Doğan, 2018: 90). Neden olduğu bazen net olmamakla beraber belirli üç ana sebebi vardır (Ying, 2019: 1). Çok fazla gürültü içeren bir öğrenme seti olması durumunda öğrenmenin gürültülü veri setine dayanması sebebi ile test veri setinde düşük sonuçlar vermesi bu durumlardan ilkidir. Varyans ve yanlılık arasındaki açmaz (Geman vd., 1992:1) ise bir diğer aşırı öğrenme nedenidir. Model daha iyi uyum sağlamaya çalışarak yüksek varyans gösterdiğinde yanlılık artabilmektedir. Son olarak yapay zekâ algoritmalarında girdiler birçok değerlendirme işlevi ile karşılaştırılır ve en iyi sonuç veren değerler seçilir. Ying'e (2019) göre bu durum modeli iyileştirmek yerine doğruluğu düşürmeye sebep olabilmektedir. Aşırı uyum gösteren bir modelin örnek gösterimi Şekil 1.5'te gösterilmiştir. Aşırı öğrenmenin gösterildiği 3. resimde veri setine göre fazla öğrenme gerçekleşmiştir. Test verisi veya gerçek veri üzerinde aynı şekilde iyi performans gösteremeyebilir. Uygun öğrenme ise 2. resimde gösterilmiştir. Burada yanlış gruplanan bir veri olsa dahi benzer bir test verisi üzerinde performans değerleri düşmeyecektir.



Şekil 1.5 Makine Öğrenmesi Uyum Tipleri (Singh, 2018:1'den alınıp yazar tarafından düzenlenmiştir)

Aşırı öğrenme durumunda kullanılacak çeşitli yöntemler bulunmaktadır. Erken durdurma yöntemi bunlardan biridir. Erken durdurma yöntemi ile doğrulama hatası ile test

hatası karşılaştırılır ve ikisinin birbirine en yakın olduğu noktada model durdurulur (Yao vd., 2007: 290). Şekil 1.6 hangi aşamada erken durdurmanın daha faydalı olabileceğini göstermektedir. Veri setini genişletmek önerilen bir diğer yöntemdir (Sun vd., 2014: 1891). Son olarak düzenleme önerilmektedir. Genellikle bir modelde çıktı birçok farklı girdi verisinden etkilenmektedir, fakat her girdi özelliği aynı önemde olmadığı gibi kimisi anlamlı bir katkıda dahi bulunmamaktadır. Düzenleme yöntemi ile katkısı olmayan değişkenlerin atılması veya anlamlı katkıda bulunmayanların katsayı değerinin düşürülmesi ile modellerde iyileştirme gerçekleştirilmektedir. Minimum maliyet üzerine kurulu bir amaç fonksiyonunda L1 düzenlemesi için bir doğrusal regresyon teorisi olan Lasso Regresyonunu kullanılmaktadır. Tüm değişkenlerin ağırlıklarınının mutlak değerler toplamları hesaplanarak fonksiyona uzaklık durumu incelenir. Buna göre uzaklığa göre ceza terimi elde edilmektedir. Maliyet işlemini en aza indirmek amacıyla bazı değişkenlerin ağırlıklarını azaltmak gerekecektir böylece en az etkisi olan değişkenler modelden çıkarılmaktadır. L2 düzenleme ise Öklid uzaklıklara odaklanmaktadır. Uzak olan değerlere daha düşük ağırlıklar vererek maliyet fonksiyonunu iyileştirmeye çalışmaktadır (James vd., 2017: 225).



Şekil 1.6 Aşırı Öğrenmeye Karşı Erken Durdurma Yöntemi (Chen, 2020:1'den alınıp yazar tarafından düzenlenmiştir)

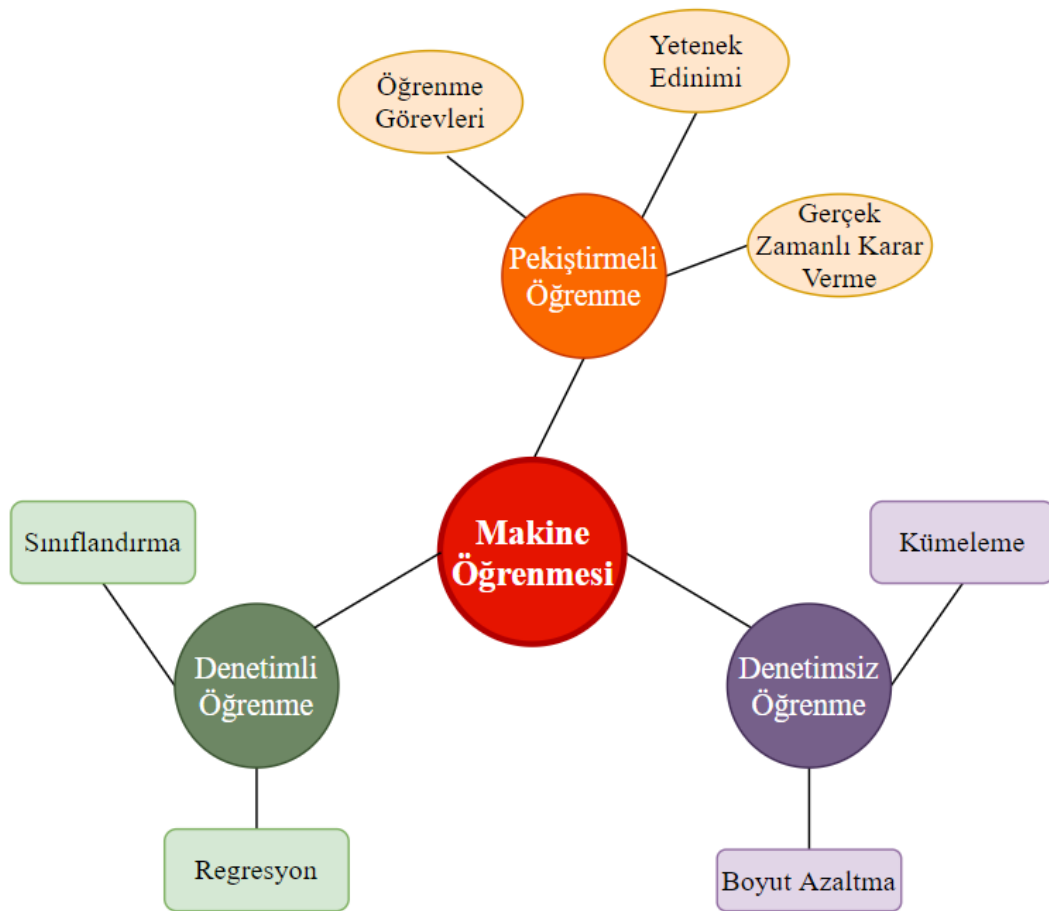
#### 1.4.4. Makine Öğrenmesinde Yetersiz Uyum

Eğer yanlılık durumu var ise modelin ilgili sınıfa ait çözümü karşılamadığı bir durum ortaya çıkar ve bu durum yetersiz öğrenme olarak adlandırılmaktadır (Alpaydın, 2014: 82). Performans metrikleri ile modeller değerlendirildiği için çok sık karşılaşılan bir durum değildir. Şekil 1.5'te gösterdiği gibi uygun model yetersiz öğrenme ve aşırı öğrenme arasında bir yerde olmalıdır (Van der Aalst vd., 2010: 87). Yetersiz öğrenme durumu ile karşılaşıldığında karar ağacı tipindeki öğrenciler için budama işlemi yapılması önerilmektedir (Ying, 2019: 2). Böylece anlamlı olmayan veya ilgisiz veriyi azaltarak sınıflandırıcının doğruluğunu arttırmak mümkündür. Bir başka yöntem ise öznelik mühendisliğine başvurulmaktadır. Genellikle girdi setinde çıktı üzerinde sınırlı ve hatta anlamsız değişkenler taşıyan değişkenler bulunmaktadır (Ying, 2019: 2). Bu durumda sadece çıktı üzerinde olumlu etkileri olan değişkenler seçilerek

kullanışsız olanların çıkarılması veya kullanışsız verilerin ağırlıklarının azaltılarak sonuç üzerindeki etkilerinin azaltılması önerilmektedir. Öznitelik mühendisliği yanlılık tipi hata üzerinde önemli bir değişiklik olmaksızın varyansı önemli ölçüde azaltmaktadır.

#### 1.4.5. Öğrenme Tipleri

Algoritmalar veri setlerine ve tercih edilen yöntemlere göre farklı şekillerde eğitilmektedir. Öğrenme tiplerini ise, denetimli öğrenme, denetimsiz öğrenme ve pekiştirmeli öğrenme olarak üç ana başlıkta incelemek mümkündür. Öğrenme tiplerinin kullanılabildikleri problem tipleri ve bu problem tiplerine ait örnekler Şekil 1.7’de gösterilmiştir.



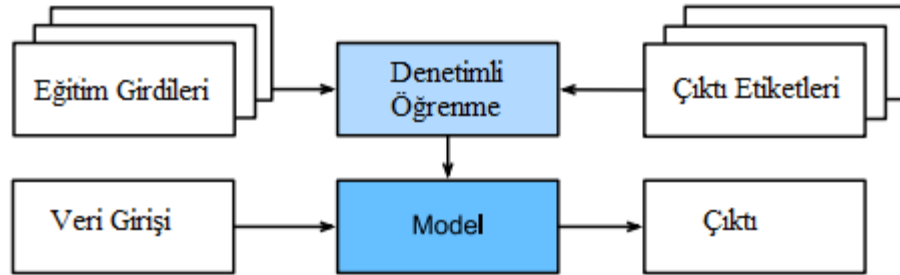
Şekil 1.7 Makine Öğrenmesi Öğrenme Türleri (Guru99, 2019:1'den alınıp yazar tarafından düzenlenmiştir)

##### 1.4.5.1. Denetimli Öğrenme

Denetimli öğrenme bir diğer adıyla gözetimli öğrenme, her bir girdi için etiketlenmiş sonuç gözlemleri ile öğrenme sürecidir. Bu şekilde eğitilen bir makine öğrenmesi yöntemi girdi verileri ile çıktı değerleri arasındaki ilişkiyi gözlemleyerek öğrenmeyi tamamlamaktadır (Russell ve Norvig, 2009: 695). Denetimli öğrenmeye çıktı değerleri (etiketleri) bulunan problemlerde sınıflandırma ve regresyon yapılırken başvurulmaktadır. Eğer veri iyi, kötü ve

orta veya var/yok gibi kategorik bir çıktı ise sınıflandırma, eğer ev fiyatları, gelir veya kilo gibi sürekli bir çıktı ise regresyon uygulanmaktadır (Bramer, 2016: 5).

Şekil 1.8'de gösterildiği gibi modele verilen girdilerin bir kısmı eğitim verisi olarak ayrılır. Bu veriler ile öğrenme süreci gerçekleştirilir, böylece veri seti ile ilgili genel kavramları rastgele oluşturulmuş daha küçük bir set üzerinden öğrenilir. Sonrasında veri setinin kalanı üzerinden eğitim bir model olarak kullanılır ve çıktı elde edilir.



Şekil 1.8 Makine Öğrenmesi Sürecinde Denetimli Öğrenme

Çıktı etiketleri  $y_i$  olmak üzere ve çıktıyı tahmin etmek üzere kullanılan eğitim girdileri  $x_i$  (genellikle birden çok değişken içerirler) ile matematiksel bir yapı oluşturularak tahmin aşaması gerçekleştirilmektedir. Yaygın olarak kullanılan doğrusal regresyon modelinin matematiksel gösterimi aşağıda yer almaktadır (1). Ağırlıklı girdi değişkenlerinin doğrusal bir kombinasyonu ile  $\hat{y}$  oluşturulur. Tahmini değer ( $\hat{y}$ ) amaca göre farklılık gösterir; regresyon modellerinde sürekli değer alabilirken, lojistik regresyon kullanılan bir sınıflandırma modelinde pozitif değerler alabilmektedir. Veriler aracılığı ile gerçekleştirilen öğrenme aslında parametrelerin öğrenimidir, bu kısım modeldeki belirsizlik kısmını oluşturmaktadır. Parametreler ise  $\theta$  ile gösterilmiştir.

$$\hat{y}_i = \sum_j \theta_j x_{ij} \quad (1)$$

Hem modeli eğitmek için hem de modelin eğitim verileri ile olan uyumunu ölçmek için amaç fonksiyonu tanımlamak gerekmektedir. Amaç fonksiyonları ise kayıp fonksiyonu (eğitim esnasında ortaya çıkan kayıplar) ve ihtiyaç duyulduğu takdirde düzenleme işlemini içerir.

$$amaç(\theta) = L(\theta) + \Omega(\theta) \quad (2)$$

$L$  kayıp fonksiyonunu,  $\Omega$  ise düzenleme fonksiyonunu gösterir. Eğitim verisine ait kayıp fonksiyonu eğitim sonuçları ile gerçek veri arasındaki farklıklardan oluşur. Sıklıkla kullanılan kayıp fonksiyonlarından ortalama hata kare (3) ve lojistik kayıp fonksiyonları (4) örnek olarak gösterilmiştir.

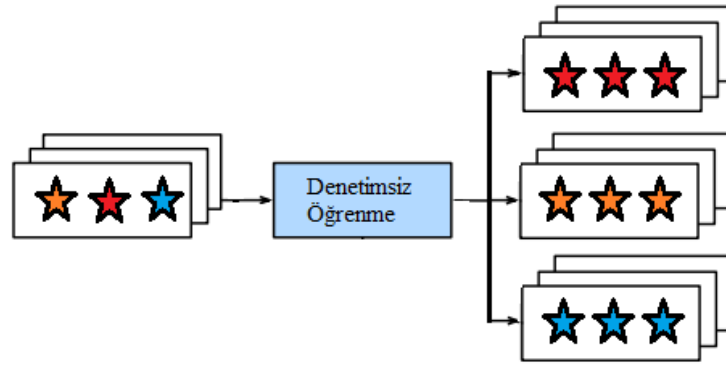
$$L(\theta) = \sum_i (y_i - \hat{y}_i)^2 \quad (3)$$

$$L(\theta) = \sum_i [y_i \ln(1 + e^{-\hat{y}_i}) + (1 - y_i) \ln(1 + e^{\hat{y}_i})] \quad (4)$$

Düzenleştirme terimi ile modelin karmaşıklaşmasının önüne geçmek için kullanılmakta ve böylece aşırı öğrenmenin önüne geçmektedir. Eğer kayıp fonksiyonuna göre her bir eğitim verisi mükemmel tahmin yaparsa yanlışlık ve varyans değerleri arasındaki uyum kaybolacak ve aşırı öğrenme gerçekleşecektir. Dengeyi korumak içinse düzenleştirme terimi modele belli bir ceza ekleyerek yapılan genellemeyi geliştirebilmektedir. Eğitim verisi gürültülü veriler içeren modellerde bu şekilde amaç fonksiyonu farklı sonuçlara uyumlu hale gelebilmekte ve düşük performans gösterebilmektedir.

#### 1.4.5.2. Denetimsiz Öğrenme

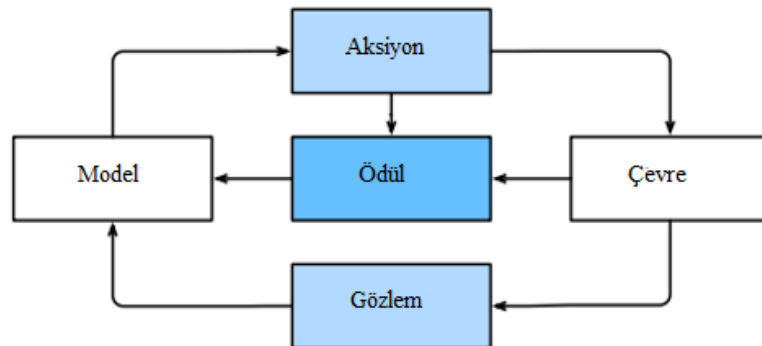
Herhangi bir niteliğe dayandırılmamış, etiketlenmemiş veya diğer bir değişle veri setinde çıktı içermeyen veriler için denetimsiz öğrenme kullanılmaktadır (Bramer, 2016: 5). Denetimsiz öğrenmenin (gözetimsiz öğrenme) çalışma mantığı ise algoritmanın kendi kendine keşif yaparak veri içindeki örüntüleri bulmasıdır (Skiena, 2017: 372). Veri seti içerisindeki düzen ve düzensizlikler veri keşfi ile bulunabilmekte ve dağılım kestirimleri ile gruplanabilmektedir (Alpaydın, 2017: 8). Şekil 1.9'da gösterildiği gibi farklı renklerde yıldızlardan oluşan veri seti denetimsiz öğrenme ile benzerliklerin keşfi özelliği sayesinde çıktı verisi olmaksızın, aynı renkteki yıldızlar olarak ayrıştırılabilmektedir. Denetimsiz öğrenmeye sadece çıktı verisi olmadığında değil ayrıca boyut azaltma ve kümeleme tekniklerinin kullanımı için de başvurulmaktadır (Kubat, 2017: 273). Bir başka kullanım alanı ise eksik verilerin tahmini olarak gösterilmektedir (Kubat, 2017: 276). Eksik veri setlerinde gruplama yaparak ilgili verinin hangi gruba daha yakın olduğunu tahmin ederek eksik verileri tamamlayabilme özelliği bulunmaktadır. Farklı bir uygulama alanı ise görüntü sıkıştırma kullanılmasıdır. Bu kullanımında ise benzer renkteki imge noktalarını aynı öbek altında toplar böylece çok tonlamalı detaylı görüntülerin yer tutma alanı görüntü boyutları küçültülür ve saklanmasını kolaylaştırır (Alpaydın, 2017: 9). Denetimli öğrenme yöntemlerine göre daha az tercih edilmesinin sebebi ne kadar iyi sonuç verdiğini gösteren bir doğru yanıtı sahip olmamasıdır (James vd., 2017: 374). Fakat bu durum yöntemi daha işlevsiz olduğunu göstermemektedir, belli bir ölçüm kriterine sahip olmasa dahi veri elde etmenin güç veya pahalı olduğu durumlarda çok kıymetli bir yöntemdir. Sıklıkla tavsiye sistemlerinde kullanılmaktadır. Geçmiş alışverişlerine göre müşterilere ürün ve promosyon önerileri yapan firmalar, izlenen film ve dizilere göre yeni önerilerde bulunan Netflix gibi öneri sistemleri denetimsiz öğrenme temeli ile çalışmaktadırlar.



Şekil 1.9 Denetimsiz Öğrenme Çalışma Prensibi

### 1.4.5.3. Pekiştirmeli (Takviyeli) Öğrenme

Pekiştirmeli öğrenmede çıktı genellikle bir eylem dizisinden oluşmaktadır. Oyun oynamak bu duruma örnek olarak verilmektedir. Nedeni ise oyunların tek belli bir eylemi değil yeni durumlara özgü eylem dizileri gerektirmesidir. Geçmişteki örneklerden en iyi politikaları belirleyip karşılaşılan yeni durumlarda bilinen deneyimlere başvurularak başarılı bir politika seçimi yapılması bu yönetime özgü bir durumdur (Alpaydın, 2017: 9). Şekil 1.10'da gösterildiği gibi model karşılaşılan durum hakkında yetersiz bilgiye sahip ise neden sonuç arasında bağ kurmak amacıyla gözlem yapar, alacağı aksiyon ile ödülü en yüksek seviyeye çıkaracak şekilde karar almaya çalışır. Aksiyon sonucu alınan ödül ile aksiyon arasında bağ kurularak bu durum deneyim haline getirilir ve yeni durum için tekrar gözlem durumuna geçer. Pekiştirmeli öğrenme yöntemi problemlerdeki ödülü sayısallaştırır ve bu durumu maksimize etmeye çalışır, bu aşamada ise hangi durumun hangi eylemle eşleşeceğini keşfederek öğrenmeyi gerçekleştirir (Sutton ve Barto, 2011: 2). Günümüzde sürücüsüz otomobil kullanımında, oyun oynamada, doğal dil işlemede, finansal işlemleri gerçekleştirmede pekiştirmeli öğrenmeye başvurulabilmektedir.



Şekil 1.10 Denetimsiz Öğrenme Çalışma Prensibi

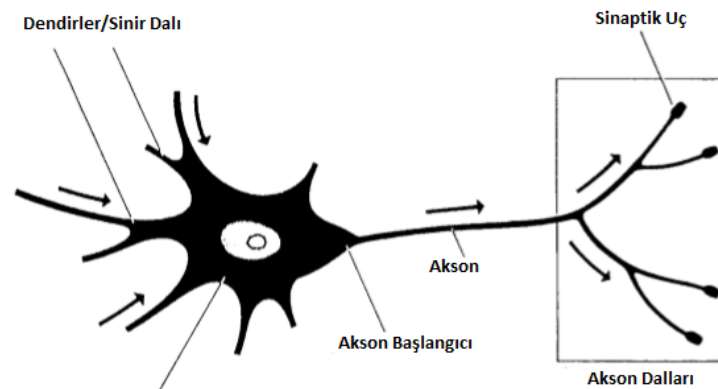


#### 1.4.6. Çalışmada Kullanılan Yöntemler

Çalışmanın amacını oluşturan uçuş gecikmeleri yapısı itibariyle ikili sonuca sahiptir. Şöyle ki uçaklar zamanında veya gecikmeli olarak kalkacaklardır. Böyle bir durumda sınıflandırma problemine uygun makine öğrenmesi yöntemleri seçilmelidir. Yöntem olarak makine öğrenmesi teknikleri içinden yapay sinir ağları, rastgele ormanlar, extreme boosting (ekstrem gradyan artırma -XGBoost), light gradient boosting machine (hafif gradyan artırma - LightGBM) ve category boosting (kategori artırma -CatBoost) tercih edilmiştir. Yapay zekâ kavramının ilk ortaya koyulduğu zamanlardan beri bilinen yapay sinir ağları yöntemi (Bostrom, 2020: 24) diğer yöntemlere göre daha eski ve bilinir olması sebebiyle çalışmada kullanılmıştır. Rastgele ormanlar diğer yöntemlerin temelinde bulunan topluluk öğrenme ve karar ağaçları ile geliştirilen ilk yöntem olması sebebiyle daha eski bir yöntemin sonuçlarını inceleyebilme imkanı sağlayacağı için seçilmiştir. XGBoost, LightGBM ve CatBoost yöntemleri ise diğer yöntemlere göre çok daha güncel yöntemlerdir ve sıklıkla birbirleriyle literatürde birbirlerine göre üstünlükleri de karşılaştırılmaktadır (Daoud, 2019: 6; Kimnander, 2020: 4; Punmiya ve Choe, 2019: 2326; Sahin, 2020: 1) sebebi ile bu çalışmada da performans karşılaştırma imkanı olması amacıyla tercih edilmiştir. Bu başlık altında birlikte yöntemlerin tanıtımı ile detaylı bilgi verilecektir.

##### 1.4.6.1. Yapay Sinir Ağları (Artificial Neural Network)

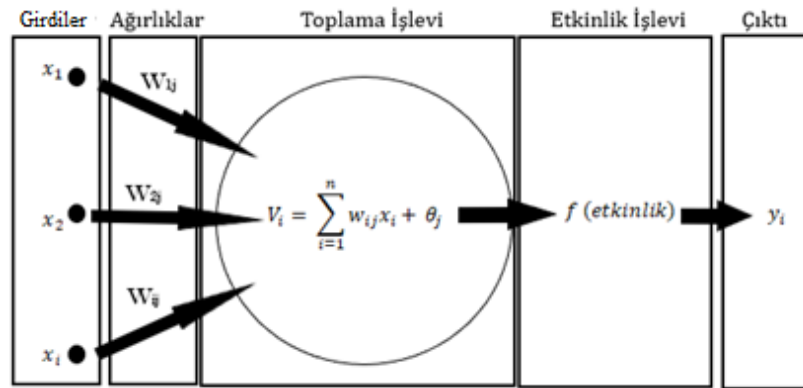
Makine öğrenmesinin alt dallarından olan yapay sinir ağları yöntemi, öğrenme işleminin hayvanlardaki biyolojik sinir ağlarının taklit edilerek yapılması fikrine dayanmaktadır. Fakat sadece taklit etme ile sınırlı kalmamıştır; yapay sinir ağları bu taklidi kabaca yapabiliyor olmasına rağmen karışık, doğrusal olmayan, stokastik problemler gibi çeşitli problem türlerine çözüm sunabilmektedir (Graupe, 2007: 2). Nöronlar veya sinir hücreleri vücutta tek tip değildir, fakat en temel hali ile sinir hücresi Şekil 1.11'de gösterilmiştir. İlgili hücre memeli omuriliği



Şekil 1.11 Memeli Sinir Hücresi (Arbib, 1989: 52)

hücresinin temsili görünümüdür. Sinir hücrelerinin iletişimi dendiritlerden başlayarak sinaptik uca doğru iletim şeklindedir, fakat bu iletim tersine de olabilmektedir (Arbib, 2003: 4).

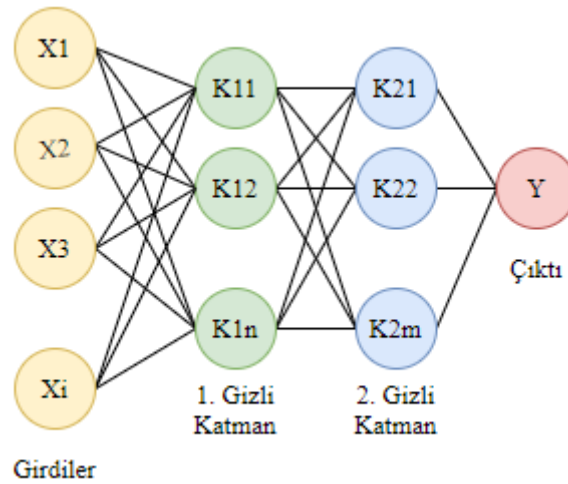
Yapay sinir ağları ise aynı mantıkla birden çok işlem elemanının birbirine bağlanması ile elde edilir. Bu ağlar insan beynine benzer olarak öğrenme, hatırlama ve genelleme yeteneklerine sahiptirler. Yapay sinir ağlarında gerçek bir sinir hücresine karşılık olarak kullanılan terim düğüm olarak karşılık bulmaktadır. Biyolojik sinir sisteminin kaba bir taklidi olsa da gerçek hücrelerin yerine getirmekte olduğu işlevleri yerine getirmeye çalışmaktadır. Şekil 1.12’de ise düğümün yapısına yer verilmiştir.



Şekil 1.12 Yapay Sinir Düğümü

Şekil 1.11 ve 1.12 yapısal ve sistematik olarak birbirine oldukça benzemektedir. Şekil 1.11’de dendiritler gelen çeşitli bilgiyi hücre gövdesine iletip oradan ilgili tepkiyi sinaptik uçlarla çıktı olarak iletmektedir. Şekil 1.12’deki düğümde girdiler  $x_i$  sembolü ile temsil edilmiştir. Girdilerin her biri ağırlık değeri olan  $w$  ile çarpılır ve hücrenin kendisinin sahip olduğu  $\theta_j$  (bias) değeri ile toplanarak sonucu oluşturmak için düğümün sahip olduğu etkinlik işlevine göre işlem yapılır ve çıktı elde edilir.

Genel bir yapay sinir ağı görünümü ise Şekil 1.13’te gösterilmektedir. Burada  $X$  ile gösterilen ilk değerler modele girdi olarak verilen veri setinin değişkenlerini temsil etmektedir. Örneğin,  $X1$  veri setinin ilk değişkenini temsil etmektedir. Girdi katmanı ile gizli katman arasında verilerde değişiklik olmaksızın iletim gerçekleştirilir. Gizli katman Şekil 1.13’te gösterilmiş olan düğümlerden oluşmaktadır. Temsili olarak çizilen bu YSA iki gizli katmana sahiptir. Katman sayısı kullanılacak probleme ve veri setine göre farklı seçilebilecek bir parametredir. Girdi verilerinden çıktıya doğru bir akış gerçekleşmektedir. İleri beslemeli bir YSA sadece girdilerden çıktıya doğru bir akışta işlem gerçekleştirirken tekrarlayan bir YSA (Recurrent Neural Networks) modelinde en az bir düğümün çıktısı kendine ya da bir başka düğüme girdi verisi olarak iletilmektedir.



Şekil 1.13 Yapay Sinir Ağları

Denetimli öğrenme yöntemi ile geliştirilen bir YSA modelinde öncelikle düğüm içindeki ağırlıklar rastgele belirlenmektedir. Ardından geri yayılım algoritmasına başvurulur ve gerçek çıktı değerleri ile tahmin değerleri karşılaştırılır ve aradaki farka göre gizli katmanlardaki düğüm ağırlıkları güncellenmektedir. En düşük hata elde edilinceye veya model için belirlenen sonlanma durumu gerçekleşinceye kadar devam ederek en iyiyi aramaya devam edilmektedir.

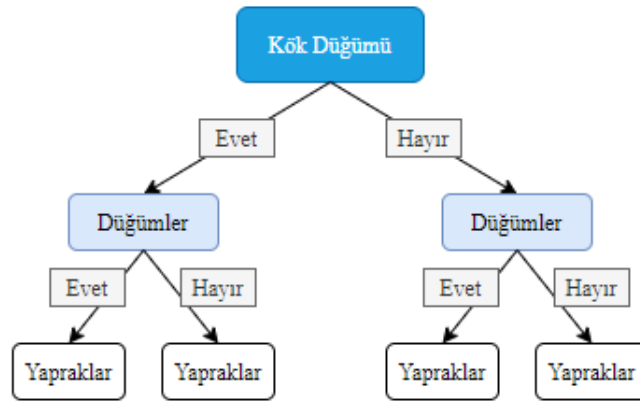
#### 1.4.6.2. Karar Ağaçları

Karar ağaçları, makine öğrenmesi teknikleri içinde basit ama aynı zamanda son derece başarılı yöntemlerden biri olarak bilinmektedir (Russell ve Norvig, 2009: 697). Problem çözümü veya veri inceleme aşamasında bilgi elde etme amacıyla karar ağaçları kullanılabilir. Karar ağaçları tümevarım yöntemi mantığı temelinde çalışmaktadır (Yuan ve Shaw, 1995: 125). Aynı zamanda hem regresyon hem de sınıflandırma için kullanılmakta olsa da bu modeller arasında bazı hesaplama farklılıkları mevcuttur (Loh, 2011: 14). Büyük miktardaki girdiyi hedef değişkene ulaşmak için küçük gruplara bölerek ilerleyen model yapısına sahiptir. Başarılı bölümlenme yapıldığında elde edilen grup diğer veri bütününe göre heterojen ama kendi grubu içinde homojen bir yapıya sahip olmaktadır. Bu yapısı itibarıyla analiz etmesi daha güç olan ham veri üzerinde anlamlı bilgiler çıkarmayı amaçlarken kullanılabilir (Kantardzic, 2003: 140).

Karar ağacı oluşturma aşamasında birçok farklı algoritma kullanılabilir. Bu algoritmalar da sonrasında farklılaştırılarak farklı yöntemlere de dönüşebilmektedir. Kullanılan farklı algoritmalar: CHAID, Exhaustive CHAID, CRT, ID3, C4.5 (J48), MARS, QUEST, C5.0, SLIQ, SPRINT, CRUISE, GUIDE, AID, THAID olarak örneklendirilebilir (Loh, 2011: 15). Birbirlerine göre çeşitli üstünlükleri olmasının yanı sıra farklı özelliklere de sahiptirler. Örneğin

AID, regresyon için geliştirilen ilk karar ağacı algoritması olmuştur (Morgan ve Sonquist, 1963: 415).

İnsan düşünme tarzına çok yakın olması ile ünlenmiş olan yöntemin ana çalışma prensibi şöyledir: 15:30'da gerçekleşecek toplantı için saatine bakan kişinin “saat daha 15:10 biraz daha beklemeliyim” şeklinde karar vermesi en basit karar ağacı modelini oluşturur. Daha karmaşık modellerde ise ağaç oluşturulurken entropi değerini yani rastgelelik derecesini azaltan seçimler yapmak amaçlanır ve her adımda daha iyi bilgi kazanımı elde edilmeye çalışılmaktadır. Entropi denklemi  $H = -\sum p(x) \log p(x)$  olmak üzere  $p(x)$  sınıflandırma yapılırken kullanılan sınıfın yüzdesini,  $H$  ise entropi değerini belirlemektedir (Shannon, 1948: 393). Bu durum her adımda veya düğümde hata fonksiyonuna başvurulurken en düşük değeri veren düğümün seçimine dayanır (Oded ve Lior, 2005: 165). Kök düğümü ham verinin olduğu yerdir. İç karar düğümleri ise soruların sorulduğu ve test sonuçlarına göre karar ağacının hangi yöne gideceğinin belirlendiği aşamadır. Dallar belirlenen sorulara verilen cevapları temsil eder. Uç yapraklar ise kategorinin bulunduğu sınıfın etiketidir (Rokach ve Maimon, 2014: 151). Örnek karar ağacı görünümü Şekil 1.14'te gösterilmiştir.



Şekil 1.14 Karar Ağacı Gelişimi

Entropi değeri en aza indirilmeye çalışılırken karar ağacındaki bilgi kazanımı ise artırılmaya çalışılır. Bilgi kazancı için ise kazanç fonksiyonuna başvurulur. Ham veri  $S$  ile gösterilirken ilk bölüm  $D$  ham verinin bölünmüş bir parçasını gösterir. Formülde  $D$  bölümünün  $S$  için bilgi kazancı, alt küme olan her bir  $V$  için bölünmeden önce elde olan entropi ile alt kümelerin entropi değeri arasındaki farktan elde edilmiştir (5).

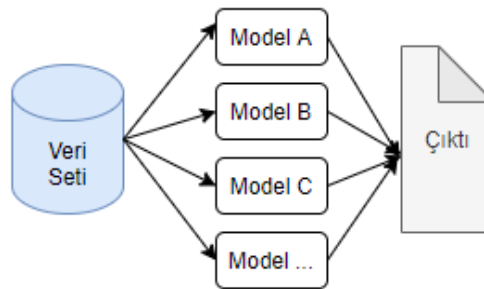
$$\text{Bilgi Kazancı}(S, D) = H(S) - \sum_{V \in D} \frac{|V|}{|S|} H(V) \quad (5)$$

Karar ağaçları genellikle kullanımlarının kolay ve anlaşılabilir olması sebebi ile tercih edilmektedir. Görselleştirmenin mümkün olması bir başka önemli avantajını oluşturur. Veri hazırlama ve ön işleme aşaması göreceli olarak kısadır. Parametrik test önkoşullarının olmayışı da bu işlemleri azaltmaktadır. Karar ağaçları aynı zamanda eksik veri içeren veri setlerinde de

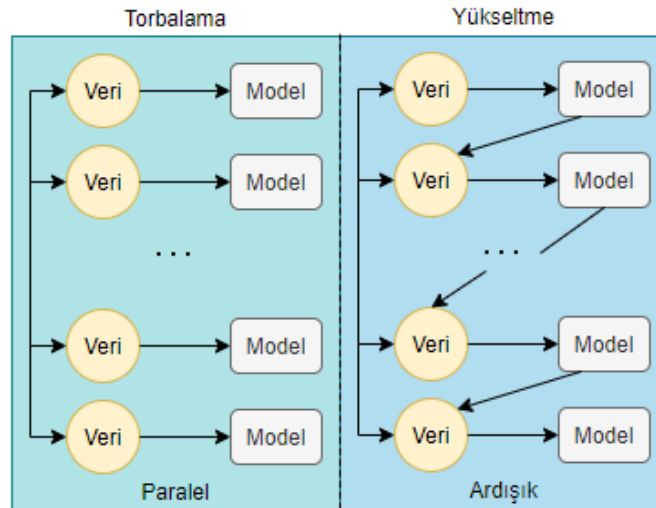
kullanılabilmektedir. Dezavantajları ise veri setindeki küçük bir değişikliğin tüm ağaç yapısında önemli bir değişikliğe yol açabilmesi, veri setinin eğitiminin göreceli olarak uzun sürmesi, zaman zaman güç hesaplamalar gerektirmesi olarak gösterilmektedir (Gareth vd., 2013: 315).

### 1.4.6.3. Topluluk Öğrenmesi

Topluluk öğrenmesi basit mantıkla bakıldığında bir konuda birden çok kişinin fikrini sorup en doğru kararı alma temeline dayanmaktadır. Temel bir öğrenci yerine kolektif öğrenmeye başvurulması sebebiyle daha doğru kararlar alabilme yeteneğine sahip olmaktadır. İndirgenmesi mümkün olmayan gürültü tipi hata hariç varyans ve yanlılık faktörlerini azaltmaya yardımcı olmaktadır. Kullanım amacı hatayı azaltmak olan bu yöntem daha iyi bir tahminci olmasını birçok zayıf tahminci barındırmasına borçludur (Suchetana vd., 2017: 251). Topluluk öğrenme yönteminin yapısı görsel olarak Şekil 1.15'te gösterilmektedir. Torbalama ve yükseltme yöntemleri topluluk öğrenmesinde kullanılan iki ana yöntemi oluşturmaktadır. İki yöntem arasındaki en önemli fark seçilen eğitim setidir. Torbalama (Bagging) yöntemi Breiman (1996) tarafından geliştirilmiştir. Yöntem her çalıştığında eğitim seti içinden rastgele bir alt eğitim seti oluşturur ve bu rastgele set ile temel öğrenci set yeniden düzenlenir. Veri setinde bulunan ve eğitim için ayrılmış olan  $n$  örnek var ise yerine  $n$  örnekle eğitim seti rastgele seçimle üretilerek konur (Dietterich, 2000: 5). Eğitim paralel bir şekilde ilerler. Yükseltme (Boosting) yönteminde veri setinden seçilen rastgele eğitim seti ile modeller ardışık bir şekilde eğitilir, böylece her model bir önceki modelden belirli bir girdi alır ve buna göre örneklerin ağırlıklarını değiştirerek devam eder. Yükseltme yöntemi başlangıç aşamasında eşit ağırlığa sahip veri setine ağırlık değerleri vererek geliştirilen ağaçlardan çıkarımlar yapmaya dayanmaktadır (Schapire, 1990: 197). Yükseltme çalışmaya devam ettikçe yanlış sınıflandırılan gözlemlere daha fazla ağırlık verilmesini böylece bir sonraki adımda daha az görülebilir olmasını sağlamaktadır (Freund ve Schapire, 1996: 149). Şekil 1.16 üzerinde seçim yöntemlerinin farklılıkları gösterilmektedir.



Şekil 1.15 Topluluk Öğrenme Yapısı



**Şekil 1.16 Torbalama ve Yükseltme Yöntemleri**

En sık başvurulan iki topluluk öğrenimi yöntemi olan Torbalama ve Yükseltme yöntemleri dışında kullanılan teknikler de vardır. Bunlar Maksimum Oylama (Maximum Voting), Ortalama (Averaging), Ağırlıklı Ortalama (Weighted Averaging), Yığıma (Stacking) ve Harmanlama (Blending) teknikleridir. Sınıflandırma problemleri için kullanılan Maksimum Oylama tekniği çoğunluğun verdiği kararların daha güvenilir olduğu temeline dayanmaktadır (Bulut, 2017: 70). Ortalama ise regresyon problemlerinde tahmin yapmak için kullanıldığı gibi sınıflandırma problemleri olasılıkları değerlendirilirken de kullanılmaktadır. Aynı temele dayalı Ağırlıklı Ortalama yöntemi ise farklı örneklemelere farklı ağırlıklar vererek topluluk öğrenmeyi gerçekleştirmek üzere kullanılmaktadır. Yığıma modeli iki veya daha fazla modeli bir araya getirerek tahmin yapan hibrit bir tekniktir.

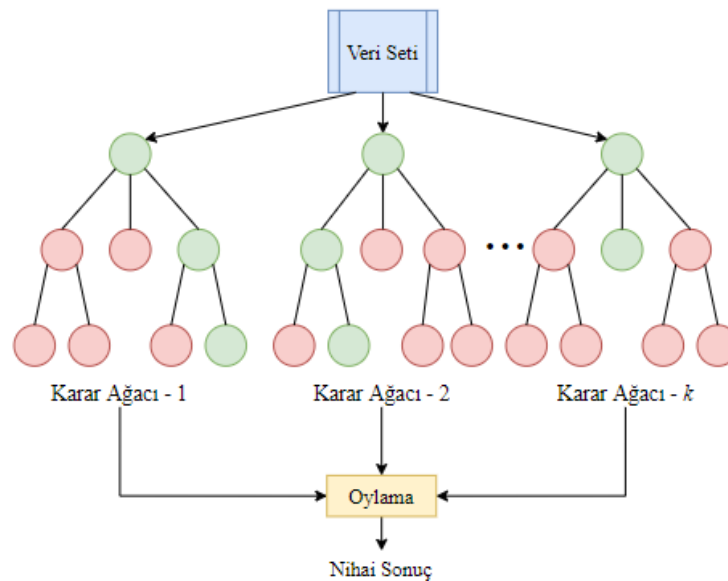
#### 1.4.6.4. Rastgele Ormanlar

Karar ağaçlarının birleştirilebileceği (birçok ağacın bir arada kullanılabilmesi) düşüncesi ile olarak Kwok ve Carter (1990) tarafından ortaya koyulmuştur. Rastgele ormanlar yöntemi olarak ortaya çıkışı ise Amit ve German'ın (Amit ve German, 1997: 1545) yapmış olduğu çalışmayı temel alan Leo Breiman (Breiman, 2001: 5) tarafından yapılan çalışmaya dayanmaktadır. Breiman'ın (2001) ortaya koyduğu haliyle rastgele ormanlar yöntemi hem sınıflandırma hem de regresyon problemlerinde kullanılabilir. Bu avantajının yanı sıra göreceli olarak hızlı eğitilmesi ve tahmin sonuçlarına hızlı ulaşabilmesi, birkaç parametre ayarı ile çalışabilmesi, hem kategorik hem sürekli veriler ile birlikte çalışabilmesi, hata tahmini yapabilmesi, yüksek boyutlu problemlerde kullanılabilir olması ve paralel ağaçların üretimini kolayca yapması gibi avantajlara sahiptir (Cutler vd., 2011, 157). Rastgele ormanlar yönteminin ana fikri, çok sayıda tahminleyici ağaç üretimi ve sonrasında üretilen ağaçlar içinden rastgele seçilerek alt topluluklar oluşturmaya dayanmaktadır (Breiman, 2001: 6). Burada temel olarak

ortaya koyulan fikir torbalama yönteminin kendisidir diğer bir deyişle karar ağaçlarının torbalama tekniği ile birlikte çalıştırılması fikri rastgele ormanlar yönteminin kendisini oluşturmaktadır.

Modelin en önemli kısmı rastgele ağaç oluştururken belli bir iyileştirmeyi aramaması tamamen rastgele bir şekilde ilerlemesidir. Oluşan rastgele alt kümeler içinde en iyi sonuç veren adım kullanılır. Böylece model çeşitlilikle ilerlemiş belli bir kalıba sıkışmamış olmaktadır. Normal karar ağacı adımlarında en iyi eşik değerleri aranmaktayken modelin en önemli farkını bu adım oluşturmuş ve alt modelleri en iyileme amacı yerine rastgele bölümler yapıp içlerinden en iyisini seçmek tercih edilmiştir (Salford Systems, 2014: 15). Bu duruma topluluk öğrenmesi yöntemlerinde olduğu gibi kalabalıkların bilgeliği denmektedir; tek ağaç ile yapılabilecek tahminden daha yerinde olmaktadır (Clemen ve Winkler, 1986: 39; Armstrong, 2001: 417). Karar ağaçlarında belirlenen hedefe göre kurallar oluşturulduğu belirtilmiştir. Bu kurallar dahilinde bir olayın gerçekleştirilip gerçekleştirilmeyeceği kararını vermek, birçok rastgele alt dal oluşturup bunlar arasından en iyi sonuç verenleri birleştirmeye göre daha kısa süre almaktadır. Süre konundaki dezavantajı olsa dahi aşırı öğrenme problemini yenerek destek vektör makineleri, yapay sinir ağları ve ayırma (diskriminant) analiz kadar iyi sonuçlar da vermektedir (Ali vd., 2012: 277).

Şekil 1.17’de  $k$  tane karar ağacından oluşan rastgele ormanlar yöntemi bileşenleri gösterilmiştir. Orman oluşturulurken sınıflandırıcı problemleri için  $x$  değişkenler,  $\theta_k$  ise rastgele vektörü olmak üzere ağaçlar  $\{h(x, \theta_k), k = 1, \dots\}$  şeklinde gösterilmektedir (Breiman, 2001: 6). Ağaç oluşumunun asıl farklı noktası özelliklerin alt kümelerinin rastgele seçiliyor olmasıdır.



Şekil 1.17 Rastgele Orman Yöntemi

Modelin yapısı incelendiğinde ise dallanma kriteri için gini indeksi kullanıldığı görülmektedir. Modelin kazanımı hesaplanırken  $T$  olarak belirlenen bir eğitim kümesinde  $C_i$  rastgele seçilen kümeyi temsil eder.  $f(C_i, T)/|T|$  ise seçilen örneğin  $C_i$  sınıfına ait olma olasılığıdır. Gini indeksi şöyle hesaplanır (Pal, 2005, s. 218):

$$\sum_{j \neq i} \sum (f(C_i, T)/|T|)(f(C_{ij}, T)/|T|) \quad (6)$$

Hesaplanan Gini indeks değeri büyüdüğü oranda sınıf daha heterojen hale gelir, azaldıkça ise sınıf kendi içinde homojendir. Başarılı bir dallanma işleminde alt düğüm Gini indeksinin bir üst düğüm Gini indeks değerinden daha düşük olması gerekmektedir. İndeks değeri sıfır olduğunda ağaç dallanması sonlanır. Bu aynı zamanda yaprak düğümlerinde bir sınıf kalmış olduğunu gösterir (Watts vd., 2011: 68). Ağaç geliştirildikten sonra ise girdiler üretilen her ağaca yerleştirilir ve bu ağaçlar oylanır. En yüksek oyu alan ağaç sınıfa atanır (Liaw ve Wiener., 2002: 20).

#### 1.4.6.5. Gradyan Yükseltme (Gradient Boosting Machines)

Regresyon, sınıflandırmaya ve sıralama problemlerinde kullanılabilen bir makine öğrenmesi yöntemidir (Cheng, 2016: 2). İlk uygulama 1996 yılında Robert Schapire ve Yoav Freund tarafından geliştirilen algoritma ile gerçekleştirilmiştir. Bu algoritma ise Adaptive Boosting (Adaboost) ismi ile tanınmaktadır (Freud ve Schapire, 1996: 148). Bu teknik bir önceki tahminleyicinin hatasına göre öğrenme gerçekleştirir. Öğrenme tekniği ceza sistemine dayanmaktadır. Ceza tekniği algoritmanın sonuçlarını bozan değişkenlere negatif ağırlık değerleri verilerek gerçekleştirilmektedir. Adaboost algoritması maliyet fonksiyonunu en aza indirme için kullanılmıştır ve başarılı sonuç alınmıştır (Breiman, 1998: 801). Bu yaklaşım doğrusal bir şekilde ilerleyerek güçlü sınıflandırıcıya ulaşmayı amaçlamaktadır. Geliştiricileri Adaboost algoritmasının keşfi ile Gödel ödülünü almaya da hak kazanmışlardır (Ozker, 2019: 1). Ardından bir başka gelişme gradyan azalma yöntemi kullanılması ile olmuştur. Sonrasında ise daha genel kullanıma uyarlanmış ve birçok ceza fonksiyonunda kullanılır hale getirilmiştir (Friedman, 2001: 1189).

Şekil 1.15'te ardışık çalışma prensibi gösterilmektedir. Bu teknik sayesinde önceki aşamadan alınan sonuçlara göre hatalar azaltılır. Diğer bir deyişle model önceki işlemlerden edindiği deneyim ile hataları azaltarak ilerlemektedir (Grover, 2017: 1).



Genellikle maliyet fonksiyonunu en aza indirmeye çalışan modeller üzerinde kullanıldığı için örnek olarak ortalama karekök hatasını (OKH) en aza indirecek bir model ile gösterilmek üzere:

$$y_i = i. \text{ hedef deęer,}$$

$\hat{y}_i$  ise  $i$ . tahmin,  $L(y_i - \hat{y}_i)$  maliyet fonksiyonu olmak üzere,

$$\text{Maliyet} = OKH = \sum (y_i - \hat{y}_i)^2 \quad (7)$$

$\alpha$  öğrenme oranı ve  $\sum (y_i - \hat{y}_i)$  kalanların toplamı olmak üzere,

$$\hat{y}_i = \hat{y}_i + \alpha * \delta \sum \frac{(y_i - \hat{y}_i)^2}{\delta \hat{y}_i} \quad (8)$$

$$\hat{y}_i = \hat{y}_i - \alpha * 2 * \sum (y_i - \hat{y}_i) \quad (9)$$

Maliyet fonksiyonunu (7) en aza indirmek için  $\alpha$  öğrenme oranı olmak üzere tahminler güncellenmektedir (8-9). Özetle kalanları en aza veya sıfıra indirme amacıyla tahminler gerçekleştirilmekte ve en düşük tahmin modelinin ne zaman elde edildiği araştırılmaktadır. Burada gösterilen aşamalar ile yapılacak olan tahminin ortalama karekök hatasına göre en aza indirgenmiş düzeyde olacağı dolayısıyla gerçek değerlere en yakın şekilde ayarlanmış olacağı öngörülmektedir.

Genel model gösteriminde ise  $y(i)$  çıktı yani tahmin edilmek istenen değer,  $x(i)$  ise tahmin etmek için kullanılan girdiler (değişkenler) olmak üzere,  $N$  tane örnek ile kayıp fonksiyonu çalıştırılır (10). Bu fonksiyonda  $y_i$  gerçek değerler gamma ( $\gamma$ ) ise tahminleri temsil etmektedir. Bu fonksiyon veri setindeki her bir girdi ile çalıştırılmaktadır. Bu fonksiyonda *argmin* elde edilen toplam değerini en aza indirgeyerek uygun tahmin değerlerini bulmaya yardım etmektedir.

$$f_0(x) = \underset{\gamma}{\text{argmin}} \sum_{i=1}^N L(y_i, \gamma) \quad (10)$$

$M$  tane karar ağacı kullanılarak  $i = 1, 2, \dots, N$  her bir örnek değer için kayıp fonksiyonunun  $x$  değişkenine göre türevi alınmakta ve hata değeri olduğu için -1 ile çarpılmaktadır.

$$r_{im} = - \left[ \frac{\partial L(y_i, f(x_i))}{\partial f(x_i)} \right]_{f=f_{m-1}} \quad (11)$$

Böylece her örnek için kayıp fonksiyonun  $x$  değerine göre türevi hesaplanır ve -1 ile de çarpıldığında gerçek değer tahmini elde edilmiş olmaktadır. Temel öğreniciyi  $h_m(x)$  ile uyumlandırmalı yani  $\{(x_i, r_{im})_{i=1}^n$  değişkenler ve hata değerleri ile eğitim seti eğitilmektedir.

Sonrasında ise  $\gamma_m$  çarpanı tek taraflı optimizasyon problemi ile belirlenmelidir. Burada satır arama yaklaşımı kullanılır. Hedef fonksiyonun azaltılacağı bir iniş yönü gradyan inişi ile aranmaktadır.

$$\gamma_m = \underset{\gamma}{\operatorname{argmin}} \sum_{i=1}^n L(y_i, F_{m-1}(x_i) + \gamma h_m(x_i)) \quad (12)$$

Model eldekilere göre güncellenmelidir (13).

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x) \quad (13)$$

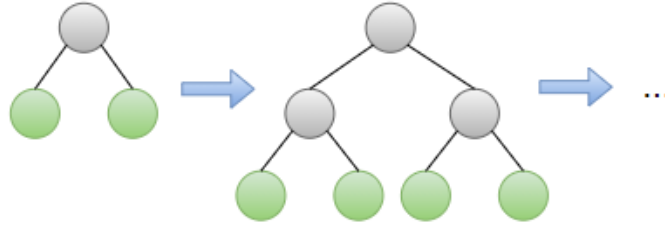
Sonuçta ise en uygun  $F_M(x)$  elde edilmektedir.

#### 1.4.6.6. EXtreme Gradient Boosting (XGBoost)

XGBoost da gradyan yükseltme temelli bir yöntemdir. Asıl ortaya çıkış amacı gradyan yükseltme yönteminin eğitim aşamasını kısaltmaktır. Chen ve Guesterin'nin 2016 yılında yayınlanan çalışmaları ile ün kazanmıştır (Chen ve Guestrin, 2016: 785). İsim olarak tam Türkçe karşılığı "Ekstrem Gradyan Yükseltme" olarak çevrilmektedir. Model isminin bu şekilde verilmesinin nedeni hesaplamaların mühendislik açısından en iyi sonucu vermeyi amaçlamasından kaynaklanmaktadır. Modelin diğer amacı ise aşırı uyum gösterme problemine karşı daha düzenli bir yöntem geliştirmektir (Chen, 2015: 1). Özetle model tam olarak randımanlı, esnek ve portatif olması amaçlanarak kurulmuştur (XGBoost, 2020: 1). Sınıflandırma problemleri için geliştirilmiş olmasına karşın regresyon ve sıralama problemlerinde de kullanılabilir.

Yöntem denetimli öğrenmeye uygun bir şekilde geliştirilmiştir. Yöntem çalışmaya ilk tahmini yaparak başlar, bulunan değer in önceden belirlenen baz bir değere (genellikle 0,5 varsayılan değerdir) karşılaştırılması ve buradan elde edilen değere göre yeni tahminin değerlendirilmesi ile ilerlemektedir. Ağacın gelişimi seviye bazlı ilerlemektedir. Şekil 1.18'te bu gelişim gösterilmiştir, her bir düğümden gelişim eşit olarak büyütülmektedir. Ağaçlar topluluk öğrenme yöntemlerinden paralel öğrenme ile üretilmektedir. Bu sebeple ayrı ayrı değerlendirilir. İlk ağaç üretildikten sonra hatalı tahminler gerçek gözlemlenen değere göre değerlendirilir ve aralarındaki fark tahmin hatasını vermektedir. İlerleyen aşamalarda ise örnek gözlem değerlerinden çıkarılarak tahmin hatası elde edilir. Her bir tahmin ağacı için benzerlik değeri (14) hesabı yapılır ve eldeki değişkenlerin dallarda ne kadar iyi gruplandığı bu değere

göre belirlenir. Lamda ( $\lambda$ ) değeri aşırı öğrenmenin önüne geçmek için düzenleyici bir değişken olarak kullanılmaktadır,  $\lambda$  değeri arttıkça benzerlik değeri hassaslaşacaktır. Zira benzerlik değeri ne kadar küçük ise yapraklar birbirinden o kadar ayrılmış olduğu anlaşılmaktadır. En iyi ağacı bulabilmek içinse tüm olası sonuçları içeren ağaçlar üretilmekte ve her biri için ayrıca benzerlik değeri hesabı yapılmaktadır.



Şekil 1.18 Seviye Bazlı Ağaç Gelişimi

$$\text{Benzerlik Değeri} = \frac{\text{kalanların toplamının karesi}}{\text{kalanların sayısı} + \lambda} \quad (14)$$

Benzerlik değeri en iyi olan ağaç, olası ağaçlar içinde en iyi ağaç anlamına geliyor gibi düşünülmesi mümkün olsa da bu şekilde bir seçimin takip edilmesi oldukça zordur. Bu sebeple tüm ağacın kazanç değeri (KD) hesaplanmaktadır (15).

$$\begin{aligned} KD = & \text{Sol yaprak}_{\text{benzerliği}} + \text{Sağ yaprak}_{\text{benzerliği}} \\ & - \text{Önceki ağacın}_{\text{benzerliği}} \end{aligned} \quad (15)$$

En yüksek değerli ağaç belirlendikten sonra budama işlemine başlanacaktır. Budama işlemi de kazanç değeri ile belirlenen Gama ( $\gamma$ ) değerine göre hesaplanmaktadır (16-17). Gama değeri yüksek seçilir ise aşırı öğrenmenin önüne geçilecektir. XGBoost yönteminde budama geriye doğru yapılmaktadır.

$$KD - \gamma > 0 \text{ ise ağaç olduğu hali ile tutulur} \quad (16)$$

$$KD - \gamma < 0 \text{ ise ağaç budanır} \quad (17)$$

Son aşamada ise tahmin yapılmaktadır. XGBoost yönteminde öğrenme oranı eta olarak adlandırılmaktadır ve tahmin değeri eta ile oluşturulmaktadır (18).

$$\text{Tahmin Değeri} = \text{İlk tahmin} + \text{eta} * 1. \text{ Ağaç} \quad (18)$$

Düzenleştirme ve budama işlemleri dışında Ağırlıklandırılmış Quantile Sketch de bu modele özgü bir gelişmedir. Bu bir karar ağacındaki bölümlenimin nasıl yapılacağını belirlemek için kullanılan bir yaklaşım algoritmasıdır. Bu algoritma büyük veriler için zor olan bir yöntem olduğundan yazarlar birleştirme ve budama işlemlerini destekleyen tamamlayıcı bir çalışma daha yayınlamıştır (Chen ve Guestrin, 2016: 785). Her değişken için yeni bir kazanç değeri hesaplanması XGBoost algoritmasının temelini oluşturur. Bu tipteki algoritmalara verilen genel ad açgözlü (greedy) algoritmalarıdır. Genel yapıdan farkı ise verileri parçalar ve

elde ettiği parçalara göre inceleme yapar. En küçük parçalara kadar inerek aralıkları incelediği için iyi sonuçlar verirken sonuç verme süresinde uzamalara sebep olabilmektedir. Bu duruma çözüm olarak eskiz (Stetch) aşaması sunulmuş ve parçaların gidebileceği sınırların belirlenmesi amaçlanmıştır. Ağırlık değeri hesaplanarak yüksek tahmin değerlerinin kararsız olduğu ortaya koyulur ve ağırlıklar eşitlenmeye çalışılacak şekilde bölümlenmeler yapılır. Asıl eklenen yenilik de modele eklenen ağırlıklı verileri işlemek için dağıtılmış bu algoritma yapısına sahip olmasıdır.

Sparsity-aware split finding (seyreklik uyumu/boş değer ile çalışabilme) ise algoritmanın diğer bir katkısı olmuştur. Yapısı itibariyle baz skora göre çalışan XGBoost bu tahmin değerine ilerken boş değerler için de hata değerleri belirleyecektir. Sonrasında oluşturulan her karar ağacı bir önceki üzerinden ilerleyeceği için elde edilen hata verilerine başvuracak, böylece her olasılık için boş değerleri de farklı dallara yerleştirme yapacaktır. Bu sayede eksik olmayan girdiler üzerinden doğrusal bir aramaya girilir. Daha iyi fayda değeri elde ettikçe veri eksikliği olan örnekler de ilgili dallarda kendilerine yer bulmaktadır. Özetle ana fonksiyondaki kaybı en aza indirecek şekilde ayrımlar yapılmakta olduğu için eksik veriler en iyileme aşamasında nerede daha iyi sonuç veriyor ise o yaprağa yerleştirilmektedir. Bunun haricinde özellikleri dallara ayırmak (split) için gradyan artma makinelerinden farklı olarak özel bir teknik kullanılmamaktadır.

Matematiksel gösterimi ise  $K$  ağaç sayısı olmak üzere,  $F$  tüm mümkün CART ağaçlarını göstermektedir.  $n$  tane gözlem  $m$  tane özelliğin olduğu bir veri setinde  $D = \{(x_i, y_i), x_i \in R^m, y_i \in R\}$  ( $|D| = n$ ),  $K$  kere deneme yaparak ağaç topluluğu oluşturmak böylece kendini geliştiren bir öğrenme amaçlanır (19).

$$\hat{y}_i = \sum_{k=1}^K f_k(x_i), \quad f_k \in F \quad (19)$$

$F = \{f(x) = w_{q(x)}, q : R^m \rightarrow T, w \in R^T\}$  ağaç üretim aşamasını göstermektedir,  $w$  tüm yaprak değerlerini tutan vektör,  $q$  karşılık gelen yapraktaki değer fonksiyonu ve  $T$  yaprak sayısıdır.  $l$  tahmini değer ile hedef değer arasındaki farkı ölçmek için kullanılan türevli dış bükey kayıp fonksiyonudur.  $g_i$  ve  $h_i$  ise kayıp fonksiyonuna ilişkin birinci ve ikinci derece gradyan istatistikleridir (20).

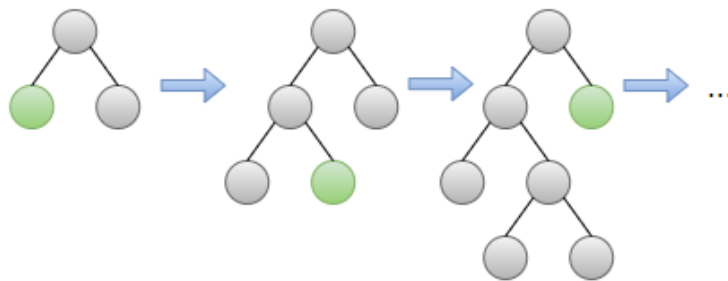
$$g_i = \partial_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)}), h_i = \partial_{\hat{y}_i^{(t-1)}}^2 l(y_i, \hat{y}_i^{(t-1)}) \quad (20)$$

Buna göre amaç fonksiyonu (21) şekillenmektedir.

$$obj^{(t)} = \sum_{i=1}^n \left[ l(y_i, \hat{y}_i^{t-1}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) \quad (21)$$

#### 1.4.6.7. LightGBM

LightGBM yöntemi de yüksek efektif gradyan yükseltme karar ağaçları olarak 2017 yılında yapılan çalışmaya dayanmaktadır (Ke vd., 2017: 3146). Makaleye göre diğer algoritmalara göre 20 kat hızlı sonuç vermesi, daha az kaynak gerektirmesi, tahmin oranının iyi olması gibi özellikleri ile ortaya çıkmıştır. Diğer geliştirilmiş gradyan artma makineleri yöntemlerinden farklı olarak yaprak odaklı (leaf-wise/level-wise) büyüme yöntemi tercih edilmiştir. Dikey olarak büyüyen algoritma yapısı diğer ağaç tabanlı öğrenme yöntemlerinden farklıdır. Yaprak odaklı ilerleyen karar ağacı örneği Şekil 1.19'da gösterilmiştir. Derinlik bazlı ilerleyen klasik modelde daha dengeli modeller elde edilerek ağacın büyümesi sağlanırken, yaprak odaklı büyümede kaybı en aza indiren bölünme işlemine gidilir. Daha az hata aynı zamanda daha çabuk öğrenme anlamına geldiği için yüksek hızını Gradyan Tabanlı Tek Yönlü Örnekleme (Gradient-Based One-Side Sampling –GOSS) borçludur (Li, 2018: 63). Fakat dikkat edilmesi gereken nokta veri sayısıdır. Az veri sayısında aşırı öğrenme gerçekleşebilmektedir (Nahon, 2019: 1). GOSS ile veri sayısı azaltılır böylece önemli veriler üzerinde çalışır ve bu da modele hız kazandırmaktadır. Azalma doğruluk oranını koruduğu veya yükselttiği sürece kullanılmaktadır. Bunu gerçekleştirmek için eldeki tüm veriler algoritma tarafından taranır ve sadece bilgi kazancı (information gain) olan verileri seçilerek hesaplamalar yapılmaktadır. Bu kazancı hesaplamak amacıyla maliyet fonksiyonu kullanılmaktadır. Tanjant eğimi ile belirlenen maliyeti yükselten veriler model için tercih edilmemektedir. Böylece modelin kendisi uç veya gürültü yaratabilecek örnekleri sadeleştirme yoluyla veri dağılımını da düzenlemiş olmaktadır (Ke vd., 2017: 3150).



Şekil 1.19 Yaprak Bazlı Gelişim

Bir diğer farklı yönü ise Özel Değişken Paketleme (Exclusive Feature Bundling – EFB) kullanılıyor olmasıdır. Değişken sayısını azaltma amacıyla kullanılan EFB aynı zamanda daha az değişken sayesinde model verimliliğini artırma amacıyla eklenmiş bir tekniktir. Seyrek

değişkenleri birleştirip daha güçlü değişkenler oluşturur. Veri setinde bulunan her bir satır değişkenlerine göre incelenmekte ardından aynı olan veriler birleştirilmektedir. Böylece daha belirgin özellikli değişkenlerle veri setindeki karmaşa azaltılır, daha düşük güç ve bellek tüketimi ile daha hızlı sonuç alınması sağlanmaktadır (Ke vd., 2017: 3151).

Eksik veri yönetimi XGBoost tekniği ile aynıdır.

Matematiksel olarak incelendiğinde model öncelikle GOSS yardımı ile varyans kazanımını hesaplanmaktadır. Gradyan artmanın her aşamasında negatif gradyan değerleri hesaplanır ve  $g_i$  olarak modelde yer almaktadır. Bu hesaplama mutlak değere göre yapılmaktadır. GOSS yönteminde bu hesaplama sonuçları azalan yöntem ile sıralanır ve en büyük gradyan değerlerine sahip ilk  $a * 100\%$ 'e göre veri örnekleri seçilir ve böylece  $A$  alt kümesi elde edilmektedir. Ardından, kalan set içinden  $(1 - a) * 100\%$  ile küçük gradyanlı örneklerden oluşan  $A^c$  oluşturulmaktadır. Sonrasında geri kalan verilerden  $b * |A^c|$  boyutunda rastgele örnekler seçilir ve  $B$  alt kümesi göre elde edilir. Sonunda ise örnekler  $A \cup B$  alt kümesi üzerindeki tahmini varyans  $\tilde{V}_j(d)$  kazancına göre bölünür (22).

$$\tilde{V}_j(d) = \frac{1}{n} \left( \frac{\left( \sum_{x_i \in A_l} g_i + \frac{1-a}{b} \sum_{x_i \in B_l} g_i \right)^2}{n_l^j(d)} + \frac{\left( \sum_{x_i \in A_r} g_i + \frac{1-a}{b} \sum_{x_i \in B_r} g_i \right)^2}{n_r^j(d)} \right) \quad (22)$$

Burada,  $A_l = \{x_i \in A : x_{ij} \leq d\}$ ,  $A_r = \{x_i \in A : x_{ij} > d\}$ ,  $B_l = \{x_i \in B : x_{ij} \leq d\}$ ,  $B_r = \{x_i \in B : x_{ij} > d\}$ ,  $d$  iterasyon sayısı,  $x_i$  girdi değişkenleri,  $B$  üzerindeki gradyanların toplamını  $A^c$  boyutunda normalleştirmek için ise  $\frac{1-a}{b}$  katsayısı kullanılmaktadır. GOSS örnek verileri küçük gradyanlarla artırırken orijinal veri dağılımını çok fazla değiştirmemeye odaklanmaktadır. Özel değişken algoritması bu aşamada seyrek bir özelliği daha az yoğun bir özellikle bir araya getirerek gradyan çözümünün hızlanmasını sağlamaktadır. Karar ağacından farklı olarak negatif zarar fonksiyonunu tek tek uydurmasıdır. LightGBM denklemi  $F_M(x)$ , ağırlıklı kombinasyon şemasıyla bulunur (23):

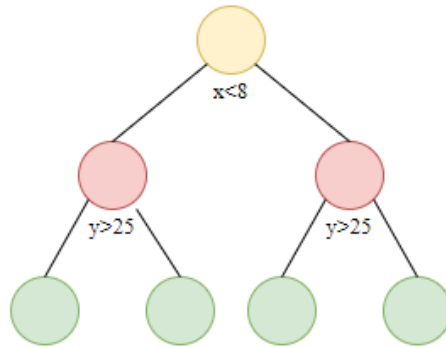
$$F_M = \sum_{m=1}^M \gamma_m h_m(x) \quad (23)$$

#### 1.4.6.8. CatBoost Algoritması

Categorical Boosting (CatBoost) makine öğrenmesi yöntemi diğer ana rakipleri olarak tanıtılan XGBoost ve LightGBM gibi gradyan artma karar ağaçları temellidir ve diğer

yöntemler gibi açık kaynak kodlu olarak yayınlanmıştır. Adını Category (kategori) Boosting (yükseltme) kelimelerinin birleşiminden almıştır (Vermeulen, 2020: 223). Yandex araştırmacıları ve mühendisleri tarafından arama, öneri sistemleri, kişisel asistan, hava tahmini ve diğer Yandex, CERN, Cloudflare gibi firmaların görevlerini üstlenmek üzere geliştirilmiştir. Özellikleri olarak: parametre ayarlamaları yapılmaksızın çok iyi sonuçlar vermesi, kategorik verileri ek işlemler gerekmeden kullanım desteği sunması, hızlı ve ayarlanabilir grafik işlemci kullanımı olması, yüksek doğruluk oranı ile sonuç vermesi ve son olarak da hızlı tahminlerde bulunması olarak gösterilmektedir (Yandex, 2017a: 1).

Karar ağacı bölümlenmelerini yeni bir teknikle yapmakta olan Catboost, bu tekniğe en az varyans örnekleme, “Minimal Variance Sampling” (MVS) adını vermiştir (Ibragimov ve Gusev, 2019: 1). Stokastik gradyan artma yönteminin ağırlıklandırarak hesaplanmasına dayanmaktadır. Teknik dâhilinde ağırlıklandırılmış örnekleme ağaç seviyesinde yapılmaktadır, dallara ayırma seviyesinde değildir. Oluşturulan her ağaç için ayrıca hesaplanarak en iyi doğruluk oranını veren bölümlendirilme kullanılır. Ağaç geliştirme aşamasında ise dengeli bir ağaç büyütme modeli kullanılır. XGBoost yöntemi ile benzer bir şekilde seviye bazlı bir ağaç gelişimi kullanılmaktadır. Diğer yöntemlerden farklı olarak Şekil 1.20’de gösterildiği gibi simetrik ağaç yapısı kullanılmaktadır.



**Şekil 1.20 CatBoost Ağaç Gelişim Örneği**

Eksik verilerle baş etme yöntemi XGBoost ve LightGBM’e göre farklıdır. Her bir eksik değer yerine minimum veya maksimum farklı bir değer verilir. Bu değerler eldeki verilerden az veya çoktur. Böylece ağaç bölümlenmesi yaparken gerçek verilerden ayırmak mümkün olmaktadır (Yandex, 2019: 1).

Farklı olarak her bir girdi değişkeni için ortalama bir tahmin değeri hesaplanmaktadır. Seçilen özelliğe olabilecek değişikliklerin nihai değeri ne kadar etkileyeceğine bakarak yapılan özel bir hesaplama sayesinde, fark yaratan girdi değerlerinin hangileri olduğu belirlenir (PredictionValueChange) (Yandex, 2017b: 1). Ayrıca girdi değişkenlerinin model üzerindeki kayıp değerinin hesaplandığı bir yöntem daha kullanılmaktadır. Bu yöntem ile girdinin seçilen

değişken değeri olmasa nasıl sonuç elde edilirdi sorusuna yanıt olarak yine önemlilik belirlenmektedir (Yandex, 2017b: 1).

Diğer ana rakiplerine göre farklarından biri kategorik verileri ön işlem gerektirmeden tanıyıp kullanabilmesidir (Micci-Barreca, 2001: 27). Bunu yaparken de aşırı öğrenmenin önüne geçecek bir mekanizmaya da başvurmuştur (Dorogush, vd., 2018: 2). Bütün girdiler alınır ve satır verileri birçok defa karıştırılır. Böylece karıştırma sayısı kadar permütasyon elde edilir. Veri seti  $\mathcal{D} = \{X_i, Y_i\}_{i=1, \dots, n}$  olmak üzere  $X_i = (x_{i,1}, \dots, x_{i,k})$  vektörü kimisi kategorik kimisi sayısal değerler olmak üzere,  $Y_i \in \mathbb{R}$  çıktı değerleri olmak üzere bütün kategorik veriler aşağıdaki formüle göre değerlendirilir (24).

$$Ort_{değer} = \frac{SınıftakiSayı + Öncü}{Toplam_{sayı} + 1} \quad (24)$$

Burada *SınıftakiSayı* bu kategorik özellik için toplam etiket değeri sayılarının maksimum etiket değeri tam sayısı  $k$ 'ya oranı, *Öncü* başlangıçta belirlenen sabit değer, *Toplam<sub>sayı</sub>* incelenmekte olan değişken değerine sahip değişkenlerin toplam sayısı olarak kullanılmaktadır.

CatBoost gradyan artma temelli bir yöntem olsa da, klasik gradyan artma yöntemlerinde önlenemez kayıp olduğunu öne sürmektedir. Bu durumun önüne geçmek için kullanmış oldukları kategorik veri değişimini önermektedirler fakat bu yöntemler kendi başına aşırı öğrenme yarattığı için eğitim verisinin bir kısmı iç öğrenme için ayrıca kullanılmaktadır (Prokhorenkova vd., 2019: 4).



## İKİNCİ BÖLÜM

### HAVAYOLU SEKTÖRÜ

Serbest piyasa ekonomisi koşullarında dünya ekonomisi açısından vazgeçilmez bir konuma ulaşmış olan havayolu taşımacılığı her geçen sene yük ve yolcu kapasitesi, kullanılan teknolojik altyapı ve havayolu şirketlerinin uçak filosu kapasiteleri konularında gelişme göstermektedir. Diğer taşıma modlarına göre yüksek hız ile ulaşım sağlaması yönüyle hem ulusal hem de uluslararası ulaşımında yüksek talep görmektedir. Bu sebeple bu bölüm kapsamında havayolu ulaşımının tarihçesi, günümüzdeki mevcut durumunun analizi, havayolu sektörünün güncel problemleri, son olarak ise gecikme sorunlarına karşı literatürde öncelikle makine öğrenmesi teknikleri ile yapılmış olan çalışmalar sonrasında ise kullanılan diğer yöntemlerin analizlerine yer verilmiştir.

#### 2.1. Havayolu Ulaşımının Tarihçesi

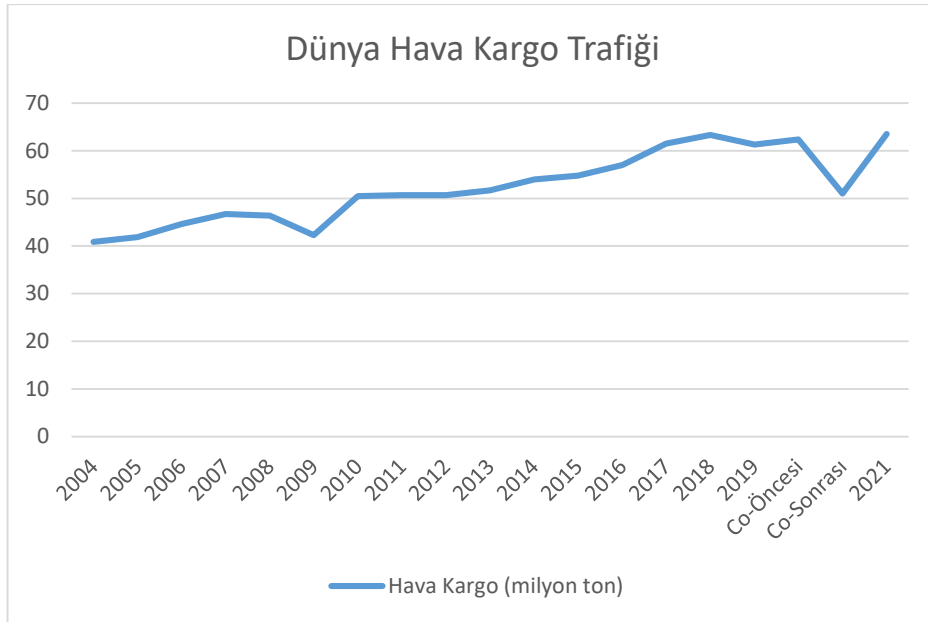
Ulaşım, ekonomik ve sosyal kalkınmanın kritik bir itici gücüdür. Bu durum ulaşım altyapısının insanları işlere, eğitime ve sağlık hizmetlerine bağlamasıyla; dünya çapında mal ve hizmetlerin tedarikini sağlamasıyla, insanların etkileşime girmesini sağlamasıyla ve uzun vadeli büyümeyi teşvik eden bilgi ve çözümleri üretmesi sayesinde gerçekleşmektedir (World Bank Group, 2016: 7). Havayolu ulaşımı için ise ilk adımlar Wright Kardeşlerin 1903 yılında yaptıkları uçak ile uçuşlarının ardından, ilk ödeme alınarak yapılan uçuş olan 1 Ocak 1914 tarihinde Tampa ile St.Petersburg arasında gerçekleşen uçuşla atılmıştır (Akbulut, 2019: 1). Bu uçuştan kısa süre sonra başlayan Birinci Dünya Savaşında hava gücünün kullanılması ve ortaya çıkan ihtiyaç ile birlikte daha iyi uçakların tasarlanması ticari havacılık için de teşvik sağlamıştır (Yalçın, 2016: 209). II. Dünya Savaşı'na kadar hava yolculuğu ciddi bir yükseliş göstermiştir. Amerika'da 1939-1943 yılları arasında 462.000 olan yolcu sayısı beş kat artarak 1.900.000'e yükselmiştir (Bowen ve Rodrigue, 2020: 1). Bununla birlikte ilgili dönemlerde uzun mesafeli veya kıtalar arası uçuşun çoğu sadece seçkinler veya hükümet yetkilileri tarafından kullanılmıştır. Bunun sebebi uçuşların çok yüksek ücretlere gerçekleşmesi ve çoğunlukla bu tip uçuşların kolonilere veya bağımlı bölgelere yönelik olmasıydı (McCormack, 1989: 374). II. Dünya Savaşı ise birçok yeni teknoloji üretimine olanak sağladı. Sonrasında uçakların kapasitesinin artması ile gelişmiş ülkelere uzun mesafeli yolculuklar için uçaklar tercih edilir hale gelmiştir (Brooks, 1967: 179). 1956'da Amerika Birleşik Devletleri'ndeki Pullman adı verilen araba (yataklı) ve yolcu otobüsü sınıfı trenlere kıyasla hava yoluyla

şehirlerarası seyahat edenlerin sayısı daha fazlaydı (Bowen ve Rodrigue, 2020: 1). 1960'larda ise jetlerin ortaya çıkışı devrim niteliğinde olarak değerlendirilmektedir (Brooks, 1967: 179). Bu keşif ile hava taşımacılığının hızı iki katına çıkmış ve havayolu üretkenliğini radikal bir şekilde artırmış, böylece ücretlerin düşmesini sağlamıştır (Bowen ve Rodrigue, 2020: 1). O günlerden bu yana en önemli gelişmelerden biri havacılıktaki kaza risklerinin azaltılması olmuştur. Sivil havacılık, 1960'ların başındaki ile aynı, milyon kalkış başına kaza oranına sahip olsaydı, 2018'de dünyanın herhangi bir yerinde günde yaklaşık üç ölümcül kaza olurdu (Bowen ve Rodrigue, 2020: 1).

Gelişim süreci incelenen havayolu ulaşımı teknolojik gelişmelerle günümüzde oldukça yaygın bir şekilde kullanılmaktadır. Devam eden bölümde ise havayolu ulaşımın günümüzdeki durumu tartışılmaktadır.

### 2.1.1. Günümüzde Havayolu Ulaşımı

Dünya Bankasına göre ticaret, küresel yoksulluğu sona erdirmenin merkezidir. Uluslararası ticarete açık olan ülkeler daha hızlı büyüme, yenilik yapma, üretkenliği artırma ve insanlarına daha yüksek gelir ve daha fazla fırsat sağlama eğilimindedir (World Bank Group, 2018: 1). Havayolu ulaşımı da küresel ticarete açılan önemli taşıma modlarından biri olma özelliğini korumaktadır. Hava taşımacılığının dünya mal ticaretindeki payına göre ölçüldüğünde % 1'den az fakat değer olarak % 35'ten fazladır (Shepherd vd., 2016: i). Tipik olarak, hava taşımacılığı, uzun mesafelerde taşınan zamana duyarlı, değerli veya dayanıksız yükler için tercih edilmektedir. Hızlı malzeme teslimatının maliyet sorunlarının önüne geçtiği acil durumlarda hava kargo hayati önem taşır. Hızlı teslimatı ile bilenen Amerikalı firma Amazon genellikle iyi depo yönetimiyle adını duyurmuş olsa da aynı zamanda teslimatları için 2016 yılında ilk uçağını almış 2021 yılına kadar da uçak filosunu 70'e kadar çıkarmayı planlamaktadır (Lojiport, 2019: 1). Yıllar bazında hava kargo miktarı incelendiğinde 2004 yılından itibaren genellikle artma eğiliminde olduğu görülmektedir. Şekil 2.1'e göre sadece 2009, 2019 ve 2020 yıllarında gerileme görülmüştür (Mazareanu, 2020a: 1). 2007 yılında ABD'de riskli konut kredilerinin geri ödemelerinde başlayan sorunun 2009 yılında dünya çapında bir küresel krize dönüşmesi nedeniyle 2009 yılında gerileme yaşanmıştır (Elele, 2009: 9). 2019 yılı ise küresel krizden bu yana gerçekleşen en düşük performanslı yıl olmuş ve taşınan yük miktarında %3,3 oranında düşüş gerçekleşmiştir (Ankahaber, 2020: 1). 2021 için ise araştırmacının öngörüsü grafikte yer almaktadır.



Şekil 2.1 Dünya hava kargo trafiği (Mazareanu, 2020a)

Yıllara göre havayolu tercih eden yolcu sayısı da Şekil 2.2’te görüldüğü gibi sadece küresel krize denk gelen 2007-2009 yılları arasında daha durağan hale gelmiş onun dışında hep yükseliş eğiliminde olmuştur. IATA’ya göre 2020 yılı için beklenen durum ise bu gidişatın aksine, Covid-19 nedeniyle yolcu trafiğinin %66 oranında düşmesidir (Hürriyet, 2020: 1). 2021 için ise araştırmacının öngörüsü grafikte yer almaktadır.



Şekil 2.2 Yıllara göre havayolu kullanan yolcu sayısı (Mazareanu, 2020b)

### 2.1.2. Havacılıkta Güncel Sorunlar

Hava yollarında yaşanan sorunlara bakıldığında çeşitli problemler göze çarpmaktadır. Bunlar uçuş gecikmeleri, havayoluna sektörünün küçük topluluklar tarafından ulaşılabilir olmaması, güvenliğin artırılma ihtiyacı ve havacılık sektörünün çevresel uyumu olarak

belirlenmiştir (Transportation Research Board, 2002: 51). 2014 yılında sektörün önde gelenleri tarafından yapılan ankette paydaşların öne çıkardığı problemler ise alt yapı problemleri, sektörün karlılığı, emniyet ve güvenlik, vergilendirme ve petrol fiyatlarındaki istikrarsızlık olarak ortaya koyulmuştur (Maslen, 2014: 1). Daha güncel bir habere göre ise havayolları ile ulaşım eskisine göre çok daha güvenli, verimli ve karlı olmakla birlikte endüstrinin küresel doğası gereği yetersiz altyapı sorunları, salgınlar ve siyasete kadar birçok unsura karşı hala kırılgan bir yapıya sahip olduğudur (Zhang , 2018: 1). Bir başka haber ise politik sorunlar, eğitilmiş iş gücü, havayolları kapasitesi, sürdürülebilirlik, güvenlik, artan pazar rekabeti olarak durum değerlendirmesi yapmaktadır (Beard, 2019: 1).

İncelenen tüm görüşler, özetlemek gerekirse havayolu sektöründe güvenlik, sektörün kendine has kırılgan yapısı ve çevresel etkileri öne çıkan problemler olarak görülmüştür. Buna istinaden devam eden alt başlıklarda bu konular incelenmiştir

#### **2.1.2.1. Uçuş Güvenliği**

Diğer taşıma türlerine göre daha güvenli bir taşıma modu olarak görülmekte olsa da uçak kazalarının ölümcül sonuçları olması nedeniyle uçuş güvenliği son derece önemlidir. Aviation Safety Network'e (2017) göre 2017 yılı havacılık sektöründe en güvenli yıl olarak ilan edilmiştir. 2017 senesinde 44 ölümlü sonuçlanan toplam 10 uçak kazası gerçekleşmiştir. Bu her ne kadar artan önemlerin faydalı olduğu sonucunu ortaya koyan bir durum olsa da takip eden senede ise uçak kazası sonucu olan ölüm oranı %15 yükseliş göstermiştir (Beard, 2019: 1). Hollandalı danışmanlık şirketi To70'in raporuna göre 16 milyon uçuş için bir ölümcül kaza olduğu tahmin edilmektedir (BBC, 2018: 1).

#### **2.1.2.2. Kırılgan Yapı**

Havacılık sektörü yapısı itibariyle birçok beklenmedik olaydan etkilenmektedir. Sektör bugüne kadar yaşanan birçok krizden -11 Eylül, benzin fiyatlarındaki dalgalanmalar ve ekonomik durgunluk gibi- etkilenmiştir. Son olarak korona virüsün ortaya çıkması ile bütün sektör beklenmedik bir şekilde sarsılmıştır (Reed, 2020: 1). Kırılgan yapı çevresinde politik etkiler, petrol fiyatlarındaki dalgalanmalar, salgın hastalıklar, uçuş gecikmeleri ve çevresel etkiler devam eden alt başlıklarda ele alınmış ve detaylandırılmıştır.

##### **2.1.2.2.1. Politik Etkiler**

Havacılık sektörü uluslararası yapısı nedeni ile politik gelişmelerden yoğun bir şekilde etkilenmektedir. Uluslararası güncel problemler incelendiğinde Amerika ve Çin arasındaki ticaret savaşları dikkat çekmektedir. Amerika 360 milyar doları aşan miktardaki Çin mallarına

gümrük vergisi koymuştur. Buna karşın Çin yönetimi de değeri 110 milyar dolardan fazla olan Amerikan mallarına gümrük vergisi uygulaması başlatmıştır. Bu durum ekonomik dinamikleri çok ciddi etkilemiştir (BBC, 2019: 1). Çin aynı zamanda birçok uçak parçasının da üreticisi konumundadır. Ticaret savaşlarından yansiyacak herhangi bir olumsuz etki, parçaların Çin'den dünyadaki diğer üreticilere ihracatını etkileyebileceği ve kıtalar boyunca havacılık tedarikçileri üzerinde zincirleme yıkıcı etkiye dönüşebileceği ihtimali değerlendirilmektedir (Beard, 2019: 1). Brexit ile İngiltere'nin Avrupa Birliği'nden (AB) çıkması, bunun sonucunda AB ile ticaret anlaşmalarının önemli derecede etkilenmesi ile farklı belirsizlikler ortaya çıkmıştır. 31 Aralık 2020 tarihine kadar ticaret anlaşmaları sürecektir olan İngiltere'nin yeni yapacağı anlaşmalarla ticaretinin şekillenmesi beklenmektedir (Department for International Trade, 2020: 1). Bu tip belirsizlikler sektör üzerinde olumsuz etkiler yaratmakta ve tahmin edilebilir olmaktan uzaklaştırmaktadır.

#### **2.1.2.2.2. Petrol Fiyatlarındaki Dalgalanmalar**

Petrol fiyatları havacılık sektörü için çok önemlidir, çünkü hava taşımacılığının ana gider kalemini yakıt maliyeti oluşturmaktadır (Kaya ve Kendirli, 2018: 115; Zhang, 2018: 1). Petrol fiyatları birçok değişkenden etkilenen hassas bir piyasaya sahiptir. IATA'nın 2019 yılında hazırlanmış olduğu raporda ABD ve İran arasında gerilim sebebi ile varil petrol fiyatları 70\$'dan 84\$'a kadar yükseldiği belirtilmiştir. Raporun yazıldığı tarihte ise ham petrol 65\$'a gerilemiş jet yakıt fiyatı 77\$ olarak belirlenmiştir. Rapor oynaklığın 2020 yılında da devam edeceğini öngörmüştür (IATA, 2019: 1). Ek olarak IATA'nın finansal müdürü petrol fiyatlarının "yoğunlaşmış jeopolitik risklere karşı savunmasız" olmaya devam edeceğini öngörmektedir (IATA, 2020a: 1).

#### **2.1.2.2.3. Hastalık**

Havayolu sektörünün kırılgan yapısı ortaya koyulurken salgından politikaya kadar büyük bir yelpazede olduğu Zhang (2018) tarafından belirtilmiştir. Fakat etkilerinin ne derece yıkıcı olabileceği 2020 yılında Korona virüs salgını ile net bir şekilde görülmüştür. Bulaşma tehlikesi sebebi ile birçok ülke uçuşları süresiz durdurulmuştur. Örneğin, 12 Mart'ta Amerika 26 Avrupa ülkesinden tüm uçuşları durdurduğunu açıklamıştır (BBC, 2020b: 1). Havacılık araştırma şirketi Cirium'a göre, 3 Nisan 2019 Çarşamba günüyle 2020 karşılaştırıldığında, Çinli operatörler %40 daha az Airbus ve Boeing tipi uçak ile uçmuş, toplamda %65 daha az uçuş saati ve %63 daha az uçuş döngüsü gerçekleştirmiştir (Ciricum, 2020: 1). IATA'nın 2020 yılı için öngörüsü %-20,1'lik net kar marjı ile 84,3 milyar dolar kaybetmesi yönünde olmuştur

(IATA, 2020b: 1). Bu tip küresel gelişmelerin birincil etkilendiği sektörlerden biri havacılık sektörü oldu (BBC, 2020a: 1).

#### **2.1.2.2.4. Uçuşlarda Yaşanan Gecikmeler**

Havayolları ve yolcular için tekrarlayan ve öngörülemez gecikmeler en büyük sorunları oluşturmaktadır (Transportation Research Board, 2002: 51). Sınırlı kapasiteye sahip olmaları nedeniyle ticari havacılık endüstrisinin ihtiyaçlarını sağlayabilmek amacıyla havaalanı işlemleri akıcı ve kesintisiz olmalıdır. Bu sebeple havalimanlarındaki kaynakların verimliliğinin ve etkinliğinin iyileştirilmesi artan talebi karşılamak amacıyla kritik öneme sahiptir. Sağlanamadığı durumda uçuş sektöründe artan talep ayrıca uçuş gecikmelerine de sebep olmaktadır (Meel vd., 2020: 71). Gecikmeler ile ilgili verilebilecek en önemli maliyet kaynaklı örnek, tüm yolcuların toplam seyahat sürelerinin yaklaşık %20'sinin gecikme kaynaklı olmasıdır. Bu durum ise tahmin edilmesi mümkün maliyetlere sebep olabilmektedir. Çoğunlukla gecikme maliyetleri sadece yolculara ilişkin olarak değerlendirilse de bu açıklama yetersizdir (Cook ve Tanner, 2009: 9). Gecikmelerin yıllık 35 milyar dolarlık bir zarara sebep olduğu ve gerçekleşen bu zararın sadece havacılık sektöründe ait olmadığı aynı zamanda perakende hizmeti verenlere, konaklama hizmeti sağlayanlara ve turizm hizmeti sağlayıcılarına da yansması maliyetlerin son derece köklü olduğunu göstermektedir (Yang vd, 2020: 1). Gecikmelerin önceden doğru ve kesin tahmini gecikme azaltma önemlerinin çevik bir şekilde uygulanmasına imkan sağlayacaktır. Sağlanabilecek faydalar operasyon, ekonomik, çevresel ve sosyal olarak özetlenebilmektedir (Yang vd., 2020: 5). Gecikmesi muhtemel uçuşların önceden bilenebilmesi halinde havaalanlarının sınırlı kaynaklarını diğer gecikme öngörülme-yen uçuşlar için değerlendirilmesi mümkün olabilir. Böylece operasyonel işlem yüklerinin hafifleyeceği tahmin edilmektedir. Ekonomik faydaları ise önceden bilgilendirme sayesinde oluşması muhtemel müşteri memnuniyetsizliklerinin önlenmesi, gecikme yapılacak uçakların apron işlemlerinin başlamaması sayesinde zaman, yer ve yakıt tasarrufu sağlanması ve operasyonel faydaların ekonomik olarak dönüşleri gibi faktörler değerlendirilmektedir. Uçak gecikmelerinin yolcuların seyahat kararlarını ve havayollarının ücretlendirmelerini etkilediği bilinmektedir. Bu durum şöyle gelişmektedir; bir rotada bir havayolu firmasının sıklıkla gecikme yaşaması üzerine yolcu talebinin düşeceği ve ortalama ücretlerin de dolaylı olarak yükseleceği kanıtlanmıştır. (Britto vd., 2012: 466). Çevresel fayda olarak inceleme yapıldığında ise yıllık uçak gecikmelerinden dolayı uçakların pistte geçirdiği süre boyunca 2,8 milyar litre yakıt kullanmaları (Crawley, 2008: 1) sonucu ortaya çıkan emisyonların ve kullanılan yakıt miktarın azaltılacağı tahmin edilmektedir. Sosyal fayda ise yolcuların daha rahat, kısa ve ucuza

seyahat etmesinin mümkün olması yanı sıra havaalanı çevresinde yaşayan yerleşik haneler için ses kirliliğinin azaltılması anlamına gelmektedir (Yang vd., 2020: 4-5).

### 2.1.2.3. Çevresel Etkiler

Havayolu ulaşımının tarihsel olarak ilk verdiği rahatsızlık gürültü yönünde olmuştur (Transportation Research Board, 2002: 71). Sebebi ise 1960 yıllarında artan kentsel havaalanları ve daha büyük jetlerin yapımı olarak düşünülmektedir. Uçak gürültüsüne maruz kalma ile psikoterapik ilaç kullanımı arasında bir ilişki olduğuna dair bazı kanıtlar mevcuttur (Morrell vd., 2008: 221). Bir başka çalışmaya göre ise çocuklarda okuduğunu anlama bozukluğu ve yüksek düzeyde gürültü rahatsızlığı arasında ilişki bir olduğu ortaya koyulmuştur (Haines vd., 2001: 265). Yüksek uçak gürültüsünün insanlar üzerindeki olumsuz etkisinin yanı sıra hayvanlar üzerinde de negatif etkileri olduğu ortaya konulmuştur (Pepper vd., 2003: 429). Ses dışında diğer çevresel etkiler de mevcuttur. Bunlara ozon, karbon monoksit, nitrojen dioksit, sülfür dioksit, asılı partikül madde ve kurşun dahil olmak üzere uçak kullanımı ile ortaya çıkan hava kirleticileri dahildir (Transportation Research Board, 2002: 72). Kalkışın ilk üç dakikasında okyanus ötesi bir yolcu uçağı, yaklaşık 50 bin metre küp hacme ve 800 °C sıcaklığa sahip, iki Mg karbondioksit içeren bir sıcak egzoz gazı akışı yayar ve bununla birlikte bir uçak motoru kullandığı petrolden birçok farklı gaz salınımı yapar. Bunlar nitrojen (% 75,2), oksijen (% 16,3) ve egzoz gazlarından (% 8,5) oluşmaktadır (Jakubiak, 2015: 279). Bu gazlarla birlikte, uçakların kullanımıyla da ortaya çıkan ve atmosferde uzun süre kalma özelliğine sahip bir gaz olan karbondioksit de ortaya çıkmaktadır. Bu durum da dünyadaki iklim değişikliğine katkıda bulunmaktadır (Daley, 2016: 47).

Havayolu ile ulaşımın günümüzde oldukça yaygınlaştığı, yapısı itibariyle küresel olaylardan kolaylıkla etkilendiği görülmüştür. Sektörün kırılğan yapısı ve karşılaşmakta olduğu yüksek maliyetler (yakıt, ilk alım maliyeti, gecikme maliyetleri) daha verimli bir çalışma sistemi ve gelecek ile ilgili iyi bir tahmin yeteneği gerektirmektedir. Bu ihtiyaç çalışmanın amacındaki uçakların gecikme tahminini ortaya çıkarmıştır. Devam eden başlıkta bu konudaki literatüre yer verilmiştir.

## 2.2. Literatür Taraması

Literatür incelemesine yer verilen bu başlık altında öncelikle makine öğrenmesi tekniklerini kullanarak konuyu inceleyen çalışmalar sonrasında ise diğer kullanıldığı çalışmalar incelenmiştir. Son olarak literatürün çalışmaya etkisine yer verilmiştir.

### 2.2.1. Makine Öğrenmesi ve Uçuş Gecikmeleri

Makine öğrenmesi teknikleri ile uçuş gecikmeleri konusunda yapılan ilk çalışmanın 2014 yılında olduğu tespit edilmiştir. Bahsi geçen çalışmada rastgele ormanlar ve regresyon algoritmaları kullanılmış, 60 dakikalık geç kalmalar incelenmiş ve %81'lik bir doğruluk oranı ile sonuç alınmıştır. Çalışma Amerika'daki havaalanları ile 2007 ve 2008 yılları için yapılmıştır, girdi değerleri olarak sezon bilgisi, kalkış zamanı, haftanın günü, havayolu firması, uçak numarası, kalkış zamanı, planlanan kalkış zamanı, planlanan ve gerçekleşen çıkış kapısı işlemleri, kalkış ve varış hava limanları, hava durumu verileri kullanılmıştır (Rebollo ve Balakrishnan, 2014: 231). Bir diğer çalışmada hava durumu sebepli uçuş gecikmeleri tahmin edilmiştir. Öncelikle veri setini dengelemek için SMOTE kullanılmıştır. Ardından karar ağaçları ve rastgele ormanlar yöntemleri kullanılarak uçuşların da gecikme yaşanıp yaşanmayacağı edilmiştir. Girdi olarak ise Amerikan yerel uçuş verisi ile hava durumu verileri 2005 ve 2015 yılları arası alınarak kullanılmıştır. Seçilen değişkenler tarih bilgileri, planlanan kalkış ve varış zamanları, varış gecikme indikatörü 15 dakikadan düşük ise 0 fazla ise 1, rüzgârın yönü, rüzgar hızı, görüş mesafesi, yağış miktarı, kar kalınlığı ve yüksekliği, hava durumu kodu (iyi, orta, kötü, çok kötü), genel hava durumu (8 grup), yağış kodu (9 grup), görüş mesafesi kodu (8 ayrı grup), diğer hava durumu kodları (5 grup), kombinasyon kodları (farklı hava durumlarının kombinasyonu olup olmadığı) olarak belirlenmiştir. En iyi sonuç rastgele ormanlar yöntemi ile elde edilmiş ve SMOTE kullanılan durumda doğruluk %83,4 kullanılmayan durumda ise %81,37 olarak bulunmuştur (Choi vd., 2016: 1).

Aynı yıl yapılan bir başka çalışmada ise iki aşamalı bir model kurulmuştur. Modelin ilk aşamasında uçuşların geç kalıp kalmayacağı bilgisi sorgulanmış sonrasında ise eşik değerleri aralığında ne kadar geç kalacağına bakılmıştır. Girdi olarak 10 Amerikan havaalanı seçilmiştir. Metot olarak tekrarlayan sinir ağları (RNN) ve uzun kısa süreli bellek (LSTM) kullanılmıştır. Girdi değişkenleri haftanın günü, mevsim, ay, tarih, rüzgar yönü, rüzgarın hızı, bulut yüksekliği, görüş uzaklığı, yağış, kar birikimi, yoğunluğu gibi hava durumu bilgileri günlük ortalamalar olarak belirlenmiştir. İkinci aşamadaysa ilk aşamanın çıktıları (uçanın geç kalıp kalmayacağı) haftanın günü, mevsim, ay, tarih, kalkış havaalanı, varış havaalanı, planlanan kalkış saati, planlanan varış saati, kalkış havaalanında geç kalma durumu, varış havaalanına geç kalma durumu, rüzgarın yönü, rüzgarın hızı, bulut yüksekliği, görüş mesafesi, yağış, kar birikimi, yoğunluğu kullanılmıştır. Çalışmada geçen havaalanları için en az %85 olmak üzere %92'ye kadar tahminlerin doğruluk oranları olduğu tespit edilmiştir (Kim vd., 2016: 1). Aynı yıl yapılan son çalışmada ise uçağın gelişinin geç kalıp kalmayacağı tahmini yapılmıştır. Çalışmada 15, 30 ve 60 dakika gibi belirli gecikme sürelerine göre inceleme yapılmıştır. Girdi



olarak kalkış ve varış havalimanı, planlanan kalkış ve varış zamanı, hem kalkış hem varış havalimanlarının hava durumu seçilmiştir. Karar ağaçları yöntemlerinden olan C4.5, destek vektör makineleri (SVM), stokastik gradyan artma, naive bayes ve lojistik regresyon yöntemleri kullanılmıştır. En iyi sonucu rastgele ormanlar yöntemi ile elde etmişlerdir. Veri olarak Bureau ulaştırma istatistikleri 2009 ve 2013 yılları aralığında kullanılmıştır. 15 dakikalık tahminlerin doğruluk oranı %74,2 iken 60 dakikalık geç kalma periyodu için %85,8'e kadar çıkmıştır (Belcastro vd., 2016: 19).

Literatürdeki boşluğu kapatmak için Manna ve diğerlerinin (2017) çalışmasında önceki çalışmalardan farklı olarak gradyan artırılmış karar ağacı yöntemini kullanmışlar ve uçuşların geç kalma süresini bulmayı amaçlamışlardır. Veri seti olarak Amerika'nın en yoğun 70 havaalanı için 2013 yılının Nisan ve Ekim ayları arası seçilmiştir. Girdi değişkenleri ise tarih bilgileri, havayolu firması, kalkış havalimanı, varış havalimanı, planlanan kalkış, kalkış geç kalma süresi, planlanan varış ve varış geç kalma süresi olarak belirlenmiştir. Hem kalkıştaki geç kalmalar hem de inişteki gecikmeleri dahil eden çalışmada belirleme katsayısı kullanılmış (coefficient of determination) ve modelin açıklama oranını iniş için %92,3, kalkış için %94,9 olarak bulunmuştur. (Manna vd., 2017: 5). 2017 yılında yapılmış olan bir diğer çalışmada ise hava kontrol için basit bir karar destek modeli sağlamak amaçlanmıştır. Karar ağaçları, rastgele ormanlar, adaboost ve k-en yakın komşular yöntemleri kullanılarak modeller karşılaştırılmıştır. Çalışmadaki farklı bir yaklaşım ise modelleri zarar modeli üzerinden çalıştırmak olmuştur. Bu model hata matrisine dayanmakla beraber yanlış pozitif oranı ile yanlış negatif oranının ağırlıklandırılması ile elde edilmiştir. 2005-2015 yılları arasındaki Amerika'ya ait yerel uçuş trafik bilgisi ile hava durumu verileri çalışmada kullanılmıştır. Veriler dahilinde girdi olarak kalkış ve varış yeri, tarih bilgisi, planlanan kalkış varış, varış geç kalma indikatörü, rüzgarın yönü, rüzgarın hızı, görüş mesafesi, kar derinliği, yağış miktarı, hava durumu kodu (iyi, ortalama, kötü, çok kötü) hava durumu açıklayan kod (güçlü toz, kum fırtınası, ani şiddetli fırtına, hortum, vb.) kullanılmıştır. Çalışmanın sonucunda en iyi sonucu adaboost yöntemi %83,07 doğruluk oranı ile vermiştir. Diğer yöntemlerin doğruluk oraları ise rastgele ormanlar için %82,75, karar ağaçları için %82,49 ve k-en yakın komşular için %80,6 olarak elde edilmiştir (Choi vd., 2017: 7).

Bir başka çalışmada ise Tokyo'daki uçakların geç kalma tahmini yapılmıştır. Çalışmada tüm Japon hava sahasındaki uçaklar, kalkış öncesi bilgiler; hava durumu geç kalma sapmaları gibi, kalkışa dair zaman bilgileri enlem, boylam, rakım, uçak tipi verileriyle 10 saniyede bir güncellenen 2012 yılına ait 42 günlük veri kullanılmıştır. Deterministik kuyruk modeli uygulanarak kalkış ve inişler arasındaki zaman hesaplanmış ardından yapay sinir ağları

kullanılmış ve iki model birbiri ile karşılaştırılmıştır. Yapay sinir ağlarına girdi olarak verilen verilerde ise; uçak tipi, kalkış saati, tahmini geliş saati, sektöre tahmini giriş saati, sektöre giriş yeri, tahmini rüzgar, sektördeki uçak sayısı, sektöre giriş aralıkları olarak belirlenmiştir. Modelin amacı hava kontrol kaynaklı geç kalışları tahmin etmektir. Performans metriği olarak kare ortalamalarının karekökü (RMS) seçilmiştir. Bu metrik genellikle sayısal değerler içeren tahminlerde kullanılmakta olduğu için her iki yöntemi karşılaştırabilmek amacıyla seçilmiş 183,7 ile kuyruk yöntemine göre (230,4) daha iyi sonuç vermiştir. Ek bir düzenleme yaparak zaman aralıkları belirleyen bir gösterge ile RMS değeri 174'e kadar indirilmiştir (Takeichi vd., 2017: 6).

İki aşamalı bir başka çalışmada ise öncelikle gecikme olup olmayacağı konusunda uçuşlar sınıflandırılmış sonrasında regresyon modeli ile gecikme süresi tahmin edilmiştir. Çalışmanın son aşamasında ise gerçek zamanlı bir karar destek sistemi geliştirilmiştir. Böylece yolcuları ve havayollarını geç kalması muhtemel uçaklar için önceden uyararak amaçlanmıştır. Modele girdi olarak kalkış anındaki sıcaklık, rüzgâr hızı, rüzgâr yönü, nem, görüş mesafesi, havayolu adı, uçuş numarası, kalkış havaalanı, varış havaalanı, yıl, yılın hangi çeyreği olduğu, tarih, haftanın günü, planlanan kalkış zamanı, planlanan varış zamanı, rüzgâr yönü, nem, basınç ve sıcaklık verilmiştir. Sınıflandırılma yapılan ilk aşama için örneklem üzerinde rastgele ormanlar, gradyan artma, AdaBoost ve ekstra ağaçlar yöntemleri kullanılmış; SMOTE ve Tomek Link bir arada kullanıldığında en yüksek doğruluk değeri aldığı tespit edilmiştir. Uygulamada ise rastgele ormanlar, gradyan artma, AdaBoost, ekstra ağaçlar kullanılmıştır ve en yüksek doğruluk değerini gradyan artma yöntemi vermiştir. İkinci aşamada ise regresyon ile geç kalma süreleri tahmin edilmiş bu durum içinde en düşük  $R^2$  değeri (0,012) ve hata kareler ortalamasına göre de en iyi değer (1261,75) YSA ile elde edilmiştir (Thiagarajan vd., 2017: 1).

Kuhn ve Jamadagni (2017) yılında yapılan çalışmada ise uçuşların geç kalıp kalmayacağını tahmini için sınıflandırma problemi önerilmiş, üç farklı yöntem ile denemeler yapılmıştır. Karar ağaçları, lojistik regresyon ve yapay sinir ağları kullanılan çalışmada yöntemler birbirleri ile karşılaştırılmıştır. Girdi olarak kalkış ve varış havalimanı, planlanan kalkış, gerçekleşen kalkış, kalkışın geç kalma süresi, taksi yolu çıkışı ve havalanma saati, planlanan uçuş süresi, geç kalınan süre, mesafe, piste varış, planlanan varış, varış zamanı, geç kalma süresi, uçuş geç kalma grupları (uçuş sistemleri, güvenlik önlemi gibi) verileri kullanılmıştır. Veri seti Kaggle sayfası üzerinden çekilmiş ve 2015 yılı için olan beş milyon satır üzerinde veri kullanılmıştır (Kuhn ve Jamadagni, 2017: 1). Venkatesh ve diğeri (2017) ise aynı şekilde uçuşların geç kalıp kalmayacağını yapay sinir ağları ve derin inanç ağları kullanarak tahmin etmişlerdir. Kullanılan veri seti Kaggle üzerinden temin edilmiştir, fakat

hangi verinin hangi döneme ait olduğuna yer verilmemiştir. Girdi olarak 30 farklı değişken olduğu bunların en önemlilerinin haftanın günü, varış zamanı, kalkış zamanı, havada geçen süre, toplam geç kalma süresi ve mesafe olduğu belirtilmiştir. Doğruluk oranı derin sinir ağlarında %77 iken, YSA'da %92 olarak elde edilmiştir (Venkatesh vd., 2017: 562).

Mogha ve diğerleri (2017) ise yaptıkları çalışmada yine uçuşların geç kalıp kalmayacağı sorgusunu yapmış fakat farklı yöntemler kullanmışlardır. İlgili yöntemler; karar ağaçları, Naive Bayes, rastgele ormanlar ve destek vektör makineleri olmuştur. Büyük veri üzerinde makine öğrenme yöntemlerini karşılaştırmak isteyen araştırmacılar veri setini Github sayfasında hazır bulunan veri üzerinden temin etmiştir. Girdi olarak tarih bilgileri, taşıyıcı firma, uçuş numarası, uçak numarası, kalkış ve varış noktaları, gerçekleşen ve planlanan kalkış saati ve varış saati, geç kalma süresi, mesafe kullanmışlardır. En iyi sonuç karar ağaçları yöntemi ile 0,70 doğruluk oranı olarak elde edilmiştir (Mogha vd, 2017: 17). Chandramouleeswaran ve diğerlerinin (2018) çalışmasında ise 15 dakikalık periyodlara göre uçuş gecikmeleri incelenmiştir. Çalışmada yapay sinir ağları ile lojistik regresyon modelleri karşılaştırılmıştır. Veri seti olarak 2005 yılından 2017 yılına kadar olan Amerika Birleşik Devletleri uçuş taşımacılığı verileri kullanılmıştır. Girdiler uçuşların zamanlaması, hava durumu, tüm uçuşların trafik durumu, bağlantı geç kalmaları (geçmiş ve şimdiki bağlantılı geç kalmalar) ve performans verileri (geçmiş ve şimdiki ortalama kalkış gecikmeleri ve hedeflenen geç kalma durumu) olarak belirlenmiştir. Yöntemler arasında en iyi sonucu YSA ile elde etmişlerdir. Tahmin periyodları uzadıkça doğruluk oranının da arttığı gözlemlenmiş olsa da 30 dakikalık gecikmenin doğruluk oranı %98 olarak elde edilmiştir (Chandramouleeswaran vd., 2018: 1).

Aynı yıl yapılan bir başka çalışmada ise karar ağacı temelli yöntemlerden; RepTree, Decision Stump, rastgele ormanlar ve karar ağacı yöntemleri kullanılmıştır. Veri seti Mısır Havayollarının 512 uçuşundan, girdileri ise tarih, uçuş numarası, kalkış ve varış noktası, planlanan kalkış, planlanan varış, gerçekleşen kalkış, gerçekleşen varış, uçak tipinden oluşmaktadır. Doğruluk F1 değeri ile ölçülmüş ve RepTree modeli %80,3 ile bu veri seti için en iyi algoritma olarak belirlenmiştir (Mohamed vd., 2018: 399). Moreira ve diğerlerinin (2018) çalışmasında ise sadece uçuşların geç kalıp kalmayacağı bilgisine değil ek olarak daha doğru bir tahmin yapabilmek için gerekli olan veri azaltma, dönüştürme ve dengeleme tekniklerine odaklanmışlardır. Yöntem olarak YSA, k-en yakın komşuluk, destek vektör makineleri ve rastgele ormanlar seçilmiş, bu modeller üzerinde karşılaştırmalar yapılmıştır. Veri seti olarak Brezilya yerel uçuş bilgileri 2009 ve 2015 yılları arası için kullanılmıştır. Ayrıca Weather Underground üzerinden de hava durumu ile ilgili veriler temin edilmiştir. Girdiler tarih ile ilgili veriler, planlanan kalkış varış, gerçekleşen kalkış varış, sıcaklık, basınç, rüzgar yönü, nem ve

hava durumu olarak seçilmiştir. Bu çalışma dahilinde dengeleme teknikleri kullanılan modellerin daha iyi tahminde bulunduğu ortaya koyulmuştur (Moreira vd., 2018: 1).

Yu ve diğerleri (2019) tarafından yapılan çalışmada ise havaalanı temel alınarak uçuşların ne kadar süreyle geç kaldığı ve hangi faktörlerin geç kalma üzerinde etkili olduğu araştırılmıştır. Beijing Capital uluslararası havalimanından Hangzhou Xiaoshan uluslararası havalimanına olan ve Ocak 2017 - Mart 2018 tarihleri arasında gerçekleşen 528.471 uçuş modelin veri setini oluşturmuştur. Girdi değişkenleri olarak ise kalkış havalimanı, uçak büyüklüğü, kalkış opsiyonu, uçuş yolu durumu, bir önceki uçuşun geç kalması, iki uçuş arası süre, uçuş rotalarının sayısı, planlanan uçuşların sayısı, pazar yoğunluğu, hava trafik kontrolü, planlanan uçuşlardaki yoğunluk, gerçekleşen uçuşlardaki yoğunluk, uçuşa giriş zamanı ile planlanan kalkış arasındaki zaman, yolcuların kalabalık durumu, bir uçaktaki yolcu sayısı, uçak kapasitesi, kapı kapanış zamanı, kargo kapısı kapanış zamanı, servis aracı ve jet körüklerinin hazırlık zamanı belirlenmiştir. Derin inanç ağları yöntemi kullanılmıştır. Geç kalmalar üzerinde en etkili faktörün hava trafik kontrolünün olduğu sonrasında ise bir önceki uçuşun durumu olduğu ortaya koyulmuştur (Yu vd., 2019: 203).

Bir diğer çalışmada ise girdi olarak yolcu davranışlarına ilişkin veriye ulaşılamadığı için veriler 25.000 tekrar ile simüle edilmiş ve bu şekilde temin edilmiştir. Amaç ise daha önce çalışılmamış bir alan olan uçağa binış aşaması operasyonunu tahmin etmek olarak belirlenmiştir. Girdi olarak farklı simülasyon denemelerinden elde edilen veriler kullanılmış ve tahmin sapmaları denemeler ile %75'e kadar indirilmiştir (Schultz ve Reitmann, 2019: 407). Chen ve Li'nin (2019) çalışması ise birbirine bağlı uçuşların geç kalma problemlerine odaklanmıştır. Yöntem olarak rastgele orman sınıflaması ve gecikme yayılma modu kullanılmıştır. Bureau ulaşım istatistiklerinden ve National Oceanic Atmospheric Administration (Ulusal Okyanus Atmosfer İdaresi) aracılığı ile temin edilen 2016 Temmuz - 2017 Haziran ayları arası veriler girdi olarak kullanılmıştır. Girdi olarak kullanılan değişkenler ise tarih bilgileri, planlanan kalkış, planlanan varış, planlanan uçuş süresi, hangi grupta geç kaldığı (15 dakikalık gruplar halinde), varışın gecikmeli olup olmadığı, saatlik görüş mesafesi, saatlik hava durumu, sıcaklık, nem, donma noktası, karşılaştırmalı nem, rüzgar hızı, fırtına hızı ve basınç olmuştur. Sonrasında bu veriler içinden model için en uygun olanlar geri dönüşümlü değişken eliminasyonu algoritmasıyla belirlenmiştir. Model her iki girdi kümesiyle de çalıştırılmıştır. Eliminasyon algoritması kullanımıyla da geç varışlar için doğruluk değeri 0,93 geç kalkışlar için ise 0,90 olarak elde edilmiştir (Chen ve Li, 2019: 1).

Mangortey'in (2019) çalışmasında ise yer hizmetleri programlarının uçuş ve havalimanı üzerindeki etkilerinin modellenip azaltılması amaçlanmıştır. Veriler üç farklı kurumdan temin

edilmiştir bunlar; Traffic Flow Management System (Trafik Akış Yönetim Sistemi, TAYS), Aviation System Performance Metrics (Havacılık Sistemi Performans Ölçütleri, HSPÖ) ve Automated Surface Observing Systems (Otomatik Yer Gözlem Sistemleri, OYGS) olmuştur. Girdiler için TAYS üzerinden yer operasyonlarının başlama ve bitiş zamanları, gecikme sebepleri, detaylı gecikme bilgileri ve maksimum geç kalma süreleri temin edilmiştir. HSPÖ aracılığı ile planlanan uçuşların durumu ile ilgili veriler temin edilmiştir. OYGS ise tarih ve zaman bilgileri, hava sıcaklığı, donma noktası, nispi nem, rüzgâr yönü, rüzgar hızı, yağış miktarı, basınç altimetresi, deniz seviyesi basıncı, görüş mesafesi, rüzgar sürati, bulut tipi ve bulut yüksekliği verileri temininde kullanılmıştır. Çoklu doğrusal regresyon, regresyon ağaçları, yapay sinir ağları, destek vektör makineleri, naive bayes, ve rastgele ormanlar yöntemleri kullanıştır. Rastgele ormanlar yönteminin uçuşların yer operasyonlarına bağlı geç tahmininde, yer operasyonlarındaki gecikmelerin tahmininde ve iniş taksi yolu sürelerinin yer operasyonlarına bağlı olarak gecikmelerinin tahmininde en iyi sonuç verdiği değerlendirilmiştir (Mangortey, 2019: 1).

Farklı bir amaçla kurgulanmış bir çalışmada ise uçuş gecikme problemlerinde sıkça karşılaşılan sınıf dengesizliği problemi t-dağıtılmış stokastik komşu gömme ile incelenmiş ve görsel olarak karşılaştırılmıştır. Atlanta Hartsfield–Jackson Uluslararası Havalimanı’nda 2017 yılında gerçekleşen tüm uçuşlar veri setini oluşturmuştur. Girdi değişkenleri ise tarih, uçuş numarası, kalkış-varış havalimanı, kalkış zamanı, varış zamanı, mesafe, kalkış gecikmesi (15 dakikayı geçip geçmediği) olarak belirlenmiştir. Elde ettikleri bulgulara göre geç kalan ve kalmayan uçuşların sınıf dengesizliği probleminden ziyade her iki örneklem kitlesinin birbirine yüksek ölçüde benzemesi daha büyük bir sorun olarak ortaya koyulmuştur (Mustapha vd., 2019: 1). Uçuş gecikmelerine odaklanan bir diğer çalışmada ise 15 dakikadan daha kısa zaman dilimlerindeki gecikmeleri tahmin etmek amaçlanmıştır. Kullanılan veri seti Avrupa’da faaliyet gösteren düşük maliyetli iki havayolu firmasına aittir. Girdiler; uçuş tarihi, planlanan ve gerçekleşen kalkış ve varış, kalkış ve varış havalimanları, çocuk ve yetişkin sayısı, toplam bagaj sayısı ve ağırlığı, toplam tekerlekli sandalye kullanıcı sayısı; ek olarak tahmini olarak dönüş gecikmesi, bir önceki uçağın gecikmesi, bir sonraki planlanan kalkışa kalan zaman olarak seçilmiştir. Yöntem olarak uzun kısa süreli bellek yöntemi tercih edilmiştir (McCarthy vd., 2019: 9541).

Uçuşların geliş yönünde geç kalmalarına odaklanan bir diğer çalışma ise yöntem olarak rastgele ormanlar, destek vektör makineleri, gradyan artma ve k-en yakın komşuluk yöntemlerini karşılaştırmıştır. Veriler Bureau Ulaşım İstatistikleri üzerinden toplanmış ve 2015-2016 yılları için en yoğun beş Amerikan havaalanı ile sınırlandırılmıştır. Girdi

değişkenleri olarak tarih bilgileri, uçuş numarası, kalkış havalimanı, varış havalimanı, kalkış alan kodu, varış alan kodu, varış zamanı, kalkış zamanı olarak belirlenmiştir. Eğitim aşamasında en yüksek doğruluk değerini (%99,99) rastgele ormanlar yöntemi vermiş olsa dahi, test sonuçlarına göre gradyan artma yöntemi en yüksek doğruluk değerini (%79,7) sağlamıştır (Chakrabarty vd., 2019: 651). Bureau Ulaşım İstatistikleri verisini kullanan bir diğer çalışma ise lojistik regresyon, karar ağaçları regresyonu, bayesian ridge, rastgele ormanlar regresyonu ve gradyan artma regresyon yöntemlerini bir arada kullanmıştır. Girdi değişkenleri tarih bilgileri, havayolu adı, uçuş numarası, kalkış havaalanı, varış havaalanı, planlanan kalkış zamanı, gerçekleşen kalkış zamanı, kalkış gecikmesi, kalkış yönünde taksi yolu, tam havalanma, planlanan havalanma, uçuş süresi, mesafe, uçağın indiği an, iniş yönünde taksi yolu, planlanan varış, varış gecikmesi, iptal edilen uçuşların içinden seçilmiştir. Gün, kalkış gecikme süresi, havayolu, uçuş numarası, varış havaalanı, kalkış havaalanı, haftanın günü ve kalkış yönünde taksi yolu zamanı ise kullanılmak üzere seçilen değişkenler olmuştur. Kalkış gecikmelerinde Rastgele Orman düzenlemesi ile hata kareler ortalamasına (2261,8) ve ortalama mutlak hataya (24,1) göre en iyi sonucu vermiştir. Varış gecikmelerinde ise yine Rastgele Ormanlar yöntemi hata kareler ortalamasına (3019,3) ve ortalama mutlak hataya (30,8) göre en iyi sonucu vermiştir (Meel vd., 2020: 75).

Dou'nun (2020) çalışmasında uçuş geç kalması ilk defa CatBoost yöntemi ile Amerikan Ulaşım Yönetimi tarafından yayınlanan veriler ile birlikte uygulanarak kullanılmıştır. Veri setinin tarih aralığı Ekim 2018 tarihinden Eylül 2019 tarihine kadar belirlenmiş, girdi değişkenleri ise tarih bilgileri, havayolu firması, kuyruk numarası, kalkış havalimanı numarası, kalkış şehri, varış havalimanı numarası, varış şehrinin adı, planlanan kalkış zamanı, planlanan varış zamanı, uçuş süresi ve mesafe olarak seçilmiştir. Girdiler için önem sıralaması yapıldığında sıralama önem değeri yüksek olan değişkenler ay, ayın günü, haftanın günü, uçuş süresi ve mesafe olarak ortaya çıkmıştır. Kurulan en iyi model %80,4 doğruluk oranı ile sonuç vermiştir (Dou, 2020: 836). Yapılan diğer çalışmalardan farklı olarak oldukça uzun süreli veriye dayanarak yapılan bir diğer uçuş gecikmesi çalışması ise Patrgiri ve diğerleri (2020) tarafından yapılmıştır. Veri seti Amerikan İstatistik Derneğinden alınmış ve 1987-2008 yılları arası gerçekleşen tüm uçuş verileri kullanılarak tahmin modeli geliştirilmiştir. Hava durumu verileri ise National Weather Services National Digital Forecast Database (Ulusal Hava Durumu Hizmetleri Ulusal Dijital Tahmin Veritabanı) aracılığı ile elde edilmiştir. Girdi değişkenleri ise tarih bilgileri, havayolu firması, kalkış gecikmesi, kalkış yeri, varış yeri, mesafe, maksimum ve minimum sıcaklık, maksimum ve minimum görüş mesafesi, ortalama rüzgar hızı, yağış miktarı, bulutluluk durumu olarak belirlenmiştir. Lojistik regresyon, naive bayes, k-en yakın

komşululuk, karar ağacı ve rastgele ormanlar yöntemleri uygulamada kullanılmıştır. En yüksek doğruluk oranını %82 ile rastgele ormanlar modeli vermiştir (Patgiri vd., 2020: 1).

Yang ve diğerlerinin (2020) yaptığı çalışmada ise amaç varış ve kalkış zamanlarındaki ile kalkış ve iniş yönünde taksit yolu zamanlarındaki geç kalma sürelerinin tahmin edilmesi ile elde edilecek faydaların belirlenmesi olmuştur. Amacı gerçekleştirmek için de YSA ve Tekrarlayan Sinir Ağları önerilmiştir. Girdi olarak farklı kaynakların zayıflık ve üstünlük analizleri yapılmış sonra bu kaynaklar birleştirilmesi önerilmiştir. Bahsi geçen veri kaynakları Traffic Flow Management System (Trafik Akış Yönetim Sistemi), Performance Data Analysis and Reporting System (Performans Veri Analizi ve Raporlama Sistemi), Air Traffic Operations Network (Hava Trafik Operasyon Ağı), Airline Service Quality Performance (Havayolu Hizmet Kalitesi Performansı), Aviation System Performance Metrics (Havacılık Sistemi Performans Ölçütleri) ve Bureau of Transportation Statistics (Ulaştırma İstatistikleri Bürosu) olmuştur. Girdi değişkenleri ise tarih bilgileri, kalkış ve varış havalimanları, planlanan kalkış ve varış zamanı, kalkışta ve varışta havalimanında gerçekleşen geç kalma durumu, uçuş numarası, rota kodu, rüzgar hızı, bulut yüksekliği, görüş mesafesi, yağış miktarı ve birikimi, genel hava durumu, ölçüm kodu, havalimanı konumu, planlanan ve gerçekleşen kapı kapanış zamanı, planlanan ve gerçekleşen pisten ayrılma zamanı, planlanan ve gerçekleşen piste iniş zamanı, planlanan ve gerçekleşen kapı açılış zamanı olmuştur. Böyle bir platform kurulması ile elde edilecek faydalar belirlenmiştir (Yang vd., 2020: 6).

Uzun kısa süreli hafıza kullanan bir başka güncel çalışma ise Gui ve Yang (2020) tarafından yapılmıştır. Rastgele orman bazlı model de önerilmiştir. Rastgele ormanlar ile geç kalacak kalınmayacak kararı verirken gecikme durumuna göre üç ayrı gruba ayrılmış ve bu gruplara göre tahmin edilmiştir. Gecikme olmama durumu ile gecikmeler bir saat ve iki saatlik olarak gruplanmıştır. Gecikme olup olmama durumu için yapılan rastgele orman sınıflandırmasında %90,2 doğruluk oranı elde edilmiştir. Bir saatlik gecikme için doğruluk oranlarını %81,4 iki saatlik gecikme için %70 olarak elde edilmiştir (Gui ve Yang, 2020: 147). Bir başka çalışmada ise 300 kilometrekarelik alan içindeki uçakların sürekli olarak sinyallerini alan bir sistem aracılığı ile veri elde edilmiştir. Bu veriler uçuş numarası planlanan kalkış/varış zamanı, hava durumu, rüzgar yönü ve hızı, havalimanı adı, kalkış ve varış havalimanı ve hava trafik akışı olmuştur. Bir önceki çalışmada olduğu gibi geç kalma durumu üç gruba ayrılmış ve geç kalıp kalmama durumu %87,72; 1 saatlik geç kalma %79,45; 2 saatlik geç kalma durumu da %67,36 doğruluk oranı ile tahmin edilmiştir. Kullanılan makine öğrenmesi yöntemi ise gradyan artma temelli karar ağacı yöntemi olmuştur (Liu vd., 2020: 1).

İncelenen yöntemler kullanılan makine öğrenme yöntemlerine göre Tablo 2.1’de özetlenmiştir. En çok başvurulan yöntemler karar ağaçları ile rastgele ormanlar olarak tespit edilmiştir. Ardından yapay sinir ağları ile yapılan çalışmalar ve sonrasında gradyan artma yöntemi gelmektedir.



**Tablo 2.1 Uçuş Gecikmesi Probleminde Kullanılan Makine Öğrenmesi Yöntemleri**

	RO	KA	GA	YSA	K-EYK	LR	DVM	AB	UKSB	NB	EA	TÇÇD	DİA	DS	ÇDR	BR	CB
(Rebollo ve Balakrishnan, 2014)	X																
(Chopin, vd., 2016)	X	X															
(Kim, vd., 2016)									X								
(Belcastro, vd.,2016)		X															
(Manna, vd., 2017)			X														
(Choi, vd., 2017)	X	X			X			X									
(Takeichi, vd., 2017)				X													
(Thiagarajan, vd., 2017)	X		X	X	X			X			X	X					
(Kuhn ve Jamadagni 2017)	X	X	X	X		X		X			X	X					
(Venkatesh, vd., 2017)				X									X				
(Mogha, vd., 2017)	X	X					X			X							
(Mohamed, vd., 2018)	X	X		X		X								X			
(Moreira, vd., 2018)	X			X	X		X										
(Yu, vd., 2019)													X				

**Tablo 2.2 Uçuş Gecikmesi Probleminde Kullanılan Makine Öğrenmesi Yöntemleri (devam)**

	RO	KA	GA	YSA	K-EYK	LR	DVM	AB	UKSB	NB	EA	TÇÇD	DİA	DS	ÇDR	BR	CB
(Chen ve Li, 2019)	X																
(Mangortey, 2019)	X	X		X			X			X					X		
(Mustapha vd., 2019).									X								
(McCarthy, vd., 2019)	X		X		X		X										
(Meel, vd., 2020)	X		X			X										X	
(Dou, 2020)																	X
(Patgiri, vd., 2020)	X	X			X	X				X							
(Gui ve Yang, 2020)	X								X								
(Liu, vd., 2020)			X														
<b>Toplam</b>	14	8	6	7	5	4	4	3	3	3	2	2	2	1	1	1	1

## 2.2.2. Uçuş Gecikmeleri Tahmininde Kullanılan Diğer Yöntemler

Uçuş gecikmeleri tahmini için literatürde makine öğrenmesi dışında başka yöntemler de kullanılmıştır. Bu çalışmalar regresyon, olasılık, yöneylem araştırması ve şebeke temelli yöntemler olarak derlenmiştir.

### 2.2.2.1. Regresyon Modelleri

Uçuş gecikmelerini regresyon modelleri ile tespit eden çalışmalar da literatürde yer almaktadır. Frankfurt havalimanındaki günlük dakikliği incelemek üzere hibrit bir regresyon ve zaman serisi modeli kullanılmıştır. Modelde %60-69 oranında değişkenlik açıklanmıştır (Markovic vd., 2008: 293). Bir başka çalışmada çok değişkenli regresyon modeli ile gecikmelerin tahmini yapılmıştır ve 5,3 dakikalık doğruluk ortaya konulmuştur (Xu vd., 2008: 62). Bulanık doğrusal regresyon da kullanılan yöntemler arasındadır (Zhang vd., 2010: 109). Bir diğer çalışmada ise doğrusal regresyon modeli ile çalışılmıştır. Regresyon modeli ile Naive Bayes ve karar ağacı yöntemleri ile kurulan modeller karşılaştırılmıştır (Ding, 2017: 1). Çoklu regresyon modeli de uçuş sistemindeki gecikmeleri tahmin etmek üzere kullanılmıştır (Oza vd., 2015: 11765).

### 2.2.2.2. Olasılık Modelleri

Olasılık temelli modeller de uçuş gecikmeleri tahmininde kullanılmıştır. Geçmişe dayalı olay bazlı beklentiyi maksimize etme amaçlı olasılıksal model ile genetik algoritmalar birleştirilmiş ve Denver Uluslararası Havaalanındaki gecikmelerin dağılımı tahmin edilmiştir (Tu vd., 2008: 112). Yaşam analizi ile Cox modelini birleştiren bir başka çalışma da olasılık modeli ile uçuş gecikmeleri çalışılmıştır (Wong ve Tsai, 2012: 5). Sonuçlarında ise kalkış gecikmesi ile varış gecikmelerinin yakından ilişkili olduğu, uçuş gecikmelerinin stokastik olduğu ve tehlikeli oranları veren dönemler için uçak gecikmelerinden kaçınmak için kullanılabileceği ortaya konmuştur.

### 2.2.2.3. Yöneylem Araştırması Modelleri

Yöneylem yöntemleri kullanarak da uçuş gecikme problemleri çalışılmıştır. Simülasyon ve uçuş gecikmelerini maliyet kalemi olarak hesaplayarak optimize eden dinamik optimizasyon modeli uygulaması literatürde yer almaktadır (Jungai ve Hongjun, 2012: 348). Bir başka çalışmada uçak rotalarını ve uçuş kalkış zamanlarını optimize ederek yolcu gecikmelerini en aza indirmek amaçlanmıştır. Stokastik olarak üretilen girdilerle karışık tamsayı programlama ile gecikme yayılmasını önlemek için yöntem önerilmiş sonrasında yetersiz bağlantı süresi olan

yolcular için algoritmik bir yaklaşım sunulmuştur (Lan vd., 2006: 15). Bir diğer çalışma ise tam sayılı programlama problemine dönüştüren uçuş planlama sistemini çok amaçlı optimizasyon problemine dönüştürülerek çözüm önerisi sunmuştur. Uçuş gecikmeleri konusunda %12 ile %31'e kadar iyileşme olacağı tespit edilmiştir (Chen vd., 2020: 853). Uçuş gecikmeleri için farklı bir yöntem ise çeşitli senaryolara uygulanabilecek doğrusal programlama ve Lagrangian ikili ayrıştırılmalı dal ve sınır algoritması ile önerilmiştir (Eun vd., 2010: 359). Uçuşların birbirini etkiler yapıda olduğunu ortaya koyan ve buna göre ilk gecikmelerin etkisini en aza indirmek için gecikme yayılımı kullanımını da literatürde mevcuttur (Beatty vd., 1999: 259). Ulusal Hava Sahası Sistemindeki gecikmeleri ve gecikmelerin yayılmasını etkileyen kontrol edilebilir faaliyetleri ayıran analitik bir model sunulmuştur (Wang vd., 2003: 5). Bu modelde yinelemeli uçuş gecikmesi yayma modelini kullanmışlardır.

#### **2.2.2.4. Şebeke Temelli Modeller**

Şebeke temelli modeller de uçuş gecikmeleri probleminde başvurulan yöntemlerden biri olmuştur. Güvercin vd. (2020) çalışmalarında rotalara üzerindeki yerleşimlere ve gecikmelere dayanarak kümelenmiş havaalanı modellenmesi yapmışlardır. Böylece ağ üzerindeki uçuşların optimizasyonu çalışmışlardır. Başka bir ağ yapısına odaklanan çalışma ise uzay- zaman üzerinden bir ağ kurarak çok ürünlü minimum maliyetli bir akış problemi olarak sistem tasarımı yapmıştır (Helme, 1992: 236). Ağın tümünü incelemek üzere yapılan bir çalışmada uçuş sürelerini ve kaynakların kullanılabilirliği göz önünde bulundurulmuştur. Bu veriler ile grafik oluşturulmuş ve en kısa yol algoritmasına başvurularak şebeke temelli bir model oluşturulmuştur (Abdelghany vd., 2004: 385). Ağ temelinde yapılan bir diğer çalışmada Granger nedenselliği kullanılmış ve havalimanları düğümler olarak ele alınarak ve gecikme yayılımları incelenmiştir. (Zanin vd., 2017). Granger nedenselliği kullanımı biraz daha farklılaştırılarak bir çalışmada daha yöntem olarak tercih edilmiştir (Xiao vd., 2020: 97103).

#### **2.2.3. Literatürün Çalışmaya Katkısı**

Literatür taraması esnasında makine öğrenmesi teknikleri ve kullanılan diğer yöntemlerin problem çözümü için geliştirdikleri çözümler incelenmiştir. Güncel çalışmalarda makine öğrenmesi teknikleri ile uçak gecikmeleri tahmininin yer aldığı görülmüştür. Bu çalışmaların ise genellikle aynı veri seti kullanılarak yapılmış olduğu, farklı veri seti kullanılan çalışmalarda ise yine havalimanı verisi kullanılmış olduğu tespit edilmiştir. Bu çalışmada ise kullanılan veri seti bir havayolu firmasından temin edilmiş ve literatüre farklı bir veri seti ile katkıda bulunulmuştur.

Literatürde makine öğrenmesi ile uçak gecikmelerine odaklanan çalışmalar Tablo 2.1’de de özetlenmiştir. Bu tabloya göre ise hangi sıklıkta hangi yöntemlerin kullanıldığı görülmüş burada yer alan tekniklerden ve bunların dışındaki tekniklerden faydalanarak literatüre göre sonuçları değerlendirme imkanı elde edilmiştir.

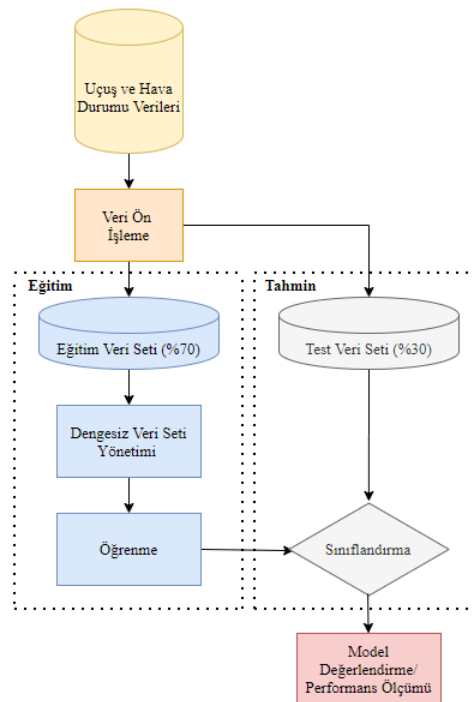
## ÜÇÜNCÜ BÖLÜM

### UÇUŞ GECİKMELERİNİN TAHMİNİ

Uçuş gecikmelerinin sektör, kullanıcılar ve çevre üzerindeki olumsuz etkileri ikinci bölümde ortaya konulmuştur. Buna göre çalışmanın amacını oluşturan geciken uçuşların tahmininin önemi belirtilmiştir. Bu duruma istinaden uçakların durumunun önceden tahmin edilmesi amaçlanan bu çalışmada, bir havayolu firmasının uçuş gecikmeleri farklı makine öğrenmesi yöntemleri kullanılarak modellenmiş böylece uçakların planlanan saate göre kalkış durumları tahmin edilmiştir. Bu bölümde ise öncelikle uygulanacak olan makine öğrenmesi tekniklerinin kullanımı için takip edilen metodoloji ortaya konulmuştur. Bu kapsamda önce veri ön inceleme aşaması, eğitim aşaması ve son olarak değerlendirme işlemleri tanıtılmıştır. Devam eden bölümde uygulamada kullanılan veri seti tanıtımına ve son bölümde ise araştırmanın bulgularına yer verilmiştir.

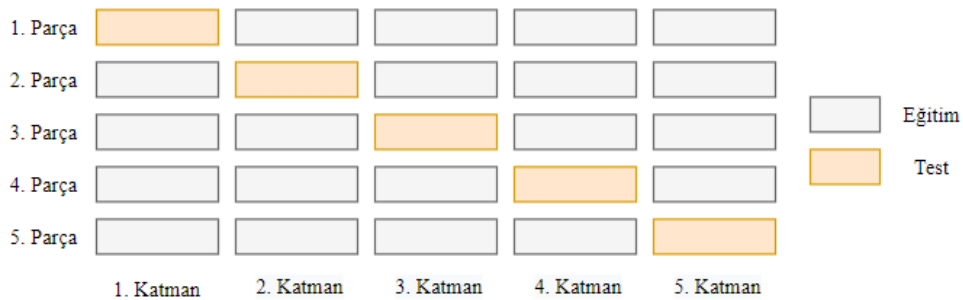
#### 3.1. Metodoloji

Belirlenen probleme farklı makine öğrenme yöntemleri ile yanıt aranmakta olsa da çalışmanın metodolojisi genel hatları ile Şekil 3.1 üzerinde gösterilmiştir. Geçmiş yıllara ait uçuş ve hava durumu verileri toplanarak birleştirilmiştir. İlgili veri setindeki eksik verilerin düzenlenmesi uygulamanın ilk aşamasını oluşturmuştur.



Şekil 3.1 Çalışmanın Metodolojisi

Sonrasında makine öğrenmesi yöntemlerine eğitim için girdi olarak verilmek üzere veri setinin %70'i eğitim seti olarak rastgele bir şekilde ayrılmıştır. Bu şekilde kullanım literatürde sıklıkla başvurulan bir yöntem olsa da öğrenme aşamasında kullanılan bir başka uygulama ise çapraz doğrulamadır (cross validation). Çapraz doğrulama modelin sadece eğitim ve test şeklinde ayrılmış halinin incelenmesine göre daha istikrarlı sonuç veren bir istatistiksel yöntemdir (Kohavi, 1995: 1137). Bu sebeple makine öğrenmesi teknikleri denenirken son aşamada çapraz doğrulamaya da yer verilmiştir. Sebebi yöntemin, modelin göremediği veri üzerinden birden çok kez denenmesini mümkün kılmasıdır. Bu yöntemde tüm veri seti kullanımı yerine veri seti öncelikle bölümlere ayrılmaktadır. Elde edilen bölümler üzerinden model çok sayıda eğitim aşamasını tamamlamaktadır. Genellikle  $k$ -katmanlı çapraz doğrulama kullanılmaktadır (Müller ve Guido, 2017: 252). Burada  $k$  değeri kullanıcı tarafından belirlenmekte; belirlenen değere göre ise veri eş parçalara bölünmektedir (James vd., 2017: 181). Şekil 3.2'de gösterildiği gibi  $k$  değerinin 5 olarak belirlendiği bir veri seti için 5 katman ve çapraz doğrulama yapmak için de 5 parça oluşturulmuştur. Elde edilen parçalar üzerinde test verisi değiştirilerek işlem yapılmaktadır. Böylece modelin geçerlik durumu pekiştirilmektedir.



**Şekil 3.2 K-katmanlı Çapraz Doğrulama Veri Seti Örneği**

Geciken uçular ile gecikmeyen uçuların eş sayıda olmaması durumu dengesiz veri seti oluşmasına neden olduğu için dengesiz veri setleri için önerilen çözümler incelenmiştir. Öğrenme aşamasında çıktı değerleri yani uçuların geç kalıp kalmayacağı verisine sahip olduğu ve performans ölçme yöntemlerini kullanmaya imkan tanıdığı göz önünde bulundurularak denetimli öğrenme tercih edilmiştir. Eğitim tamamlandıktan sonra kalan %30'luk veri seti ile test işlemi gerçekleştirilerek modellerin öğrenmiş oldukları veriler ile sınıflandırma yapılmış ve sonuçlar tahmin edilmiştir. Son aşamada ise test sonuçlarına göre modellerin seçilen performans metriklerine dahilinde karşılaştırmaları yapılmış ve elde edilen bulgular paylaşılmıştır.

### 3.1.1. Veri Ön İşleme

Veri ön işleme aşaması genellikle makine öğrenme yöntemlerinin ilk aşamasıdır. Hatta birçok "güçlü" olarak bilinen yöntemden çok daha önemli olduğunun altı çizilmektedir (Acuña,

2011: 1). Bu durumun sebebi olarak üç ana faktör ortaya koyulmuştur. Bunlar düzenlenmemiş verilerin bazı analizlerin uygulanmasını engelleyecek yapıda olmaları, yapısı itibariyle veri ön işleme tekniklerinin eldeki veri setini daha yakından incelemeye imkan vermesi ve bazı durumlarda ise modellerden daha iyi sonuç alınabilmesi olarak belirlenmiştir (Oğuzlar, 2003: 70). Eldeki veri setinde olan problemlerin giderilmesi amacıyla eksik verilerin tamamlanması, tekrarlanan verilerin kaldırılması, dönüştürülmesi, normalleştirilmesi ve boyut indirilmesi gibi işlemleri içermektedir.

Kayıp veya eksik veriler ele alınırken eksik verilerin durumuna göre farklı uygulamalara gidilebilmektedir. Eksik verilerin veri setinden çıkarılması, sürekli veriler inceleniyorsa ortalama değerin eksik veri yerine eklenmesi, çıktı değişkeni aynı olan eksik verilerin ortalamasının alınması (Roiger, 2017: 350), kategorik veriler içinse “bilinmiyor” gibi tek bir değer eklemek mümkündür. Dönüştürme işlemi sürekli verilerin kesikli aralıklara dönüştürülmesi şeklinde olabileceği gibi kategorik verilerin sayısal değerlere dönüştürülmesi şeklinde de gerçekleştirilebilmektedir. Kategorik veri dönüştürme işlemlerinde yaygın olarak kullanılan yöntemlerden olan kategorik değerleri dönüştürme (one-hot-encoding) yönteminde girdiler tam sayılı ifadelerle dönüştürülür. Yöntemin işleyişi Tablo 3.1’de gösterilmiştir. Buna göre tabloda haftanın günlerinin sayısal ifadelerle dönüştürüldüğü görülmektedir. Haftanın günü sütunu kaldırılarak her bir gözlem gününe denk gelen gün isimlerine yer verilmiştir. İlgili günlerde planlama yapıldı ise o güne 1 değeri verilerek diğer günler için ilgili gözlem 0 değerini almıştır. Bir başka veri ön işleme tekniği verilerin normalleştirilmesidir. Bazı makine öğrenme tekniklerinde yöntem sonuçlarının uygulaması için gerekli bir aşamayı oluşturmaktadır. Çok çeşitli şekillerde gerçekleştirilebilmektedir. Sürekli verilerin doğal logaritmasının alınması veya minimum maksimum ölçekleme (minmaxscaler) gibi algoritmalara başvurulması ile gerçekleştirilebilmektedir. Boyut indirilmesi ise çok fazla değişkenin modele girdi olarak verilmesi durumlarında modelin optimizasyon işlemi gerçekleştirememesi gibi durumlarda gerekli olmaktadır. Bu tip durumlarda değişken seçimi ve faktör analizi yöntemleri ile değişken sayısı azaltılarak bu sorunun önüne geçilmektedir.

**Tablo 3.1 Kategorik Veri Dönüştürme Örneği**

Gözlem	Haftanın Günleri	Uçuş Durumu
1	Pazartesi	1
2	Salı	0
3	Çarşamba	1

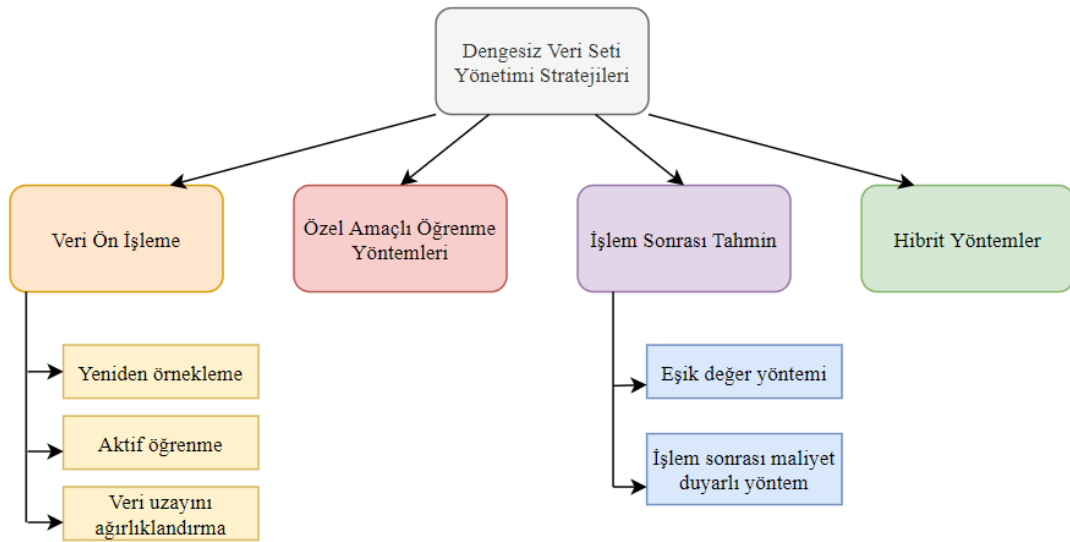
Gözlem	Pazartesi	Salı	Çarşamba	Uçuş Durumu
1	1	0	0	1
2	0	1	0	0
3	0	0	1	1



### 3.1.2. Dengesiz Veri Seti Yönetimi

Sınıflandırma problemlerinde çıktı kümesinde bulunan gözlem sayılarının birbirinden farklı olması veri setinin dengesiz veri seti olarak değerlendirilmesi anlamına gelmektedir. Sınıflandırma problemlerine has bir durumdur. İki çıktısı olan sınıflandırma probleminde bir sınıfın çoğunluk diğer sınıfın ise azınlık olması veri setinin dengesiz olmasına örnek olarak verilmektedir (Visa ve Salescu, 2005: 67). Söz konusu sınıflandırma bir hastalık durumu için yapıldığında her başvuran 1000 kişiden 100 kişinin hasta olması durumu incelendiği takdirde, başvuran kişilerin durum frekansı arasında ciddi bir fark olduğu görülmektedir. Denetimli öğrenme yöntemi ile kullanılan yöntemler veri setine göre kişilerin hasta olmadığını tahmin etmeye daha yatkın olacaktır. Zira başvuran tüm kişilerin hasta olmadığı tahmini ile dahi %90 doğruluk oranı ile tahmin gerçekleştirmiş olacaktır. Özetle bu durum tahmine dayalı model oluşturma gereksinimi olduğunda, nadir olayların ihmal edilmesine, çoğunlukta olan örneklere daha fazla odaklanılmasına neden olabilmektedir (Branco vd., 2015: 12) ve öğrencilerin kullanılabilir performansını düşürmektedir (He ve Garcia, 2009: 1263).

Literatürde dengesiz veri seti sorununu aşmak adına dört ana yaklaşım geliştirilmiştir. Bunlar veri ön işleme, özel amaçlı öğrenme yöntemleri, işlem sonrası tahmin ve hibrit yöntemler olarak belirlenmiştir. Şekil 3.3’de bu yöntemlerin genel yapısı gösterilmektedir (Branco vd., 2015: 14).



Şekil 3.3 Dengesiz Veri Seti Yönetimi (Branco vd., 2015:14)

#### 3.1.2.1. Veri Ön İşleme

Veri ön işleme dengeli bir veri setinde de sıklıkla başvurulan bir başlangıç adımdır. Fakat dengesiz veri setleri için daha önemli bir aşamayı oluşturmaktadır. Çünkü öğrenme modeli yanlı çalışarak çoğunluk olan sınıfı tahmin etmeye daha yatkın olacak ve performans

incelemesi yapıldığında da model iyi sonuçlar veriyormuş gibi yanıltıcı yönlendirmeler olacaktır (Türk ve Ozkan, 2015: 268). Dengesiz veri setleri için ön işleme yöntemleri yeni veri üretimi, aktif öğrenme ve veri uzayını ağırlıklandırma olarak üç başlıkta incelenmektedir.

### 3.1.2.1.1. Yeniden Örnekleme

Yeni veri üretimi stratejileri veri setindeki azınlık sınıfın dezavantajını ortadan kaldıracığı için efektif bir çözüm olarak gösterilmektedir (Gangawar, 2012: 43; Cao vd., 2019, s. 1567). Veri dağılımını değiştirmenin sonuçlarının pozitif ve negatif örneklerin arasındaki oran ve sınıflandırıcının doğası gibi bir dizi faktöre bağlı olduğu birçok veri üretim yönteminin denendiği bir çalışmada ortaya konmuştur (Hulse vd., 2007: 941). Bazı durumlarda dengeli bir dağılım olan veri setlerinin de iyi sonuç veremediği ortaya koyulmuştur (Weiss ve Provost, 2003: 326). Bu sebeple doğru miktarda veri üretimi yapabilmek için çok çeşitli yöntemler önerilmiştir (Chawla vd., 2005: 32). Veri setine göre uygun yöntemin seçilmesi en doğru yaklaşımdır. Bu da genellikle deneme yanılma yöntemi ile bulunabilmektedir. Yeniden örnekleme ile dengesiz veri setinden dengeli bir veri seti elde etmek için başvurulmakta olan yöntemler alt örnekleme (undersampling), aşırı örnekleme (oversampling) ve sentetik veri üretimi olarak incelenmektedir.

Alt örnekleme, girdilerin boyutunu küçülterek orijinal veri kümesinden kaldırmak temelinde geliştirilmiştir. Alt örneklemede, rastgele seçilen çoğunluk sınıfı üyeleri veri setinden atılır. Böylece performansı düşüren örneklerin ortadan kalkması mümkün olabilmektedir. Bu durum modelin yeterli uyuma ulaşamaması problemine yol açabilmektedir. Aşırı örnekleme ise örnek boyutunu artıracak veriler eklemek temeli ile geliştirilmiştir. Örneğin, rastgele aşırı örnekleme yönteminde azınlık sınıfı üyelerinin rastgele kopyaları veri setine eklenmektedir, fakat bu durum aşırı uyumlama olasılığını da arttırabilmektedir (Chawla vd., 2002: 326). Rastgele seçilen gözlemler yerine uzaklık temelli seçimler yapan algoritmalar da mevcuttur (Mani ve Zhang, 2003: 1). Bu yaklaşımın temelinde eğitim setine dahil edilecek çoğunluk sınıfının hangileri olacağına belli bir mesafe (k-en yakın komşuluk gibi) kriterine göre karar olarak alt örnekleme gerçekleştirilir. Benzer bir şekilde Tomek linkleri yöntemi de kullanılmaktadır. Bu yöntemde birbirlerinin yakın komşusu olan fakat farklı etiketlere sahip olan örnekleri hedef alan yöntem yalnızca çoğunluk sınıfa ait örnekleri kaldırarak alt örnekleme gerçekleştirir veya her iki sınıf için de uç örnekleri kaldırabilir (Batista vd., 2004: 20). Başka bir yaklaşım ise benzerlik skoruna göre bir sınıflandırma yaparak ulaşılmak istenen değer ile eldeki değerler arasına sınır çekilmesidir. Burada önemli olan belirlenecek eşik değeridir, çünkü

eğer eşik değeri düşük tutulur ise azınlık sınıfından örnekler de göz ardı edilir, aksi halde de çoğunluk sınıfından çok fazla örnek modele dahil edilir (Branco vd., 2015: 17).

Bir diğer yöntem ise sentetik veri üretimidir. Genellikle sınıflandırma problemleri için kullanılan bu yöntemler veri setine yeni örnek olarak eklendiklerinden dolayı aşırı uyum riskini azaltırlar ve aşırı örnekleme yöntemlerinin aksine genelleme yeteneğini geliştirirler (Branco vd., 2015: 18). Branco vd. (2015) göre sentetik azınlık aşırı örnekleme yöntemi (Synthetic Minority Over-Sampling Technique – SMOTE) olarak tanımlanan yöntem en popüler tekniklerden biridir. Chawla vd. (2002) tarafından geliştirilen yöntem azınlık sınıfı gözlemlerini kullanarak k-en yakın komşular tekniği ile sentetik veri oluşturup bu verileri azınlık sınıfına dahil edilmesi temelinde çalışmaktadır (He ve Ma, 2013: 47). Sentetik veri üzerine çalışan bir başka yöntem ise rastgele aşırı örneklemedir (Menardi ve Torelli, 2014: 92). Eldeki veri seti üzerinden yetersiz ve aşırı örnekleme yaparak yeni veri seti üretme temelinde çalışmaktadır. Diğer yaygın sentetik veri yöntemleri: uyarlanabilir sentetik örnekleme, çoğunluk ağırlıklı azınlık örnekleme, hızlı yakınsayan Gibbs algoritması, rastgele yürüyüş aşırı örneklemedir (Aydın, 2021: 4).

Yeniden örnekleme yöntemlerine yapılan bir eleştiri ise sonuçlarının değişkenlik gösterdiği her durumda başarılı sonuç vermediği yönündedir. Bu durum deneysel çalışmalar yapılarak gösterilmiştir (Jo ve Japkowicz, 2004: 40).

#### **3.1.2.1.2. Aktif Öğrenme**

Aktif öğrenme makine öğrenme yöntemlerinden biridir. Genellikle çıktı verilerini toplamanın maliyetli olduğu durumlarda kullanılmaktadır. Çıktıları istatistiksel tabanda tahmin ederek bulmaktadır. Çıktı verisinin olduğu durumda ise gözlemler içinden en bilgilendirici gözlemleri seçerek alt örnekleme gerçekleştirmek için kullanılabilir (Branco vd., 2015: 21).

#### **3.1.2.1.3. Veri Uzayını Ağırlıklandırma**

Bu yöntem maliyet duyarlı bir temelde çalışılmaktadır. Veri setinde yanlış sınıflandırılan gözlemlere ceza uygulanır ve böylece yanlış gözlemler maliyet minimizasyonu temelinde azaltılmaya çalışılmaktadır (Zadrozny vd., 2003: 435). Basit ve uygulaması kolay olarak tanımlanan bu tekniğin en önemli dezavantajı bazı durumlarda aşırı uyum ve bazı durumlarda ise maliyet değerlerinin mevcut olmaması olarak gösterilmektedir (Branco vd., 2015: 22).

### 3.1.2.2. Özel Amaçlı Öğrenme Yöntemleri

Bu yaklaşımda eldeki veriye göre algoritmalar yeniden düzenlenmektedir. Öğrenme aşamasında modelin karşılaşacağı dengesiz veri ile başa çıkmak için algoritma tercih kriterlerinde değişiklik yapılır. Böylelikle dengesiz veri seti sebebi ile karşılaşılabilecek yanlış öğrenme gibi sorunların önüne geçmek amaçlanmaktadır. Fakat bu özellikleriyle, kullanılan öğrenme algoritması hem de veri seti üzerinde nitelikli bilgi sahibi olmayı gerektiren bir yöntemdir. Bu yöntem tasarlandığı veri seti için çok etkili olabilmekle beraber uygulanması deneyim ve nitelikli bilgi birikimi gerektirmektedir (Branco vd., 2015: 26).

### 3.1.2.3. İşlem Sonrası Tahmin

Bu yöntem iki aşama üzerine kurulmuştur. İlk aşamada sınıflandırma tekniklerinden biri kullanılarak bir eşik değeri elde edilir sonrasında bu eşik değere göre sınıflandırma tekrarlanmaktadır. İkinci aşamada ise maliyete dayalı yöntemle tahmin hataları ceza değerleri belirlenmektedir. Beklenen maliyeti en aza indirecek sınıflandırma işlemi ile yöntem tamamlanmaktadır (Branco vd., 2015: 25).

### 3.1.2.4. Hibrit yöntemler

Hibrit yöntemler ile yeni veri üretiminin, özel amaçlı öğrenme yöntemlerinin veya işlem sonrası tahmin yöntemlerinin avantajlı yanları bir araya getirerek daha dengeli bir veri seti elde etmek amaçlanır. Bu konuda yapılmış çalışmalar incelendiğinde çok farklı tekniklerin bir araya getirilmiş olduğu görülmüştür. Bu alandaki çalışmalardan birinde Tomek bağlantıları ile SMOTE tekniklerinin ve RUSBoost ile topluluk öğrenme yöntemlerinin bir araya getirilerek performans denemelerine yer verildiği görülmüştür (Zhang vd., 2018: 20). Bir başka çalışma SMOTE ile yetersiz örnekleme yöntemini birleştirmiştir ve normale göre daha iyi sonuç verdiği gösterilmiştir (Salunkhe ve Suresh, 2016: 732). Bu şekilde birden fazla tekniğin bir arada kullanılması hibrit yöntemler olarak adlandırılmaktadır.

### 3.1.3. Öğrenme

Makine öğrenmesi yöntemleri kullanılırken, makine tarafından veriden öğrenilen veya tahmin edilmek için kullanılan parametrelerin yanı sıra bir de kullanıcı tarafından ayarlama yapılabilecek parametreler bulunmaktadır (Tanyıldızı ve Demirtaş, 2019: 1). Hiper-parametreler kullanıcı tarafından düzenlenebilen parametreler olarak bilinmektedir. Örnek olarak yapay sinir ağları incelendiğinde, algoritmada belirlenen nöron sayısı, katman sayısı veya kullanılan aktivasyon fonksiyonu gibi kararlar hiper-parametre ayarlarını oluşturmaktadır (Mantovani vd., 2016: 37). Bu ayarlamalar makine öğrenmesi algoritmalarının tahmin

yeteneğini çok büyük ölçüde etkilemektedir (Feurer vd., 2015: 1). Hiper-parametre ayarları yapmak için sıklıkla deneme yanılma temelinde uygulamalar yapıldığı görülmektedir, fakat eğitim süresi kısa olan modeller için dahi deneme yanılma yöntemini uygulamak uzun zaman alabilmektedir (Mantovani vd., 2016: 37). Yakın zamana kadar bilgisayar donanımları bu kadar gelişmiş olmadığı için, geliştirilen tarama yöntemleri ile yeterince iyi sonuçlar almak pek de mümkün olmamaktaydı (Bergstra vd., 2011: 2546). Basit makine öğrenmesi yöntemleri üzerinde daha titiz gerçekleştirilen hiper-parametreler ile yenilikçi makine öğrenme yöntemleri ile karşılaştırılmıştır. Model çözüm sürelerinin hemen hemen aynı olduğu görülmüştür. Buna karşın basit modellerin titiz hiper parametre ayarlarıyla yenilikçi makine öğrenme yöntemlerine göre daha iyi sonuç verdikleri ortaya koyulmuştur (Pinto vd., 2009: 7; Coates vd., 2011: 222; Coates ve Ng, 2011: 7).

Hiper-parametre ayarları çoğunlukla modelin kara kutusu veya kara büyü olduğu tanımları ile anılmaktadır (Sen vd., 2018: 4538; Snoek vd., 2012: 2591; Wu vd., 2020: 788). Her ne kadar hiper-parametre optimizasyonu adı ile anılmakta olan bir optimizasyon olsa da normal optimizasyon problemlerinden farklıdır. Çünkü optimizasyon problemlerinde kullanıldığı gibi bir kayıp fonksiyonu veya başka matematiksel bir formülle kapalı formda gösterilemez. Yapılan işlem meta-optimizasyon işlemidir çünkü her hiper-parametre ayarının denemesi, modelin tekrar eğitimini gerektirmektedir (Zheng, 2015: 27).

Hiper-parametrelerin doğru şekilde ayarlanmasının önemi, modelin kapasitesini veya esnekliğinin bu sayede belirleniyor olması olarak özetlenebilmektedir. Modelin kapasitesini doğru ayarlamak meydana gelmesi muhtemel aşırı uyumu önlemek ve eğitim sürecinde modelin daha iyi bir öğrenme gerçekleştirerek daha doğru tahmin yapabilmesi olarak açıklanmaktadır. Böylece test verilerinde veya ilerleyen aşamalarda modelin kullanımı aşamasında daha iyi sonuçlar elde etmeyi sağlanmaktadır (Zheng, 2015: 28).

Hiper-parametre ayarlarının uygulanmasının zor tarafları ise: bir veri seti üzerinde yüksek performans ile çalışan bir hiper-parametre optimizasyonun başka bir set üzerinde aynı derecede iyi çalışmayabilmesi, kurulan modelde tüm hiper-parametreler için çalıştırılması gerekmesi ve bu sebeple çözüm alma süresini oldukça uzatan bir durum olması olarak gösterilmektedir (Mantovani vd., 2016: 38). Literatürde sıkça kullanılan üç yöntem bulunmaktadır. Bunlar; kılavuz arama yöntemi, rastgele arama yöntemi, bayes yöntemlerinden oluşmaktadır

### 3.1.3.1. Kılavuz Arama Yöntemi (Grid Search)

Kılavuz arama yöntemi en sık kullanılan yöntemlerden birisidir (Bergstra ve Bengio, 2012: 281). Bu durumun bir sebebi de geleneksel olarak kullanılan yöntemin bu olmasıdır (Ghawi ve Pfeffer, 2019: 161). Kılavuz arama yönteminde hiper-parametreler için belirlenen aralıklar dahilinde, ilgili tüm değerlerin mümkün olan her kombinasyonu için deneme yapılmaktadır. Genellikle çok fazla zaman aldığı için bazı durumlarda verinin tamamında deneme yapmak yerine daha küçük bir alan belirlenip bu alanda deneme yapılırken optimum noktanın nerede olduğu kestirilmeye çalışılır. Buna da manuel kılavuz arama yöntemi denir (Zheng, 2015: 31). Çok boyutlu bir yapıya sahip olması Bellman'ın (2015) ortaya koyduğu boyutsallık lanetinin (curse of dimensionality) ortaya da çıkmasına sebep olabilmektedir (Bellman, 2015: 94). Bilinen tüm yöntemler içinde en uzun hesaplama süresine sahip olduğu belirlenmiştir (Zheng, 2015: 31). Uygulaması kolay, elle yapılacak optimizasyonlara göre daha iyi ve hızlı ve az boyutlu uzaylarda kullanımı daha güvenilir bir yöntem olarak değerlendirilmektedir (Ghawi ve Pfeffer, 2019: 161). Domhan vd. (2015) hiper-parametre seçim gösterimini temel alınacak olursa:

A makine öğrenicisi  $\lambda_1, \dots, \lambda_n$  hiper parametreleri ile ve karşılık gelen nüfuz alanı ise  $\Lambda_1, \dots, \Lambda_n$  olmak üzere hiper-parametre uzayı  $\Lambda = \Lambda_1 * \dots * \Lambda_n$  olarak tanımlanır. Her parametre ayarı için  $\lambda \in \Lambda$ , ve  $A_\lambda$  ise  $A$  öğrenme algoritmasının bu ayarlamayı kullanıldığını göstermek için kullanılmaktadır. Devamında ise  $l(\lambda) = L(A_\lambda, D_{\text{eğitim}}, D_{\text{doğrulama}})$  doğrulama kaybının yani yanlış sınıflandırma oranının  $D_{\text{eğitim}}$  sırasında  $D_{\text{doğrulama}}$  verisi üzerinde  $A_\lambda$ 'nın başarısını göstermektedir. Bu optimizasyon  $\lambda \in \Lambda$  değerleri arasından  $l(\lambda)$  değerini en aza indirgeyen hiper-parametrelerin bulunması ile gerçekleştirilmektedir.

### 3.1.3.2. Rastgele Arama

Rastgele arama yönteminin kılavuz arama yöntemine göre en büyük farkı, uzayda araştırılacak tüm alanı taramak yerine rastgele örnekleri taramaktır (Zheng, 2015: 31). Bu işlem, arama yönteminin işini oldukça hafifletmekte ve tüm alanı araştırmaktan daha az maliyetli bir sonuç elde etmeyi sağlamaktadır. Aynı zamanda beklenenin aksine kılavuz arama yöntemine göre daha iyi sonuçlar verebildiği ve daha tercih edilebilir olduğu da literatürde yer almaktadır (Bergstra ve Bengio, 2012: 302). Bu durumun nedenleri incelendiğinde olasılığa dayalı bir açıklaması olduğu görülmektedir. Şöyle ki, sonlu maksimum örneğe sahip bir uzayda maksimum 60 rastgele gözlem, %95 olasılıkla gerçek maksimum değerinin ilk %5'lik alanı içinde kalmaktadır. Birbirinden bağımsız  $n$  tane örnek seçilirse her birinin istenilen aralığı kaçırma olasılığı  $(1 - 0.05)^n$ 'dir. Bu durumda en az 0.95 başarı olasılığına ulaşmak istenirse

ihtiyaç duyulan  $n$  aşağıdaki fonksiyona göre bulunabilmektedir (Zheng, 2015, s. 32).

$$1 - (1 - 0.05)^n > 0.95 \quad (25)$$

ve  $n \geq 60$  elde edilir.

Bu şekilde, %95 başarı ile çalışılmak istenen durumda 60 rastgele gözlem yapmanın da yeterli olabildiği ortaya konulmuştur. Bu başarı oranından daha düşük oranların da kabul edildiği durumlarda daha az gözlem sayısı ile de çalışmak mümkün olmaktadır.

### 3.1.3.3. Bayes Optimizasyonu

Kılavuz arama ve rastgele arama yöntemleri tüm olası noktaları önceden belirleyip sonrasında tamamını taramaktayken, akıllı ayarlama teknikleri birkaç hiper-parametre ayarını seçmekte ve seçilen parametrelerin kalitesini değerlendirerek hangi örnekleri alacağına karar vermektedir (Zheng, 2015: 32). Amaç modelin nihai tahmin puanını (örneğin doğruluk değeri) yükseltecek şekilde çalışmasına olanak sağlamaktır. Böylece daha az değerlendirme yapmak ve süre tasarrufuna gitmek mümkün olmaktadır.

Hiper-parametreler için Bayes optimizasyon yaklaşımı Snoek ve diğerleri tarafından 2012 yılında ortaya konmuştur. Akıllı hiper-parametre ayarlarından biri olan Bayes optimizasyonunda ana fikir, makine öğrenme algoritmasının performansının olasılığa bağlı bir temsilini oluşturmaya dayanmaktadır. Bayes optimizasyonunda rastgele aramada olduğu gibi hiper-parametre kombinasyonlarının alt kümesi örneklenmekte; ancak kombinasyon seçimi aşamasında bayes optimizasyonu ile rastgele arama yöntemi birbirlerinden farklılık göstermektedir (Snoek vd., 2012: 2). Seçim işlemi esnasında Bayes teoremi ile olasılık hesabı yapılır ve seçim işlemi bu işlemin sonucuna göre ilerlemektedir (26).

$$P(\text{değer} | \text{hiper parametreler}) \quad (26)$$

Olasılık modeli elde edildikten sonra devam eden kombinasyonlar, performansın artma olasılığının yüksek olduğu gözlemlerden seçilerek ilerlemektedir. Ancak performans olasılığının yüksek olmadığı tespit edilirse, hiper-parametre uzayının başka bir alt kümesi dikkate alınarak denemelere devam edilmektedir. Bu yapılan seçime sömürü ve keşif ödünleşimi adı verilmektedir (Snoek vd., 2012: 12). Bayes optimizasyonunun diğer yöntemlerden farkı, testler sırayla yapılırken daha önce yapılmış olan testleri göz önünde bulundurmasıdır. Böylece Bayes optimizasyonu, rastgele arama yöntemine göre daha avantajlı olarak değerlendirilmektedir (Tanyıldızı ve Demirtaş, 2019: 2).

### 3.1.4. Modellerin Performans Ölçüm Metrikleri

Makine öğrenmesi yöntemleri her geçen gün çeşitlenmekte ve artmaktadır. Bir veri seti için hangi algoritmanın daha iyi sonuç verdiği kararını vermek üzere tekniklerin performans

metriklerinin ölçümü gerekmektedir. Kullanılmakta olan çok sayıda performans metriği olmakla beraber, sınıflandırma problemleri için başvurulmakta olan çoğu performans metriği hata matrisi temeli ile geliştirilmiş metriklerden oluşmaktadır. Bu durum gözetilerek bu başlık altında öncelikle hata matrisi tanıtılmış ardından sınıflandırma problemlerinde sık kullanılan yöntemlere yer verilmiştir.

### 3.1.4.1. Hata Matrisi

Hata matrisi, diğer adıyla karşılıklık matrisi, sınıflandırma problemlerinin sonuçlarını değerlendirmede sıklıkla kullanılmaktadır. Bu matris tahmin ve gerçek değerlerden oluşmaktadır (Kohavi ve Provost, 1998: 271). Sınıflandırıcının performansının bir dökümü olarak da adlandırılabilir, çünkü doğru sınıflandırılan ve yanlış sınıflandırılan tüm değerler matriste incelenebilmektedir (Bramer, 2016: 89). Hata matrisinin büyüklüğü çıktı sayısına göre belirlenmektedir.  $X$  sayıda çıktısı bulunan bir model için  $X * X$  büyüklüğünde hata matrisi gerekmektedir (Zhang vd., 2020: 179). Tablo olası her sınıflandırma için bir satır ve sütun içerir, satırlar doğru sınıflandırmaya karşılık gelirken, sütunlar tahmini sınıflandırmaları temsil eder (Bramer, 2016: 89). Tablo 3.2’de gösterilmiş olan hata matrisinde pozitif ve negatif değerler sınıflandırma sonucundaki iki farklı sınıfı temsil etmektedirler. Tablo incelediğinde tahmin edilen değer pozitif ve tahmin de pozitif olduğunda gerçek pozitif (GP) olarak sınıflandırılır. Tahmin değeri pozitif gerçek değer negatif olduğunda yanlış pozitif (YP) olarak tabloda yer alır. Gerçek değer pozitif olduğu durumda negatif tahmin yapıldığında yanlış negatif (YN) sınıflandırma gerçekleşmiş ve tahmin ve gerçek değer negatif olduğu sınıflandırma gerçek negatif (GN) olarak tabloda yer alırlar. Hata matrisi temel alınarak birçok performans değerlendirme kriteri geliştirilmiştir (Davis ve Goadrich, 2006: 234).

İki tip sınıflandırma hatasından bahsetmek mümkündür. Bunlar Tip 1 ve Tip 2 olarak adlandırılmaktadır. Tip 1 hata aynı zamanda yanlış pozitifler olarak da adlandırılır ve negatif olarak sınıflandırılması gereken örneklerin pozitif sınıflandırıldığı durumları kapsamaktadır. Tip 1 hata yapmanın olasılığı  $\alpha$  ile gösterilir. Gerçek pozitifleri tahmin etme olasılığı  $1 - \alpha$  olarak hesaplanmaktadır. Tip 2 hata ise yanlış negatiflerdir. Bu durum da pozitif olarak sınıflandırılması gereken örneklerin negatif olarak sınıflandırılması durumunda ortaya çıkar (Bramer, 2016: 177). Tip 2 hata yapma olasılığı ise  $\beta$  ile gösterilir. Negatif değerlerin tahminin doğru yapıldığı durumların olasılığı ise  $1 - \beta$  ile hesaplanmaktadır.



**Tablo 3.2 Hata Matrisi**

Tahmin	Pozitif	GP	YP
	Negatif	YN	GN
		Pozitif	Negatif
		Gerçek	

**3.1.4.1.1. Doğruluk (Accuracy)**

Modelin ne derece doğru tahmin ürettiğine odaklanan bir metriktir ve genellikle modelin başarısını ölçmek için kullanılan yöntemlerin başında gelmektedir (Ferri vd., 2009: 29). Eşit dağılmayan verilerin değerlendirmesinde tek başına yetersiz olabilmektedir. Doğru sınıflandırılmış olan gerçek pozitif ve gerçek negatif değerlerinin toplam tahminlere oranı ile hesaplanmaktadır (26) (Kubat, 2017: 227).

$$\text{Doğruluk} = \frac{(GP + GN)}{(GP + YP + YN + GN)} \quad (26)$$

**3.1.4.1.2. Duyarlılık (Recall)**

Pozitif tahminlerin ne derece başarılı olduğunu ölçmek için kullanılmaktadır. Gerçek pozitif değerlerin, negatif olup pozitif olarak tahmin edilmiş değerlerin (YN) gerçek pozitif değerlerin toplamına oranı ile hesaplanır (27) (Kubat, 2017: 227). Pozitif olarak doğru şekilde sınıflandırılan toplam pozitif örneklerin oranıdır. Yanlış negatif olarak sınıflandırmanın maliyetinin yüksek olduğu durumlarda tercih edilir.

$$\text{Duyarlılık} = \frac{GP}{(GP + YN)} \quad (27)$$

**3.1.4.1.3. Kesinlik (Precision)**

Pozitif olarak tahmin edilen durumdaki başarıyı ölçmede kullanılmaktadır. Diğer bir deyişle pozitif olarak sınıflandırılanlar arasında doğru sınıflandırılmış örneklerin oranı değerlendirilmektedir. Genellikle yanlış pozitif değerlemenin maliyetinin yüksek olduğu durumlarda öncelikli incelenen bir performans metriğidir. Bunun için de gerçek pozitiflerin gerçek pozitif ve yanlış pozitif toplamının oranına bakılmaktadır (28) (Müller ve Guido, 2017: 283). Dengesiz veri setlerinde duyarlılık artırılmaya çalışılırken kesinliği etkilememek önemlidir, fakat bu iki hedef birbiri ile çelişkilidir. Azınlık sınıfı için gerçek pozitif değerler artırılmaya çalışılırken yanlış pozitif sayısı da artar, bu durum da kesinlik değerinin düşmesine neden olmaktadır (He ve Ma, 2013: 55).

$$Kesinlik = \frac{GP}{(GP + YP)} \quad (28)$$

#### 3.1.4.1.4. F Değeri

F değeri pozitif durumları sınıflandırma sırasında doğruluk ve kesinlik arasındaki ödünleşmeleri analiz etmektir. Bu amaçla kesinlik ve duyarlılık ölçütlerinin yani pozitif tahmin değeri ile gerçek pozitif oranının ağırlıklı harmonik ortalaması ile hesaplanmaktadır.

$$F_{\beta} = (1 + \beta^2) * \frac{kesinlik * duyarlılık}{(\beta^2 * kesinlik) + duyarlılık} \quad (29)$$

Genellikle F1 değeri yaygın olarak kullanılmaktadır. Avantajlarından biri tüm hata maliyetlerini göz önüne alarak hesaplamaya dahil etmesi olarak gösterilmektedir (Kubat, 2017: 227). Ayrıca hem duyarlılık hem de kesinlik metriklerini birlikte değerlendirmesi sebebi ile dengesiz veri setine sahip ikili sınıflandırma problemlerinde doğruluk ölçütü yerine kullanılması tavsiye edilmektedir (Müller ve Guido, 2017: 284).

$$F_1 = 2 * \frac{kesinlik * duyarlılık}{kesinlik + duyarlılık} \quad (30)$$

F2 değeri ise yanlış negatif olarak sınıflanan gözlemlerin maliyetinin daha yüksek olduğu durumlarda kullanılmaktadır (Brownlee, 2020: 1).

$$F_2 = 5 * \frac{kesinlik * duyarlılık}{(4 * kesinlik) + duyarlılık} \quad (31)$$

#### 3.1.4.1.5. Cohen Kappa İstatistiği

Bu istatistik iki sınıflandırıcı arasındaki uyumun bir ölçüsü olarak kullanılmaya başlanmıştır (Cohen, 1960: 37). Sonrasında sınıflandırıcı performansı ölçmek için de kullanılmaktadır (Witten ve Frank, 2002: 76). Sınıflandırma problemleri için kullanılmamasını tavsiye eden çalışma da bulunmaktadır (Delgado ve Tibau, 2019: 1).  $Pr(a)$  iki değerleyici arasında oluşan uyuşmaların toplam oranı ve  $Pr(e)$  değeri ise bu uyuşmanın şans eseri denk gelme olasılığı olacak şekilde hesaplanmaktadır (32).

$$\kappa = \frac{Pr(a) - Pr(e)}{1 - Pr(e)} \quad (32)$$

Hesaplanan  $\kappa$  değerine ilişkin değerlendirme Tablo 3.3'de bulunmaktadır.

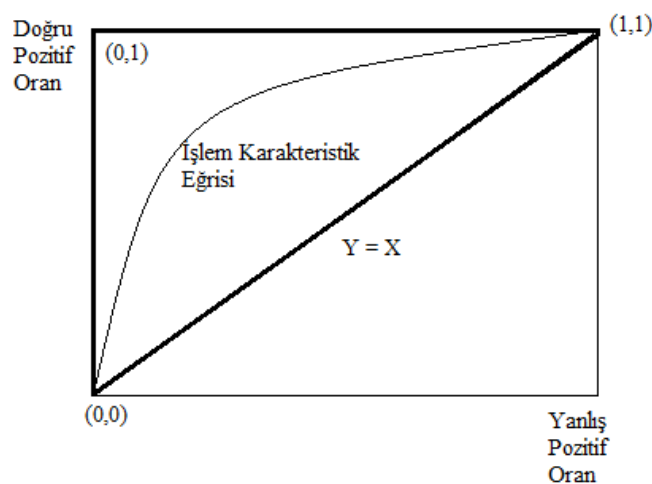
Tablo 3.3 Cohen Kappa Değeri Yorumlaması

$\kappa$	Yorum
< 0	Hiç uyuşma yok
0.0 — 0.20	Önemsiz uyuşma
0.21 — 0.40	Orta derecede uyuşma
0.41 — 0.60	Çoğunlukla uyuşma
0.61 — 0.80	Önemli derecede uyuşma
0.81 — 1.00	Neredeyse mükemmel uyuşma

Kaynak: (Landis ve Koch, 1977: 165)

### 3.1.4.1.6. İşlem Karakteristik Eğrisi ve Eğri Altında Kalan Alan

Tip 1 ve Tip 2 Hata değerlerinin diyagram üzerinde gösterimi işlem karakteristik eğrisi grafiğini oluşturmaktadır (Bramer, 2016: 182). Bu sayede testin ayırt etme gücü belirlenmektedir. Tek bir performans metriğinin yanlı sonuç vermesinin mümkün olduğu düşünülerek geliştirilmiş istatistiki bir değerlendirme yöntemidir (Metz, 1978: 283). Dikey ekseninde Tip 2 Hata, yatay ekseninde ise Tip 1 Hata yer almaktadır. Eğri üzerindeki her nokta farklı sonuçları temsil etmektedir. Şekil 3.4'de gösterildiği gibi eğrinin orijin noktasından hemen hemen dik bir şekilde (0,1) noktasına ulaşması oradan da yatay olarak (1,1) noktasına ulaşması mükemmel yakın olduğu sonucunu temsil etmektedir. Sınıflandırma problemlerini karşılaştırma ve değerlendirme için kullanılması önerilmektedir (Bradley, 1996: 1150). Fakat birden fazla modelin karşılaştırılmasında kullanıldığı durumlarda kullanışlı olmadığı belirlenmiştir (Provost vd., 1998: 445).



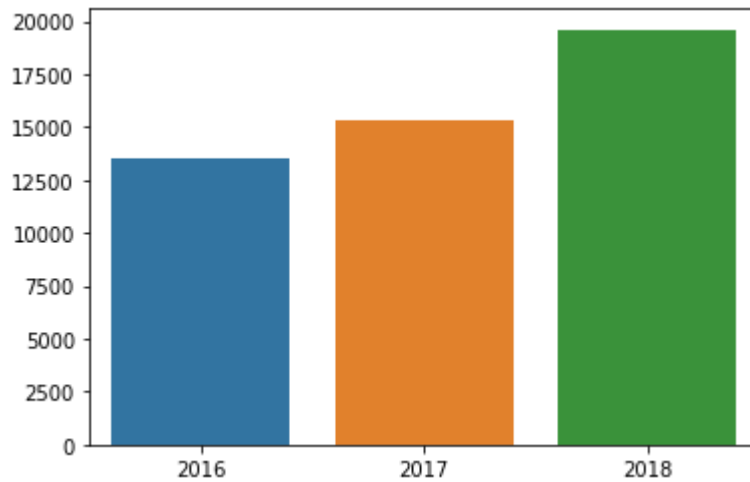
Şekil 3.4 Mükemmel Yakın İşlem Karakteristik Eğrisi

Bu zorluğu aşmak için eğrinin altında kalan alan hesaplanmaktadır. Algoritmanın tüm alan üzerine nasıl performans gösterdiğinin anlaşılması için kullanışlı bir yöntem olarak gösterilmektedir (33) (Branco vd., 2015: 7).

$$EAKA = \frac{\frac{GP}{GP+YN} + \frac{GN}{GN+YP}}{2} \quad (33)$$

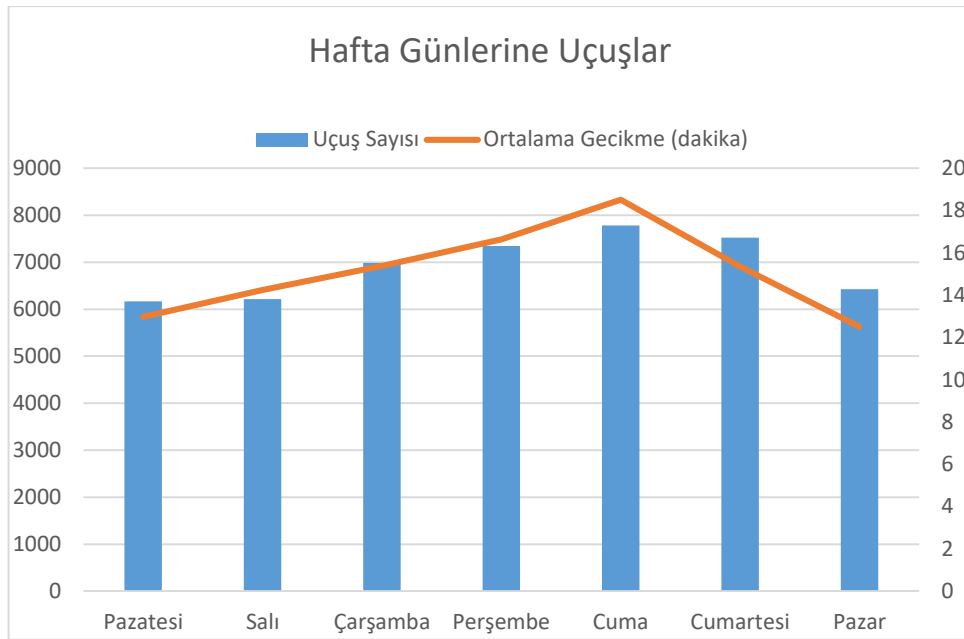
### 3.1.5. Verilerin İncelenmesi ve Analizi

Uçak gecikmelerinin makine öğrenmesi yöntemleri ile başarılı bir şekilde tahmininin gerçekleştirilmesi için gerekli olan büyük veri bir havayolu şirketinden temin edilmiştir. Veri setine havayolunun 2016-2018 yılları arasında gerçekleştirmiş olduğu tüm ulusal ve uluslararası uçuş verileri dahil edilmiştir. Süre zarfında 65 farklı ülkeye 48449 uçuş gerçekleştirilmiştir. İlgili uçuşların yıllara göre dağılımı Şekil 3.5'te gösterilmiştir. Şekilde havayolu şirketinin yıllar geçtikçe artan miktarda uçuş gerçekleştirmiş olduğu gözlemlenmektedir.



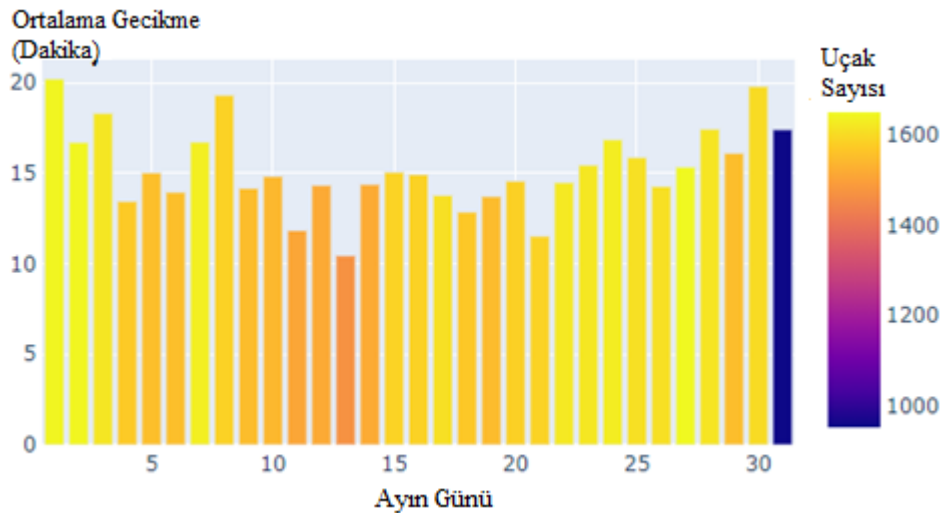
Şekil 3.5 Yıllara Göre Uçuş Sıklığı

Haftanın günlerine göre uçuş sıklığı ise Şekil 3.6 üzerinde gösterilmiş en sık uçuşun olduğu gün Cuma (7784) en az uçuşun olduğu gün ise Pazartesi (6167) olarak tespit edilmiştir. Uçuşların en yoğun olduğu Cuma günü, uçakların ortalama gecikme süresinin de en yüksek olduğu gün olarak ön plana çıkmaktadır.



**Şekil 3.6 Hafta Günlerine Göre Uçuş Sıklığı ve Ortalama Geç Kalma Süresi**

Şekil 3.7’de incelenmiş olan ayın günlerine göre yoğun günler ayların 1., 2., 7., ve 27. günlerine denk geldiği gözlemlenmektedir. Yılın 8 ayının 31 günden oluşması nedeni ile en az uçuş sayısı ayın 31’ine gerçekleşmiştir. Bu durum haricinde daha az uçuş gerçekleşen günler ayın 12’si ve 13’ü olmuştur. Ortalama geç kalma süresi ise ayın 1’i, 8’i ve 30’unda gerçekleşmektedir.



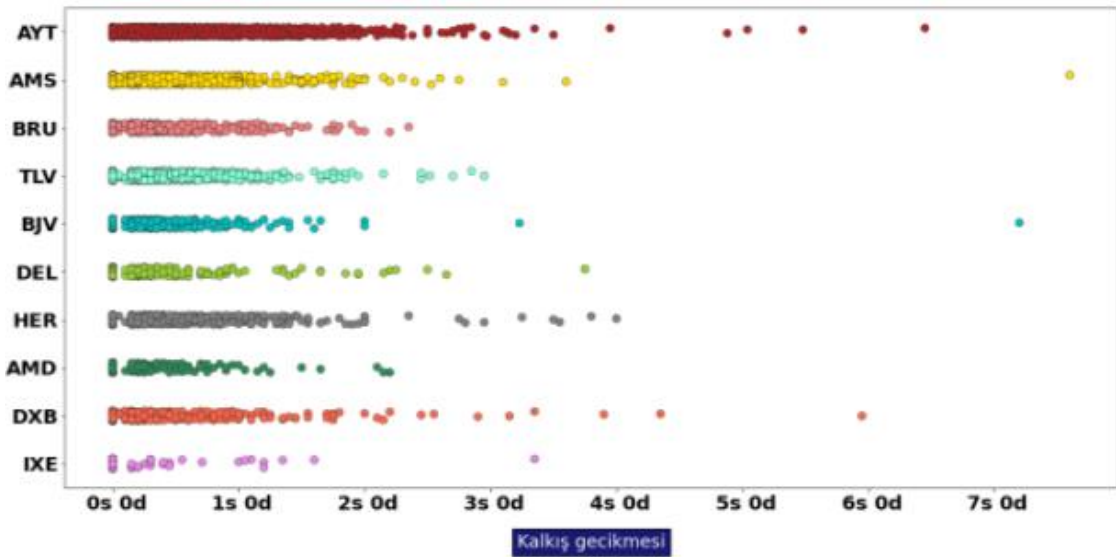
**Şekil 3.7 Ay Günlerine Göre Yoğunluk ve Ortalama Geç Kalma Süresi**

En çok gecikme yaşanan ilk 5 havaalanına ait bilgiler ise Tablo 3.4’te paylaşılmıştır. En uzun gecikme yaşanmış olan havalimanı 760 dakikalık gecikme ile Amsterdam Schiphol olarak tespit edilmiştir. Bu tabloda öne çıkan havaalanları en sık uçuş gerçekleştirilen havaalanları ile paralellik göstermektedir. Bu durum en sık kullanılan ilk 10 havalimanınının gösterilmiş olduğu Şekil 3.8 ile karşılaştırma yapıldığında görülmektedir.

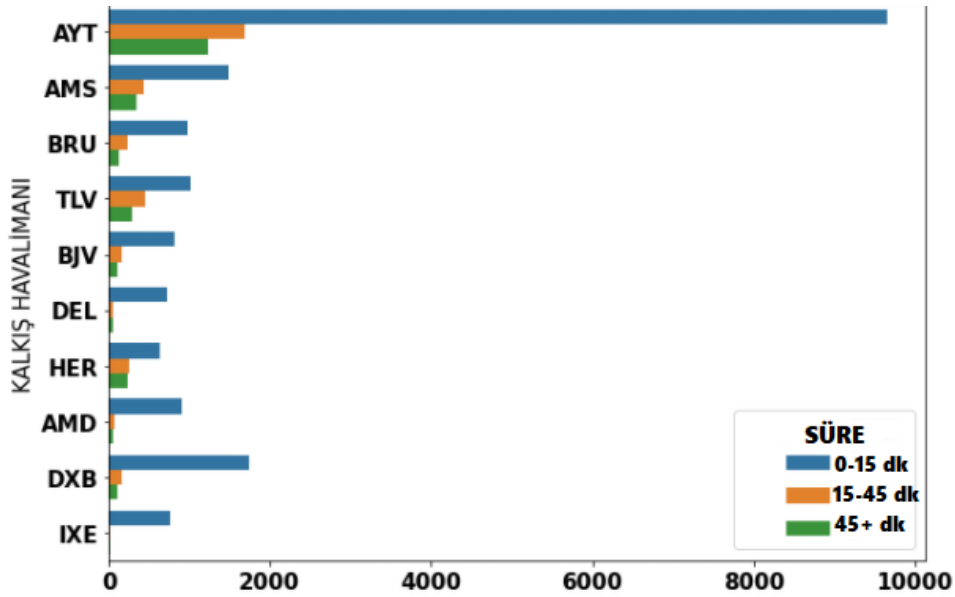
**Tablo 3. 4 En Sık Gecikme Gerçekleşen Havalimanları**

Havalimanı	En Kısa Süre (dakika)	En Uzun Süre (dakika)	Gecikme Sayısı	Ortalama Gecikme Süresi (dakika)
Antalya Havaalanı	0	645	12575	13,9
Amsterdam Schiphol Havalimanı	0	760	2259	20,5
Dubai Havalimanı	0	595	1987	9
Ben Gurion Uluslararası Havalimanı	0	295	1746	22,1
Brüksel Havalimanı	0	235	1323	14,1

2016-2018 yılları arasında havalimanı tarafından en sık kullanılan 10 havalimanının gecikme sürelerini gösteren grafik Şekil 3.8 üzerinde gösterilmektedir. Havalimanlarının açık isimleri sırasıyla Antalya, Amstredam Schiphol, Ben Gurion, Brüksel, Ben Gurion, Milas Bodrum, Indira Gandhi, Kandiye, Sardar Vallabhbai Patel, Dubai, Adani Airports Mangaluru Havalimanıdır. Bu havalimanları arasında en uzun geç kalma süresi yaklaşık sekiz saat gecikme ile Amsterdam Havalimanında gerçekleşmiştir.

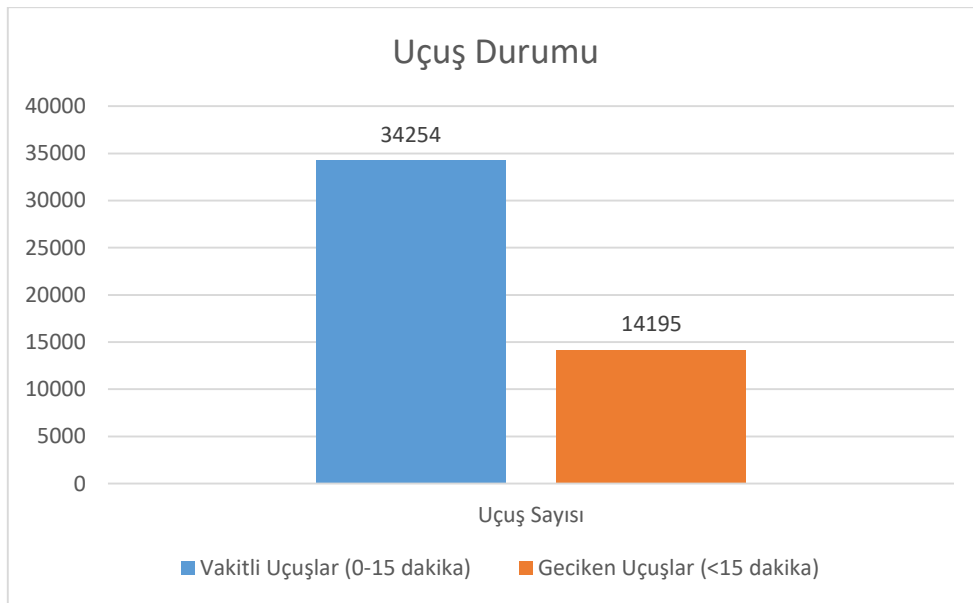
**Şekil 3.8 En Sık Kullanılan İlk On Havalimanındaki Gecikme Süreleri**

İlgili gecikmeler sürelerinde göre gruplandırıldığında ise, gecikmelerin birçoğunun ilk 15 dakika içinde üstesinden gelinmiş olduğu gözlemlenmektedir. Sonraki yarım saat içinde ise kalan uçuşların yarısından fazlasının gerçekleşmiş olduğu, daha uzun sürelerde ise çok sık gecikme yaşanmadığı gruplanma yapılmış olan Şekil 3.9 üzerinde gösterilmiştir.



Şekil 3.9 En Sık Kullan 10 Havaalanında Planlanan ve Gerçekleşen Kalkış Arasındaki Süre Farkı

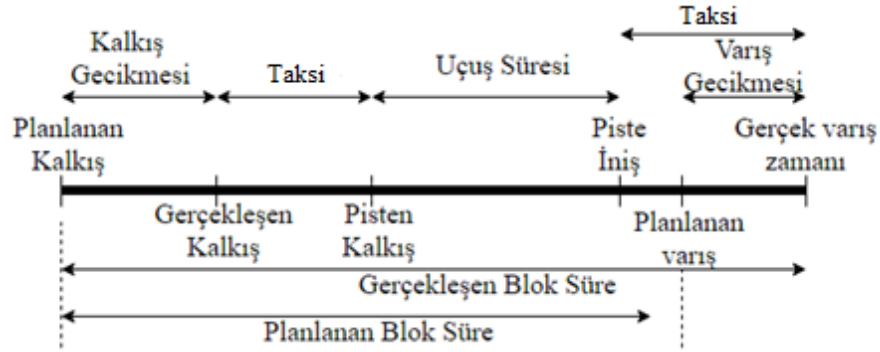
Şekil 3.10’da gösterildiği gibi 34254 uçuş belirlenen kalkış saatinden en çok 15 dakikalık gecikme ile kalkışını planlanan zaman dahilinde gerçekleştirmiş, 14195 uçuşta ise 15 dakikayı aşan gecikmeler yaşanmıştır. 15 dakika ve daha kısa süren gecikmeler Federal Aviation Administration’a (Federal Havacılık İdaresi) göre gecikme olarak kabul edilmediğinden (FAA, 2020: 1) çalışma dahilinde veri seti içinde 15 dakikayı aşan kalkışlar gecikme olarak kabul edilmiş, veri seti bu doğrultuda düzenlenmiştir.



Şekil 3.10 Vaktinde ve Geciken Uçuş Sıklıkları

Bir uçağın kalkış aşamasından varış aşamasına kadar gerçekleştirdiği tüm aşamalar ise Şekil 3.11 üzerinde gösterilmektedir. Bu şekilde en iyi durum havayolu tarafından önceden planlanmış olan saatte uçağın kalkıp, planlanan varış saatinde de varışı gerçekleştirmesidir. Uçak perondan ayrılıp hareket etmeye başladığında kalkış pistine doğru ilerler bu yola taksit

yolu adı verilmektedir. Her havaalanında farklı süreler alabilecek bu yolda geçirilecek süre de havayolları tarafından planlanmaktadır. Pistten ayrılıştan itibaren uçuş süresi başlar piste iniş ile birlikte varış havaalanındaki taksi yolunda ilerlenir ve perona yanaşan uçak varış yerine ulaşmış olur.



Şekil 3.11 Planlanan ve Gerçekleşen Uçuş Aşamaları

Literatürde sıkça kullanılan değişkenler modele dahil edilmiştir. Bu değişkenler havayolu şirketinden alınmış ve Tablo 3.5'te gösterilmiştir.

Tablo 3.5 Uçuşa Ait Değişkenler

Değişken İsmi	Açıklama	Veri Tipi	Kategori Sayısı
plan_gun	Planlanan ayın günü	Kategorik	7
plan_ay	Planlanan yılın ayı	Kategorik	12
plan_zaman	Planlanan kalkış zamanı	Sürekli	
plan_hgun	Planlanan haftanın günü	Kategorik	7
O_airport	Kalkış havalimanı	Kategorik	303
D_airport	Varış havalimanı	Kategorik	301
taxi_out	Peron ile kalkış arasında geçen süre	Sürekli	
pist_ayrilis	Uçağın yerden ayrıldığı zaman	Sürekli	
plan_varis	Uçağın varış pistine indiği an	Sürekli	
plan_ucus_suresi	Havada geçen süre	Sürekli	

Havayolu firmasından alınan verilere ek olarak literatürde de sıklıkla başvurulduğu gibi uçuşlar üzerinde etki sahibi olması bakımından hava durumu bilgileri de uçuş verileri ile birlikte kullanılmıştır. Kullanılan veriler çevrimiçi web sitesi olan rp5.ru aracılığı ile temin edilmiştir. Hava durumu ile ilgili değişkenler Tablo 3.6'da gösterilmiştir. İlgili değişkenler kalkış havalimanında kalkış saatindeki hava durumu ile kalkış saatinde varış havalimanının hava durumu olarak modele iki ek set olarak eklenmiştir.



**Tablo 3.6 Hava Durumu Değişkenleri**

Değişken İsmi	Açıklama	Veri Tipi	Kategori Sayısı	Eksik Veri Sayısı
T2	Kalkış Havalimanı Hava Sıcaklığı	Sürekli		25
Po2	Kalkış Havalimanı Ölçüm Yeri Hava Basıncı	Sürekli		30
P2	Kalkış Havalimanı Deniz Seviyesinde Atmosferik Basınç	Sürekli		13
U2	Kalkış Havalimanı Bağıl Nem	Sürekli		48
DD2	Kalkış Havalimanı Rüzgar Yönü	Kategorik	18	14
Ff2	Kalkış Havalimanı Rüzgar Hızı	Sürekli		14
C2	Kalkış Havalimanı Bulutluluk Durumu	Kategorik	5044	26
T3	Variş Havalimanı Hava Sıcaklığı	Sürekli		141
Po3	Variş Havalimanı Ölçüm Yeri Hava Basıncı	Sürekli		146
P3	Variş Havalimanı Deniz Seviyesinde Atmosferik Basınç	Sürekli		133
U3	Variş Havalimanı Bağıl Nem	Sürekli		149
DD3	Variş Havalimanı Rüzgar Yönü	Kategorik	19	152
Ff3	Variş Havalimanı Rüzgar Hızı	Sürekli		162
C3	Variş Havalimanı Bulutluluk Durumu	Kategorik	4563	3419

Veri setinde uçuş durumu verilerinde eksik veri bulunmamaktayken, hava durumu verilerinde bazı eksik veriler tespit edilmiştir. Sürekli veriler için tüm verilerin ortalaması alınarak eksik veriler tamamlanmıştır. Kategorik verilerdeki eksiklikler ise “Bilinmiyor” olarak tamamlanmıştır. Kategorik veriler kategorik değerleri dönüştürme (one-hot-encoding) yöntemi ile tam sayılı ifadeler haline dönüştürülmüştür. Dönüşüm sonucunda 10316 değişken elde

edilmiştir. Sadece YSA yönteminde veriler 0 ile 1 arasında ölçeklendirilerek (MinMaxScaler) normalize edilmiştir. Veri seti dengesiz gözlemlerden oluşmakta olduğu için seçilen makine öğrenmesi yöntemleri ayrıca sentetik veri (SMOTE) üretimi yöntemine başvurulmuştur. SMOTE ile elde edilen veri seti ile tüm makine öğrenmesi yöntemleri için ayrıca denemeler gerçekleştirilmiştir. Üçüncü deneme ise hiper parametre ayarlarına başvurularak gerçekleştirilmiştir. Tüm karar ağacı temelli yöntemler için Bayes yöntemi ile hiper-parametre araması yapılmış sonuçları ayrıca değerlendirilmiştir. Tüm makine öğrenme denemeleri Jupyter Notebook 6.1.4 versiyonu ile Python 3.8.5 versiyonu ile gerçekleştirilmiştir.

### 3.2. Araştırma Bulguları

Çalışmanın birinci bölümünde detaylı bir şekilde yer verilmiş olan makine öğrenme teknikleri, günümüzde insan eliyle analiz edilmesi güç problemlerin çözümünde daha gerçekçi sonuçların ortaya çıkmasını sağlamaktadır. Bu bakış açısıyla bu bölümde uçakların geç kalması ile ilgili olarak toplanmış olan verilerin analizi aşamasında makine öğrenme tekniklerinden YSA, RO, XGBoost, LightGBM ve CatBoost kullanılmış ve analiz sonuçları ortaya konmuştur. Bu başlık altında ilgili tekniklerin bulguları incelenirken öncelikle varsayılan ayarlar ile kurulmuş model sonuçlarına, sonrasında sentetik veri ile güncellenmiş veri seti ile model sonuçlarına ve son olarak karar ağacı temelli yöntemler için hiper-parametre ayarları yapılmış model sonuçlarına yer verilmiştir. Yapay sinir ağlarının sistem gereksinimleri yüksek olduğu ve bu donanımsal alt yapıya sahip olunamadığı için bu modele özgü olarak Bayes ile hiper-parametre ayarları gerçekleştirilememiştir. YSA ile Bayes hiper-parametre ayarları yapılamadığından yerine model ile el ile deneme yapılarak en iyi değerleri verdiği tespit edilen çift katmanlı YSA modeline yer verilmiştir.

#### 3.2.1. Yapay Sinir Ağları Kullanılarak Elde Edilen Bulgular

Yapay sinir ağları yönteminde öncelikle tanımlı kütüphane içinde varsayılan ayarlar kullanılarak uygulama yapılmıştır. Burada parametreler gizli katmanda 100 düğüm, aktivasyon fonksiyonu olarak doğrultulmuş lineer birim (relu) fonksiyonu, öğrenme oranı 0,001, maksimum iterasyon sayısı 2000 olarak kullanılmıştır. Hazırlanan hata matrisinde tahmin değerleri satırlarda, gerçek değerler sütunlarda yer almaktadır. Gerçek pozitif değerler geç kalacak uçuşları temsil ederken, gerçek negatifler ise zamanında kalkan uçuşları temsil etmektedir. Ortaya koyulan problemde geciken uçakların geç kalmayacak olarak gruplanması daha maliyetli olduğundan yanlış negatif değerlerin daha maliyetli olduğu düşünülmektedir. Tablo 3.7'de YSA modelinin Python üzerindeki varsayılan ayarları ile elde edilmiş olan eğitim sonucundaki hata matrisine yer verilmiştir. Tabloya göre eğitim aşamasında model sonuçlarla

oldukça yüksek uyum göstermiştir. Tablo 3.8’de ise YSA modelinin varsayılan ayarlarla test sonuçları yer almaktadır. Bu sonuçlar için 2855 (1238+1617) uçağın geç kalacağı 11680 (9293+2387) uçuşun ise zamanında gerçekleşeceği tahmin edilmiştir. Gerçek sonuçlara göre ise 3625 uçak gecikmiş, 10910 uçak ise planlanan zamanda kalkış gerçekleştirmiştir. 1238 uçuş gecikecek olarak tahmin edilmiş ve gerçekte de gecikmişlerdir (GP). 2387 uçuş gecikecek olarak tahmin edilmiş fakat zamanında kalkış gerçekleştirmiştir (YN). 1617 uçuş gecikmeyecek olarak tahmin edilmiş fakat uçuş kalkışı vaktinde olmuştur (YP). Son olarak 9293 uçuş gecikmeyecek olarak tahmin edilmiş ve gerçekte de gecikmemiştir. Eğitim sonuçları ile test sonuçları arasında yüksek farklılık gözlemlenmiştir. Model eğitim verisine göre fazla uyum göstermiş; fakat test verisi üzerinde çok başarılı olamamıştır.

**Tablo 3.7 YSA Varsayılan Parametrelere Göre Hata Matrisi Eğitim Sonuçları**

Tahmin	Pozitif	8318	1
	Negatif	4	25591
		Pozitif	Negatif
		Gerçek	

**Tablo 3.8 YSA Varsayılan Parametrelere Göre Hata Matrisi Test Sonuçları**

Tahmin	Pozitif	1238	1617
	Negatif	2387	9293
		Pozitif	Negatif
		Gerçek	

SMOTE yöntemiyle sentetik veri üretimi hata matrisi eğitim sonuçları Tablo 3.9’da paylaşılmıştır. Varsayılan ayarlara benzer bir şekilde eğitim aşamasında model oldukça yüksek başarı göstermiştir. Sadece 5 uçuş yanlış sınıflandırılmıştır. Tablo 3.10’da ise test sonuçlarına yer verilmiştir. Sentetik veri kullanımı ile varsayılan ayarlar test sonuçları karşılaştırıldığında yanlış pozitif sınıflandırılmış gözlem sayısında artış gözlemlenirken, yanlış negatif sınıflandırma sayısında azalma gerçekleştiği görülmüştür.

**Tablo 3.9 YSA SMOTE ile Hata Matrisi Eğitim Sonuçları**

Tahmin	Pozitif	25590	3
	Negatif	2	25589
		Pozitif	Negatif
		Gerçek	

**Tablo 3.10 YSA SMOTE ile Hata Matrisi Test Sonuçları**

Tahmin	Pozitif	1362	1638
	Negatif	2263	9272
		Pozitif	Negatif
		Gerçek	

YSA tekniği ayrıca çift gizli katman ile de çalıştırılmıştır. Çift gizli katman kullanılırken varsayılan parametrelerden farklı olarak iki katman kullanılmış buradaki gizli düğüm sayısı ilk

gizli katman için 30 ve ikinci katman için 30 olarak seçilmiş ve maksimum deneme 2000 olarak belirlenmiştir. Belirlenen nöron sayısı denemeler sonucunda elde edilen en iyi değere göre seçilmiştir. Tablo 3.11 üzerinde varsayılan ayarlarla elde edilen eğitim aşaması hata matrisi sonuçları paylaşılmıştır. Eğitim aşaması önceki sonuçlara göre daha başarısız olsa da Tablo 3.12’de paylaşılan test sonuçları incelendiğinde yanlış negatif sınıflandırma sayısının en düşük elde edildiği test sonucunun da burada elde edildiği görülmektedir.

**Tablo 3.11 Çift Gizli Katman YSA Varsayılan Parametrelere Göre Eğitim Sonuçları**

Tahmin	Pozitif	8289	90
	Negatif	33	25502
		Pozitif	Negatif
Gerçek			

**Tablo 3.12 Çift Gizli Katman YSA Varsayılan Parametrelere Göre Test Sonuçları**

Tahmin	Pozitif	2028	1763
	Negatif	1597	9147
		Pozitif	Negatif
Gerçek			

SMOTE ile çift katman uygulandığında ise Tablo 3.13 ve Tablo 3.14’de gösterilmiş olan eğitim ve test hata matrisi sonuçları elde edilmiştir. Eğitim aşamasında benzer sonuçlar elde edilirken, test aşamasında yanlış pozitif sınıflandırılan uçuşlarda varsayılan ayarlara göre azalma yanlış negatif sınıflandırılan uçuşlarda ise varsayılan ayarlara göre artış olduğu gözlemlenmiştir.

**Tablo 3.13 SMOTE ile Çift Gizli Katman YSA Varsayılan Parametrelere Göre Eğitim Hata Matrisi Sonuçları**

Tahmin	Pozitif	25386	29
	Negatif	206	25563
		Pozitif	Negatif
Gerçek			

**Tablo 3.14 SMOTE ile Çift Gizli Katman YSA Varsayılan Parametrelere Göre Test Hata Matrisi Sonuçları**

Tahmin	Pozitif	1577	1338
	Negatif	2048	9572
		Pozitif	Negatif
Gerçek			

### 3.2.2. Rastgele Ormanlar

Rastgele ormanlar tekniği uygulanırken varsayılan ayarlarda ormandaki ağaç sayısı 100, dallanmayı sağlamak için gerekli endeks gini endeksi, düğümleri bölmek için gereken minimum örnek sayısı 2, bir yaprak düğümünde olması gereken minimum örnek sayısı 1 olarak kullanılmıştır. Hata matrisi sonuçları Tablo 3.15’te eğitim Tablo 3.16’da ise test sonuçları paylaşılmıştır. Tablo 3.15’e göre model sonuçlara göre tamamen uyumlu bir öğrenme

gerçekleştirmiştir. Tablo 3.16’da paylaşılan test sonuçlarına göre ise model ağırlıklı olarak uçuşları vaktinde kalkacak olarak tahmin etmektedir. Toplamda tüm uçuşlardan sadece 225 uçağın geç kalacağı tahmini yapılmıştır. Veri setindeki dengesiz dağılım sonucu uçuşların geç kalmayacağı yönünde bir tahmin yaparak yanlış öğrenme gerçekleştirmiştir.

**Tablo 3.15 Rastgele Ormanlar Varsayılan Ayarlar ile Eğitim Aşaması Hata Matrisi Sonuçları**

Tahmin	Pozitif	25592	0
	Negatif	0	8322
		Pozitif	Negatif
Gerçek			

**Tablo 3.16 Rastgele Ormanlar Varsayılan Ayarlar ile Test Aşaması Hata Matrisi Sonuçları**

Tahmin	Pozitif	178	47
	Negatif	3447	10863
		Pozitif	Negatif
Gerçek			

Sentetik veri üretimi ile varsayılan değerler kullanılarak uygulanan modelin hata matrisi sonuçları Tablo 3.17 üzerinde eğitim ve Tablo 3.18 üzerinde test olarak yer almaktadır. SMOTE yöntemi ile de eğitim aşaması sonuçlarında %100 şekilde gerçekleşmiştir. Fakat test aşaması incelendiğinde sonuçların hala yanlış öğrenmeye yatkın olduğu sentetik veri üretiminin model sonuçlarında çok fark yaratmadığı görülmektedir.

**Tablo 3.17 SMOTE ile Rastgele Ormanlar Yöntemi Eğitim Aşaması Hata Matrisi Sonuçları**

Tahmin	Pozitif	25592	0
	Negatif	0	25592
		Pozitif	Negatif
Gerçek			

**Tablo 3.18 SMOTE ile Rastgele Ormanlar Yöntemi Test Aşaması Hata Matrisi Sonuçları**

Tahmin	Pozitif	227	76
	Negatif	3398	10834
		Pozitif	Negatif
Gerçek			

Karar ağacı temelli tekniklerde hiper-parametre ayarlarına da yer verilmiştir. Hiper-parametreler için kullanılan parametre uzayı ile optimum değerler Tablo 3.19 üzerinde gösterilmiştir.

**Tablo 3.19 Rastgele Ormanlar Parametre Uzayı ve Optimum Değerler**

Parametreler	Parametre Aralığı	Optimum Değerler
Ormandaki Ağaç Sayısı	(100, 1000)	502
Maksimum Derinlik	(5, 50)	49
Minimum Yaprak	(1, 20)	2
Yaprak Ayrımı	(2, 100)	2

Çapraz doğrulama için 5 katman oluşturulmuştur. Yöntemin tahminlerine göre oluşturulmuş eğitim ve test hata matrisleri sırasıyla Tablo 3.20 ve 3.21’de sunulmuştur. Eğitim sonuçlarına bakıldığında öğrenmenin tam model sonuçlarına gerçekleşmediği görülmektedir. Test sonuçlarında ise hiper-parametre ayarı ile model daha hassas ayarla EAKA kazancını yükseltecek şekilde problem çözümüne odaklanmış olsa da model anlamlılığı kalmamıştır, gerçekleşen hemen hemen tüm uçuşların zamanında olacağı tahmini yapılmıştır. Bu durum bazı performans metriklerinin yüksek çıkmasını sağlasa da modeli kullanılabilir olmaktan uzaklaştırmaktadır.

**Tablo 3.20 Hiper-Parametre Ayarları ile Rastgele Orman Tekniği Eğitim Hata Matrisi Sonuçları**

Tahmin	Pozitif	1147	0
	Negatif	7175	25592
		Pozitif	Negatif
Gerçek			

**Tablo 3.21 Hiper-Parametre Ayarları ile Rastgele Orman Tekniği Test Hata Matrisi Sonuçları**

Tahmin	Pozitif	24	2
	Negatif	3601	10908
		Pozitif	Negatif
Gerçek			

### 3.2.3. XGBoost

XGBoost yöntemi varsayılan ayarlarında ise öğrenme oranı 3, maksimum derinlik 6, eğitim için alt örneklem oranı 1, minimum örnek ağırlığı 1 (yaprak belli bir oranın altında düğümlerle sonuçlanırsa, daha fazla bölümlenmeden sonlandırılacaktır) parametreleri bulunmaktadır. Bu parametreler ile hata matrisi sonuçları eğitim aşaması için Tablo 3.22 test aşaması için sonuçları Tablo 3.23 üzerinde yer almaktadır. Eğitim aşaması ve test aşaması

benzer sonuçlar verdiği için öğrenmenin başarılı bir şekilde gerçekleştiği görülmektedir. Diğer yöntemlere göre daha iyi tahmin değerleri XGBoost ile elde edilmiştir.

**Tablo 3.22 XGBoost Varsayılan Ayarlarla Eğitim Aşaması Hata Matrisi Sonuçları**

Tahmin	Pozitif	8213	138
	Negatif	109	25454
		Pozitif	Negatif
Gerçek			

**Tablo 3.23 XGBoost Varsayılan Ayarlarla Test Aşaması Hata Matrisi Sonuçları**

Tahmin	Pozitif	3451	130
	Negatif	174	10780
		Pozitif	Negatif
Gerçek			

SMOTE yöntemi ile tekrar uygulanan XGBoost hata matrisi eğitim sonuçları Tablo 3.24'te test sonuçları ise Tablo 3.25'te sunulmuştur. Varsayılan ayarlar ile elde edilen hata matrisi sonuçlarına göre daha kötü sonuçlar elde edilmiştir.

**Tablo 3.24 XGBoost ve SMOTE Kullanımıyla Eğitim Aşaması Hata Matrisi Sonuçları**

Tahmin	Pozitif	25419	156
	Negatif	173	25436
		Pozitif	Negatif
Gerçek			

**Tablo 3.25 XGBoost ve SMOTE Kullanımıyla Test Aşaması Hata Matrisi Sonuçları**

Tahmin	Pozitif	3408	140
	Negatif	217	10770
		Pozitif	Negatif
Gerçek			

Bayes yöntemi ile hiper-parametre ayarı yapılmak üzere belirlenen parametre uzayı ve elde edilen optimum değerler Tablo 3.26 aracılığı ile gösterilmiştir.

**Tablo 3.26 XGBoost Parametre Uzayı ve Optimum Değerler**

Parametreler	Parametre Aralığı	Optimum Değerler
Öğrenme Oranı (eta)	(0,01, 1)	0,58
Orman Büyüklüğü	(100, 1000)	412
Maksimum Derinlik	(3, 10)	10
Alt Küme Örneklem Oranı	(0, 1)	0,88
Minimum Örnek Ağırlığı	(0, 20)	0,2
Gamma	(0, 5)	4

Çapraz doğrulama için 5 katman oluşturulmuştur. Modelin test ve eğitim hata matris sonuçları sırasıyla Tablo 3.27 ve Tablo 3.28 üzerinde gösterilmiştir. Hem eğitim hem de test sonuçları diğer XGBoost denemelerine benzer sonuçlar vermiştir. Model sonuçları başarılı

olmakla beraber Bayes ile XGBoost yönteminin en düşük yanlış negatif değeri (161) elde edilmiştir.

**Tablo 3.27 Hiper-Parametre Ayarları ile XGBoost Eğitim Hata Matrisi**

Tahmin	Pozitif	8219	145
	Negatif	103	25447
		Pozitif	Negatif
Gerçek			

**Tablo 3.28 Hiper-Parametre Ayarları ile XGBoost Test Hata Matrisi**

Tahmin	Pozitif	3464	138
	Negatif	161	10772
		Pozitif	Negatif
Gerçek			

### 3.2.4. LightGBM

LightGBM yönteminde ise varsayılan parametrelerde yaprak sayısı 31, maksimum derinlik sınırsız, yapraktaki minimum veri sayısı 20, torbalama oranı ya da diğer adıyla alt küme oluşturma oranı 1, her iterasyonda rastgele bir alt değişken kümesinin seçimi 1 (örneğin 0,8 olarak ayarlandığı durumda ağaç eğitilmeden önce değişkenlerin %80'inini seçeceği anlamına gelmektedir) olarak belirlenmiştir. Varsayılan ayarlara göre elde edilen eğitim ve test hata matrisi sırasıyla Tablo 3.29 ve Tablo 3.30'da yer almaktadır. Eğitim ve test hata matrislerinin benzer olması aşırı öğrenme olmadığını göstermektedir. XGBoost sonuçları ile karşılaştırıldığında ise yanlış negatif tahminlerde artış gözlemlenirken yanlış pozitif tahminlerde düşüş gerçekleştiği görülmüştür. Bu yönüyle yanlış negatif gözlem sayıları problem için daha maliyetli olarak değerlendirildiğinden varsayılan ayarlar için XGBoost yöntemine göre daha kötü test sonuçları verdiği ortaya çıkmaktadır.

**Tablo 3.29 LightGBM Varsayılan Ayarlar ile Eğitim Hata Matrisi Sonuçları**

Tahmin	Pozitif	7513	214
	Negatif	809	25378
		Pozitif	Negatif
Gerçek			

**Tablo 3.30 LightGBM Varsayılan Ayarlar ile Test Hata Matrisi Sonuçları**

Tahmin	Pozitif	3124	118
	Negatif	501	10792
		Pozitif	Negatif
Gerçek			

SMOTE ve LightGBM tekniklerinin bir arada kullanıldığı eğitim ve test hata matrisleri Tablo 3.31 ve 3.32 aracılığı ile gösterilmiştir. Varsayılan ayarlar ile yapılan LightGBM



sonuçlarında göre yanlış negatif ve yanlış pozitif değerlerde artış olduğu görülmüştür. Bu sebeple varsayılan ayarlar ile elde edilen çözümün daha başarılı olmuştur.

**Tablo 3.31 LightGBM ve SMOTE Kullanımıyla Eğitim Aşaması Hata Matrisi Sonuçları**

Tahmin	Pozitif	25348	244
	Negatif	1066	24526
		Pozitif	Negatif
Gerçek			

**Tablo 3.32 LightGBM ve SMOTE Kullanımıyla Test Aşaması Hata Matrisi Sonuçları**

Tahmin	Pozitif	3058	139
	Negatif	567	10771
		Pozitif	Negatif
Gerçek			

Bayes yöntemi ile hiper-parametre ayarı yapılmak üzere belirlenen parametre uzayı ve elde edilen optimum değerler Tablo 3.33 üzerinde gösterilmiştir.

**Tablo 3.33 LightGBM Parametre Uzayı ve Optimum Değerler**

Parametreler	Parametre Aralığı	Optimum Değerler
Yaprak Sayısı	(25, 45)	44
Maksimum Derinlik	(5, 35)	19
Yapraktaki Minimum Veri Sayısı	(0,1, 0,9)	71
L1 Düzenleme Terimi	(0, 0,05)	0,04
L2 Düzenleme Terimi	(0, 0,05)	0,04
Rastgele Alt Özellik Seçimi	(0,8, 1)	0,79

Çapraz doğrulama için 5 katman oluşturulmuştur. Bu ayarlara göre uygulanan hiper-parametre optimizasyonu ile elde edilen optimum değerler ile oluşturulan modelin eğitim ve test hata matrisi sırasıyla Tablo 3.34 ve Tablo 3.35 aracılığı ile sunulmuştur. Eğitim aşamasında tüm model sonuçlarına uyumlu bir model gerçekleşmiştir. Test sonuçlarında ise LightGBM yöntemleri içinde en düşük yanlış negatif değer hiper-parametre ayarları yapıldıktan sonra elde edilmiştir. Yanlış pozitif tahmin sayısı diğer LightGBM yöntemlerine göre artmış olsa da yanlış negatif tahminin eldeki problem için daha maliyetli olduğundan hiper-parametre ayarları ile elde edilen sonuçlar daha kullanışlı olarak değerlendirilmektedir.

**Tablo 3.34 Hiper-Parametre Ayarları ile LightGBM Eğitim Hata Matrisi**

Tahmin	Pozitif	8322	0
	Negatif	0	25592
		Pozitif	Negatif
Gerçek			

**Tablo 3.35 Hiper-Parametre Ayarları ile LightGBM Test Hata Matrisi**

Tahmin	Pozitif	3470	165
	Negatif	155	10745
		Pozitif	Negatif
Gerçek			

### 3.2.5. CatBoost

Son olarak CatBoost yöntemi incelenmiştir. İlk aşamada varsayılan ayarlar ile uygulanmıştır. Bu ayarlara göre maksimum kurulabilecek ağaç sayısı (orman genişliği) 1000, maksimum derinlik 6, öğrenme oranı 0,03, torbalama örneklem oluşturma 1, 12 düzenleme terimi 3, ikili sınıflandırma problemlerinde sınıf 1 için nesnelerin ağırlıklarında çarpan olarak kullanılma değeri 1 olarak belirlenmiştir. Tablo 3.36 ve Tablo 3.37’de CatBoost tekniğinin belirtilen varsayılan ayarlara göre sırasıyla eğitim ve test hata matrislerine yer verilmiştir. Test aşamasında da yanlış pozitif test tahmin sayısı gradyan yükseltme yöntemleri içindeki en düşük yanlış pozitif tahmin (110) olarak elde edilmiştir. Bu CatBoost tekniği için bir avantaj olarak değerlendirilmekle beraber geç kalacak uçuşların doğru tahmini problem için daha önemli olduğundan yanlış pozitiflerin diğer gradyan yükseltme temelli yöntemlere göre yüksek elde edilmesi modelin tercih edilebilirliğini düşürmektedir.

**Tablo 3.36 CatBoost Varsayılan Ayarlar ile Eğitim Hata Matrisi Sonuçları**

Tahmin	Pozitif	6392	184
	Negatif	1930	25408
		Pozitif	Negatif
Gerçek			

**Tablo 3.37 CatBoost Varsayılan Ayarlar ile Test Hata Matrisi Sonuçları**

Tahmin	Pozitif	2665	110
	Negatif	960	10800
		Pozitif	Negatif
Gerçek			

CatBoost ile SMOTE yönteminin birlikte kullanıldığı eğitim ve test aşaması hata matrisleri Tablo 3.38 ve Tablo 3.39’da yer almaktadır. Sentetik veri üretimi CatBoost tekniği ile birlikte kullanıldığında yanlış pozitif sayısı hem eğitim hem de test aşamasında bir miktar arttırmıştır. Aynı zamanda yanlış negatif tahminlerin sayısının da azalmasını sağlamıştır. Bu da modeli varsayılan ayarlar ile eğitilmiş CatBoost modeline göre tercih edilebilir hale getirmiştir.

**Tablo 3.38 CatBoost ve SMOTE Kullanımıyla Eğitim Aşaması Hata Matrisi Sonuçları**

Tahmin	Pozitif	23922	200
	Negatif	1670	25922
		Pozitif	Negatif
Gerçek			

**Tablo 3.39 CatBoost ve SMOTE Kullanımıyla Test Aşaması Hata Matrisi Sonuçları**

Tahmin	Pozitif	2712	118
	Negatif	913	10792
		Pozitif	Negatif
Gerçek			

CatBoost tekniğinin parametre uzayı ve bu uzay içinden elde edilen optimum değerler Tablo 3.40'ta yer almaktadır.

**Tablo3.40 CatBoost Parametre Uzayı ve Optimum Değerler**

Parametreler	Parametre Aralığı	Optimum Değerler
Maksimum Derinlik	(5, 15)	15
Torbalama Örneklem Oluşturma	(3, 10)	5
L2 Düzenleme Terimi	(2, 10)	4

Elde edilen optimum değerler CatBoost tekniği ile birlikte kullanılırken 5 katmalı çapraz doğrulama aracılığı ile uygulanmıştır. Hiper-parametre ayarı sonucunda elde edilen modelin test ve eğitim hata matris sonuçlarına sırasıyla Tablo 3.41 ve Tablo 3.42'de yer verilmiştir. Yanlış pozitif tahmin sayısı varsayılan ayarlarla elde edilen sonuçlara göre bir uçuş daha fazla tahmin edildiği halde hem varsayılan ayarlarla hem de SMOTE tekniği ile elde edilen yanlış negatif sayılarında düşüş elde edilmiştir. 705 yanlış negatif tahmin ile CatBoost teknikleri içindeki en düşük yanlış negatif tahmin sayısı bulunmuştur. Bu durumda tüm CatBoost yöntemleri içinde Bayes hiper-parametre ayarı ile elde edilen modelin en tercih edilebilir model olduğunu göstermektedir.

**Tablo 3.41 Hiper-Parametre Ayarları ile CatBoost Eğitim Hata Matrisi**

Tahmin	Pozitif	7870	97
	Negatif	452	25494
		Pozitif	Negatif
Gerçek			

**Tablo 3.42 Hiper-Parametre Ayarları ile CatBoost Test Hata Matrisi**

Tahmin	Pozitif	2920	111
	Negatif	705	10799
		Pozitif	Negatif
Gerçek			

### 3.2.6. Tüm Modellerin Performans Metrikleri Aracılığı ile Karşılaştırılması

Belirlenmiş olan performans metriklerine göre modellerin performans sonuçları Tablo 3.43'te sunulmuştur. Tablodaki kullanılan tüm yöntemlerin belirlenmiş olan performans metriklerine göre hem eğitim hem de test sonuçlarına yer verilmiştir.

**Tablo 3.43 Varsayılan Ayarlara Göre Model Performans Sonuçları**

		Kesinlik	Duyarlılık	F1	F2	Cohen Kappa	Doğruluk	EAKA
YSA	Eğitim	1	1	1	1	0,99	1	0,99
	Test	0,43	0,34	0,38	0,36	0,21	0,72	0,6
YSA Çift Katman	Eğitim	0,99	1	0,99	0,99	0,99	1	0,99
	Test	0,53	0,56	0,55	0,55	0,39	0,77	0,7
Rastgele Ormanlar	Eğitim	1	1	1	1	1	1	1
	Test	0,8	0,1	0,1	0,1	0,1	0,8	0,5
XGBoost	Eğitim	0,98	0,99	0,99	0,99	0,98	0,99	0,98
	Test	0,96	0,95	0,96	0,95	0,94	0,98	0,97
LightGBM	Eğitim	0,97	0,90	0,94	0,92	0,92	0,97	0,95
	Test	0,96	0,86	0,91	0,88	0,88	0,96	0,93
CatBoost	Eğitim	0,97	0,77	0,86	0,80	0,81	0,94	0,88
	Test	0,96	0,74	0,83	0,77	0,79	0,93	0,86

YSA modeli eğitim aşamasında eğitim veri setine göre çok iyi uyum sağlamış ve Cohen Kappa ve EAKA metriklerinde 0,99 değerini alırken diğer performans metriklerinde %100 sonucunu almıştır. Buradaki öğrenme ise test sonuçlarına yansımadağından aşırı öğrenme olduğu ortaya çıkmıştır. Kesinlik değeri 0,43 olarak elde edilmiş olsa da model için daha önemli olduğu belirtilen duyarlılık ve F2 değerlerinde daha düşük sonuçlar elde edilmiştir. En yüksek olan test sonucu doğrulukta elde edilmiş olsa da bu değer tek başına dengesiz bir veri setini değerlendirmek için yeterli olmadığından tercih edilebilir bir sonuç ortaya çıkmamıştır.

Çift katmanlı YSA modelinde ise eğitim tek katmanlı YSA modeline benzer bir şekilde çok yüksek değerler ile tamamlanmıştır. Test aşamasında ise tüm sonuçlar tek katmanlı YSA modeline göre daha iyi çıkmıştır. Duyarlılık 0,56 değeri ile daha iyi sonuç vermiştir. F2 değeri de benzer bir şekilde 0,55 oranında elde edilmiştir. Cohen Kappa değerine göre 0,39 ile orta derecede bir uyuma olduğu görülmüştür. Model tek katmanlı YSA modeline başarılı olsa da genel olarak değerlendirildiğinde orta dereceli bir başarı ortaya koymuştur. 0,77 ile en yüksek performans metriği doğruluk ile elde edilmiş olsa dahi dengesiz veri setinde yanıtıcı sonuç verebilen bir performans metriği olduğundan modelin başarılı olarak değerlendirilmesi için yeterli değildir.

Rastgele ormanlar yöntemi eğitim aşamasında sonuçlarla tamamen uyumlu bir öğrenme gerçekleştirse de test aşamasında bu başarıyı sürdürememiştir. Kesinlik, doğruluk ve EAKA

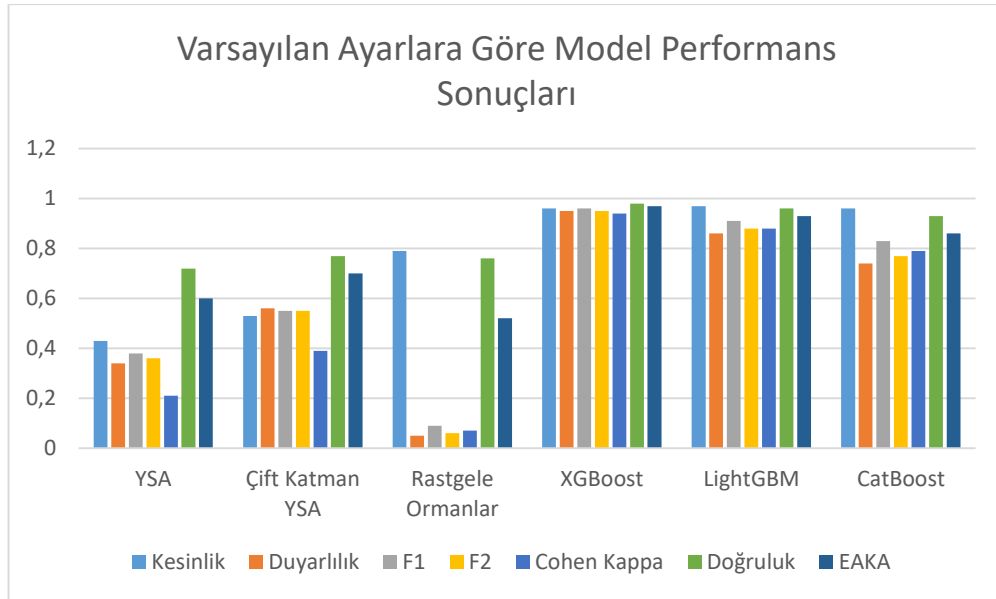
performans metriklerinde yüksek başarı sağlamanın nedeni genellikle uçuşların geç kalmayacağı yönünde bir tahmin yapması sayesinde gerçekleşmiştir. Veri setindeki çoğu uçuş geç kalmadığından dolayı böyle bir tahmin bu üç performans metriğini yükseltse de model için önem taşıyan duyarlılık, F1 ve F2 performans metriklerinde 0,1 ile son derece düşük bir başarı sergilemiştir. Bu sebeple eğitilen modelin kullanılarak tahmin yapılması uygun olmaktan uzaktır.

XGBoost tekniği ile hem test değerleri hem de eğitim değerleri tüm seçilen performans metrikleri için 0,90 üzeri olarak elde edilmiştir. Eğitim ve test aşamasındaki benzer sonuçlar başarılı bir eğitim sürecinin geçtiğinin test girdileri ile de çok yüksek başarı sağlanabildiğini göstermektedir.

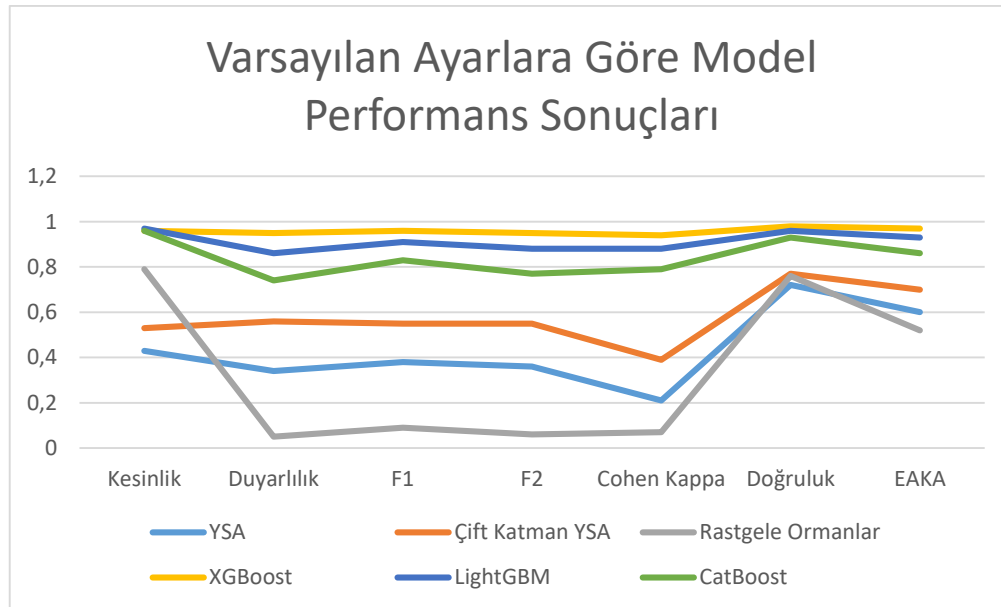
LightGBM tekniğinin varsayılan model ayarları ile sonuçları incelendiğinde ise yüksek eğitim ve test sonuçları ile iyi bir eğitim aşaması gerçekleştiği görülmektedir. Fakat XGBoost yöntemine göre model sonuçları düşüş göstermiştir. Doğruluk, kesinlik, F1 ve EAKA alan metriklerinde %90 üzeri başarı sağlanan sonuçlar alınırken, duyarlılık, F2 Cohen Kappa değerlerinde 0,90 altına hafif de olsa düşen sonuçlarla performans metrikleri sonuçları alınmıştır.

CatBoost tekniği ile de eğitim ve test sonuçları benzer başarı gösterirken duyarlılık ve F2 değerlerinde sırasıyla 0,74 ve 0,77 oranında başarı elde edilmiştir. Bu durum da diğer gradyan yükseltme temelli modellere göre daha geride kalan sonuçlar olmuştur. Başarılı bir model olarak değerlendirilebilecek olsa da diğer gradyan yükseltme tekniklerine göre geride kalan sonuçlar olarak değerlendirilmiştir.

Şekil 3.12’te Tablo 3.43 üzerinde paylaşılmış olan test performans metrikleri sonuçları sütun grafik olarak, Şekil 3.13’te ise performans metriklerine göre çizgi grafik olarak gösterimine yer verilmiştir. Bu grafiklere göre de kesinlik metriğine göre en iyiden en kötüye doğru sonuç veren yöntemler sırasıyla XGBoost, LightGBM ve CatBoost sonrasında rastgele ormanlar, çift katmanlı YSA ve YSA olmuştur. Duyarlılık, F1, F2, Cohen Kappa, EAKA performans metrikleri sonuçlarına göre performans sıralaması yapıldığında XGBoost, LightGBM, CatBoost, çift katmanlı YSA, YSA ve rastgele ormanlar yöntemleri olarak sıralama elde edilmiştir. Doğruluk metriğine göre ise XGBoost, LightGBM, rastgele ormanlar, çift katman YSA, YSA olarak yüksekten düşüğe doğru sıralanmış bir performans sıralaması elde edilmiştir.



Şekil 3.12 Varsayılan Ayarlara Göre Model Performans Sonuçları Sütun Grafik Gösterimi



Şekil 3.13 Varsayılan Ayarlara Göre Model Performans Sonuçları Çizgi Grafik Gösterimi

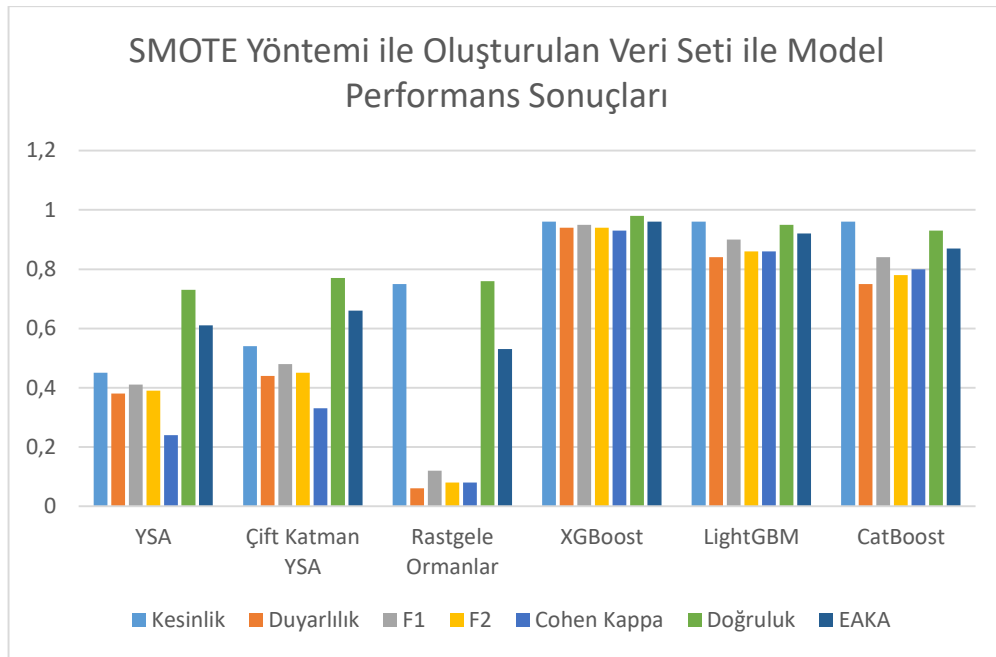
Dengesiz veri setleri için literatürde önerilen yöntemlerden biri olan sentetik veri üretimine başvurulmuş ve ayrıca bir deneme yapılmıştır. Bu denemelerin sonucunda elde edilen performans sonuçları Tablo 3.44'te gösterilmiştir. Tablo incelendiğinde tek katman YSA modelinin SMOTE tekniği ile kullanılmasının sonuçlar üzerinde olumlu etkisi olduğu görülmüştür. Tüm performans metriklerinde gelişme sağlanmıştır ancak model diğer gelişmiş modellere göre hala daha düşük performans göstermektedir. Çift Katmanlı YSA modelinde ise doğruluk ve kesinlik dışında ki performans metriklerinde gerileme görülmektedir. Rastgele ormanlar tekniğinde model performansı çok değişmemiş ve hala kullanılabilir bir model olmaktan uzak bir model olduğunu performans metrikleri ortaya koymaktadır. XGBoost tekniğinde ise test sonuçlarında kesinlik ve doğruluk performans metrikleri dışındaki

metriklerde düşüş görülmüştür. SMOTE tekniği kullanılmadan yapılan deneme XGBoost için daha başarılı sonuçlar vermektedir. LightGBM tekniğinde ise kesinlik metriği haricinde tüm performans metriklerinde gerileme olduğu görülmektedir. CatBoost tekniğinde ise SMOTE ile daha başarılı bir sınıflandırma yapılmış ve duyarlılık, F1, F2, Cohen Kappa ve EAKA performans metriklerinde iyileşme sağlanmıştır. Gradyan yükseltme temelli teknikler içinde bir tek CatBoost ile SMOTE tekniği bir arada kullanıldığında iyileşme sağlandığı tespit edilmiştir.

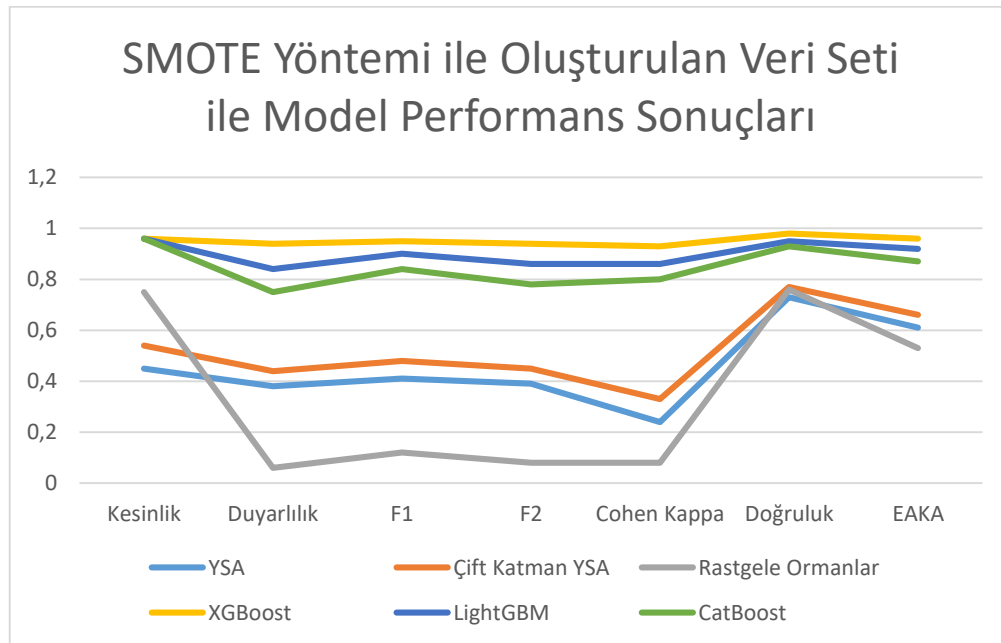
**Tablo 3.44 SMOTE Yöntemi ile Oluşturulan Veri Seti ile Model Performans Sonuçları**

		<b>Kesinlik</b>	<b>Duyarlılık</b>	<b>F1</b>	<b>F2</b>	<b>Cohen Kappa</b>	<b>Doğruluk</b>	<b>EAKA</b>
YSA	Eğitim	1	1	1	1	0,99	1	0,99
	Test	0,45	0,38	0,41	0,39	0,24	0,73	0,61
YSA Çift Katman	Eğitim	1	0,99	1	0,99	0,99	1	0,99
	Test	0,54	0,44	0,48	0,45	0,33	0,77	0,66
Rastgele Ormanlar	Eğitim	1	1	1	1	1	1	1
	Test	0,75	0,06	0,12	0,08	0,1	0,76	0,53
XGBoost	Eğitim	0,99	0,99	0,99	0,99	0,98	0,99	0,99
	Test	0,96	0,94	0,95	0,94	0,93	0,98	0,93
LightGBM	Eğitim	0,99	0,96	0,97	0,97	0,95	0,97	0,97
	Test	0,96	0,84	0,90	0,86	0,86	0,95	0,92
CatBoost	Eğitim	0,99	0,93	0,93	0,95	0,96	0,96	0,96
	Test	0,96	0,75	0,84	0,78	0,8	0,93	0,87

Şekil 3.14'te SMOTE yöntemi ile elde edilen modellerin test performanslarının sütun grafiği ile gösterilmiştir. Grafik incelendiğinde genel olarak hepsinde başarı sağlayan modelin XGBoost olduğu görülmektedir. Şekil 3.15'te ise performans metriklerine göre modellerin test aşamasındaki başarı durumu çizgi grafik ile gösterilmektedir. Grafiğe göre kesinlik metriğine göre XGBoost, LightGBM ve CatBoost 0,96 oranında başarı sağlamışlardır. Bu başarıyı rastgele ormanlar, çift katmanlı YSA ve YSA takip ettiği görülmektedir. Duyarlılık, F1, F2, Cohen Kappa ve EAKA metriklerine göre ise performans sıralaması varsayılan ayarlar ile benzer bir şekilde XGBoost, LightGBM, CatBoost, Çift Katman YSA, YSA ve rastgele ormanlar olduğu görülmüştür. Doğruluk kriterine göre ise en yüksek performans XGBoost sonrasında LightGBM, rastgele ormanlar, çift katmanlı YSA ve YSA olarak elde edilmiştir.



Şekil 3.14 SMOTE Yöntemi ile Oluşturulan Veri Seti ile Modellerin Test Performans Sonuçları Sütun Grafik Gösterimi



Şekil 3.15 SMOTE Yöntemi ile Oluşturulan Veri Seti ile Modellerin Test Performans Sonuçları Çizgi Grafik Gösterimi

Bayes ile hiper-parametre ayarları yapılmış olan modellerin eğitim ve test performans sonuçları ise Tablo 3.45 üzerinde paylaşılmıştır. Yapay sinir ağları yönteminin Bayes ile hiper parametre ayarına Burada rastgele ormanlar yönteminin zamanında kalkacak uçakları çok iyi tahmin etmiş olduğu kesinlik metriğine göre ortaya koyulmuş olsa da geç kalan uçuşların çok azının tahmini doğru yapılamadığı için duyarlılık, F1 ve F2 0,01 olarak elde edilmiştir. Burada da eğitim ve test doğruluk değerlerine bakarak başarı değerlendirilmesinin yapılmasının yanıltıcı olduğu görülmüştür. Doğruluk ve kesinlik dışındaki tüm metrikler kullanılamaz bir



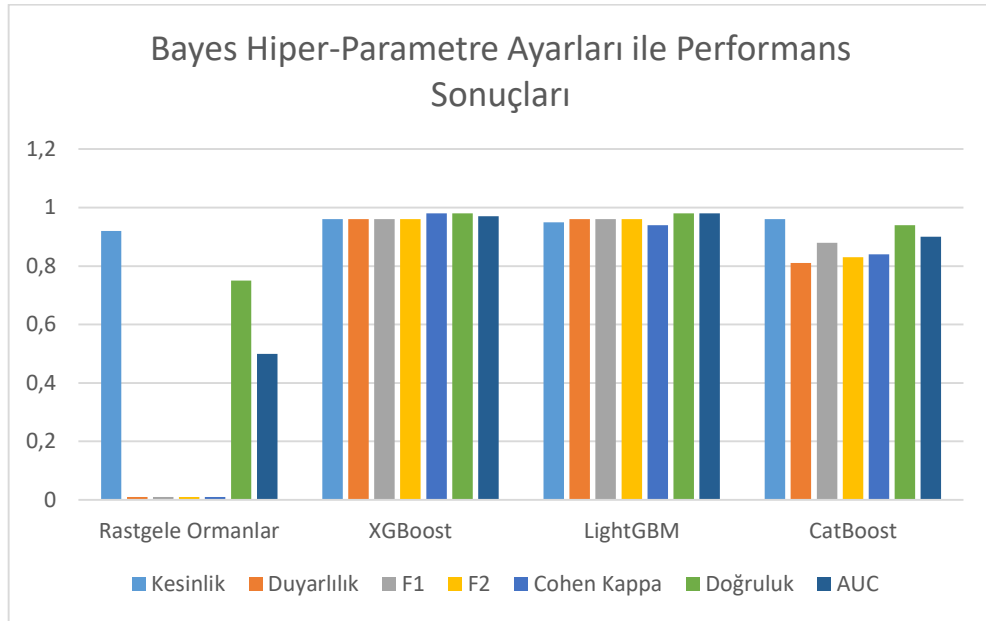
modele işaret etmektedir. Daha önceki denemelerde yüksek başarı gösteren XGBoost yönteminin test sonuçlarında daha önceki denenen modellere göre başarının biraz daha arttığı görülmektedir. Benzer bir şekilde hiper-parametre ayarları sonrası LightGBM modeli önceki modellerden çok daha iyi performans sergilemiştir. Hiper parametre ayarları incelenen dört yöntem içinde en yüksek başarıyı göstermiştir ve ilk defa XGBoost modeline yakın performans değerleri LightGBM hiper-parametre ayarları sonucunda elde edilmiştir. CatBoost yöntemi de daha önceki denemelere göre çok daha iyi sonuç vermektedir. Performans metrikleri diğer gradyan yükseltme modellerine göre geride kaldıysa da tüm performans metriklerinden %80 üzeri oranda başarı sağladığı görülmektedir.

**Tablo3.45 Bayes Hiper-Parametre Ayarları ile Modellerin Performans Sonuçları**

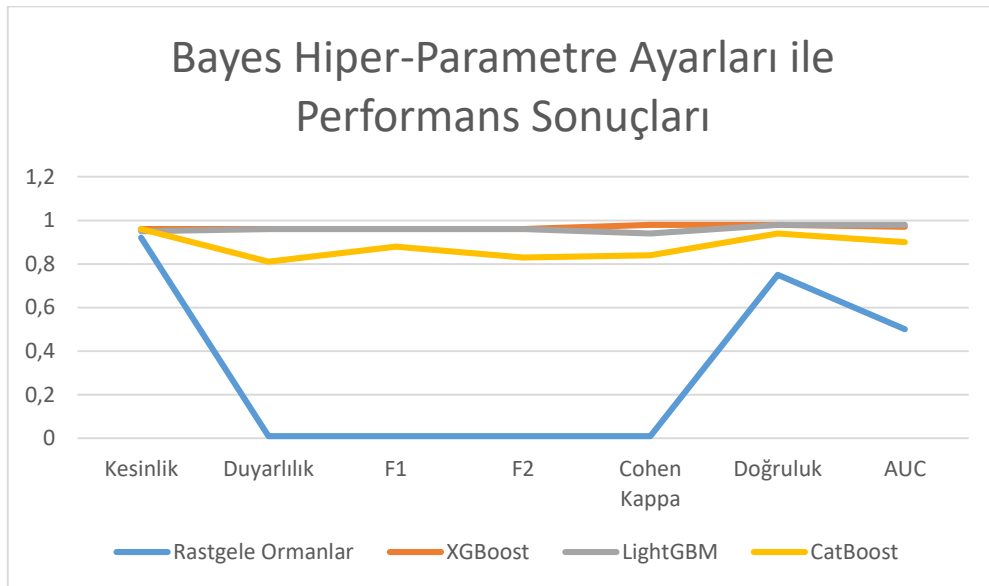
		<b>Kesinlik</b>	<b>Duyarlılık</b>	<b>F1</b>	<b>F2</b>	<b>Cohen Kappa</b>	<b>Doğruluk</b>	<b>EAKA</b>
Rastgele Ormanlar	Eğitim	1	0,14	0,24	0,17	0,19	0,79	0,57
	Test	0,92	0,01	0,01	0,01	0,01	0,75	0,5
XGBoost	Eğitim	0,98	0,99	0,98	0,99	0,98	0,99	0,98
	Test	0,96	0,96	0,96	0,96	0,94	0,98	0,97
LightGBM	Eğitim	1	1	1	1	1	1	1
	Test	0,95	0,96	0,96	0,96	0,94	0,98	0,98
CatBoost	Eğitim	0,99	0,95	0,97	0,95	0,95	0,98	0,97
	Test	0,96	0,81	0,88	0,83	0,84	0,94	0,90

Şekil 3.16'da ise Bayes hiper-parametre ayarı ile optimum hiper-parametre ayarları tespit edilen tüm karar ağacı temelli yöntemlerin test performans sonuçlarına sütun grafik aracılığı ile yer verilmiştir. Rastgele ormanlar yöntemi dışında diğer yöntemler ile başarılı sonuçlar elde edildiği bu grafik üzerinde de görülebilmektedir. Şekil 3.17'de ise hangi performans metriğinde hangi modelin başarılı olduğunu gösterebilmek amacıyla test performans sonuçlarının çizgi grafiğine yer verilmiştir. Bu şekilde göre ise kesinlik metriğine göre XGBoost ve CatBoost başta olmak üzere sonrasında LightGBM ve son olarak rastgele ormanlar yöntemi olarak başarı sıralaması gerçekleşmiştir. Duyarlılık, F1, F2, Cohen Kappa, doğruluk metriklerine göre en başarılı yöntemler XGBoost ve LightGBM olmuştur sonrasında CatBoost ve son olarak rastgele ormanlar yöntemi gelmektedir. EAKA alana göre ise en başarılı yöntem LightGBM olmuştur sonrasında XGBoost, CatBoost ve rastgele ormanlar yöntemleri başarı sergilemiştir. Hiper-parametre ayarı sonrası en düşük başarıyı rastgele ormanlar çok yüksek eğitim ve çok düşük test sonucu ile aşırı öğrenme sonucu göstermiştir. XGBoost ile LightGBM ise hem eğitim hem

de test aşamasında yüksek performans değerleri elde etmiş bu duruma göre de başarılı bir öğrenme gerçekleştirmişlerdir.



**Şekil 3.16 Bayes Hiper-Parametre Ayarları ile Modellerin Test Performans Sonuçları Sütun Grafiği**



**Şekil 3.17 Bayes Hiper-Parametre Ayarları ile Modellerin Test Performans Sonuçları Çizgi Grafiği**

## SONUÇ

Uçakların geç kalması ile ilgili yapılan çalışmalar; gecikmelerin havayolu şirketleri, kullanıcılar ve çevre üzerinde olumsuz etkiler yarattığını ortaya koymuştur. Bu durum göz önünde bulundurularak, uçak gecikmelerinin önceden tahmin edilmesinin önemi ve gerekliliği anlaşılmıştır. Bu doğrultuda makine öğrenmesi teknikleri ile yapılmış olan çalışmalardan farklı olarak, daha önce uygulanmamış yöntemlerin bir arada kullanılarak karşılaştırmalı analizlerin sunulduğu bu çalışmada, doğru yöntem seçimi sayesinde gecikmelerin önceden tahmini yapılarak uçak gecikmelerinin çok yüksek bir oranda doğru tahmin edilebilmesinin mümkün olduğu ortaya koyulmuştur. Bu sonuç, uluslararası bir havayolu şirketinin üç yıllık uçuş verileri kullanılarak kısıtlı bir zaman aralığı için elde edilmiş olsa da, test sonuçlarına göre diğer uçuşlar için de uygulandığında olumlu ve benzer sonuçlar elde edileceği tahmin edilmektedir.

Kullanılan algoritmaların performans durumları hata matrisi, doğruluk, kesinlik, duyarlılık, F-değerleri ve Cohen Kappa istatistiği, EAKA gibi performans metrikleri ile ölçülmüş ve elde edilen sonuçlar birbirleriyle karşılaştırılmıştır. Bu bağlamda üç farklı model oluşturularak tüm performans metriklerine göre karşılaştırmalar yapılmıştır. Seçilen makine öğrenmesi tekniklerinin Python programı içindeki varsayılan ayarları ile yapılarak oluşturulan modeller ilk denemeyi, sentetik veri üretimi ile dengesiz veri seti problemi aşarak elde edilen modeller ikinci denemeyi ve son olarak Bayes ile gerçekleştirilen hiper-parametre ayarları yapılarak elde edilen algoritma modelleri ise üçüncü denemeyi oluşturmaktadır. Oluşturulan bu deneme modellerinin sonuçları elde edilerek tüm sonuçların birbiri ile karşılaştırılması imkanı elde edilmiştir. Kullanılan yöntemler yapay sinir ağları, rastgele ormanlar, XGBoost, LightGBM ve CatBoost yöntemi olmuştur. YSA modeli yapay zekânın ortaya çıktığı zamandan beri bilinen ve kullanılan bir yöntem olması dolayısıyla modele eklenmiştir. Sonuçları ise diğer modellere göre yetersiz olmuştur. Makine öğrenme teknikleri ile uçak gecikmeleri tahmini yapılırken rastgele ormanlar tekniğinin en çok başvurulan yöntem olduğu görülmüştür. Bu sebeple kullanılan yöntemlere rastgele ormanlar yöntemi de dahil edilmiştir. Fakat kullanılan veri seti için rastgele ormanlar tekniği denendiğinde; kurulan alternatif üç deneme modeli için de aşırı öğrenme gerçekleşmiş, bu veri seti için kullanılması mümkün olmayan bir teknik olarak karşımıza çıkmıştır. En iyi sonucu veren yöntemin, varsayılan ayarlar ile çalıştırılmış modeller içinde de SMOTE yöntemi ile çalıştırılmış tüm yöntemler içinde de XGBoost olduğu görülmüştür. Bayes hiper-parametre ayarı ile yapılan denemelerde ise XGBoost ve LightGBM tekniğinin çok yakın oranlar ile en iyi sonucu veren yöntemler tespit edilmiştir. Birbirlerine rakip olarak gösterilen XGBoost, LightGBM ve CatBoost yöntemleri içinde bu veri seti için

CatBoost yöntemi üçüncü sırada yer almış, kullanılabilir sonuçlar sunmakla beraber diğer tekniklerin performans metriklerinin değerlerini yakalayamadığı görülmüştür.

Çalışmanın faydası olarak; elde edilen sonuçlar ışığında havayolu firması tarafından geciken uçuşların yolculara ve hava kargo firmalarına önceden haber verilmesi, uçuşu daha önceki bir zaman diliminde erteleyerek veya farklı aksiyonlar alınması sağlanarak hem uçuşların organizasyonu hem de müşteri memnuniyeti açısından olumlu etkiler yaratılmasının mümkün olabileceği düşünülmektedir. Yolcular için uçak ile seyahat süresinde beklemelemlerin çok ciddi bir zaman almasının müşteri memnuniyeti üzerindeki negatif etkisi tartışılmıştır. Uçak gecikmelerinin önceden tahminin havayolu firmalarının esnekliğini artırarak uygun çözüm sunmalarına yardımcı olması beklenmektedir. Böylece müşteri memnuniyetinin artırılması hususunda ek çalışmalar yapılırken marka değerini de artırmak mümkün olacaktır. ICAO'nun önerileri ile paralel bir şekilde gecikme yaşanması muhtemel uçuşlarda önem alınması mümkün olacak ve yakıt tasarrufu sağlamak için de fırsat yaratılmış olacaktır. Az yakıt kullanımının ve kaynakların verimli kullanımının aynı zamanda çevreye de olumlu katkısı olacağı da literatür tarafından desteklenmektedir. Olası faydaların küresel ısınma probleminin çözümüne olumlu katkılar yapacağı düşünülmektedir.

Veri seti hazırlanırken kalkış havalimanındaki hava durumu ile varış havalimanındaki hava durumu ele alınmıştır. Bunların yanı sıra uçağın istikametinde karşılaşacağı hava durumu da uçuş güvenliği ve süresi açısından son derece önemlidir. Bu çalışmada yer verilemeyen bu veri çalışmanın ana kısıtlarından birini oluşturmuştur. Bu durumun yanı sıra veri seti büyüklüğü ve hiper-parametre ayarları yapılırken uzay büyüklüğü arttıkça makine öğrenmesi teknikleriyle elde edilen sonuçlar gecikmekte ve bazı durumlarda bellek yetersizliği problemi ortaya çıkmaktadır. Donanım yetersizliği sebebi ile sadece karar ağacı temelli yöntemlerde hiper-parametre ayarlarına yer verilmiş, diğer hiper-parametre tekniklerine yer vermek ise bu sebeple mümkün olmamıştır.

Gelecek çalışmalarda havayolu şirketinden gelecek periyotların verileri alınarak elde edilen sonuçların geçerliliğinin daha iyi değerlendirilebileceği düşünülmektedir. Bunun yanı sıra birbirine bağlı uçuş verileri bilgisi alınarak aktarmalı uçakların durumunun da incelenebileceği önerilmektedir. Başka bir yaklaşımla gecikecek uçuşların hangi aşamada gecikme yaşayacağı tahmini yapılarak, havayolu şirketine gecikmeyle ilgili hangi aşamaya odaklanması gerektiği konusunda ek fikir vererek operasyonlarını yönetmesi konusunda destek verilebilir. Bu gelişmeler sonucu elde edilecek modellerin şirketin hali hazırda kullanmakta olduğu yazılım ile birleştirilerek makine öğrenmesi tekniklerinin sürekli yeni veri ile beslenmesinin sağlanması; böylece sürekli olarak daha iyi tahminler yapabilmesi sağlanabilir.

Ayrıca kullanıcı dostu bir arayüz ile kullanışlı hale getirerek şirketin kolayca kullanabileceği bir hale dönüştürülmesinin fayda sağlanabileceği düşünülmektedir.

## KAYNAKÇA

- Abdelghany, K. F., Shah, S. S., Raina, S. ve Abdelghany, A. F. (2004). "A model for projecting flight delays during irregular operation conditions". *Journal of Air Transport Management*, 10(6): 385-394.
- Acuña, E. (2011). *Preprocessing in Data Mining*. Springer, Berlin. doi:10.1007/978-3-642-04898-2
- Aksu, G. ve Doğan, N. (2018). "Veri Madenciliğinde Kullanılan Öğrenme Yöntemlerinin Farklı Koşullar Altında Katılaştırılması". *Ankara Üniversitesi Eğitim Bilimleri Fakültesi Dergisi*, 51(3): 71-100.
- Ali, J., Khan, R., Ahmad, N. ve Maqsood, I. (2012). "Random Forests and Decision Tree". *IJCSI International Journal of Computer Science Issues*, 9(5): 272-278.
- Alpaydın, E. (2014). *Introduction to Machine Learning*. The MIT Press, London.
- Alpaydın, E. (2017). *Yapay Öğrenme*. Boğaziçi Üniversitesi Yayınevi, İstanbul.
- Amit, Y. ve German, D. (1997). "Shape quantization and recognition with randomized trees". *Neural Computation*, 9(7): 1545-1588.
- Arbib, M. A. (2003). *The Handbook of Brain Theory and Neural Networks Part I: Background The Elements of Brain Theory and Neural Networks*. The MIT Press, London.
- Armstrong, S. (2001). *Principles of forecasting: a handbook for researchers and practitioners*. Springer, New York.
- Aydın, M. A. (2021). "Müşteri Kaybı Tahmininde Sınıf Dengesizliği Problemi". *Politeknik dergisi*, 24: 1-12.
- Batista, G. E., Prati, R. ve Monard, M. (2004). "A study of the behavior of several methods for balancing machine learning training data". *ACM SIGKDD Explorations Newsletter*, 6(1): 20-29.
- Beatty, R., Hsu, R., Berry, L. ve Rome, J. (1999). "Preliminary Evaluation of Flight Delay Propagation through an Airline Schedule". *Air Traffic Control Quarterly*, 7(4): 259-270.
- Belcastro, L., Marozzo, F., Talia, D. ve Trunfio, P. (2016). "Using Scalable Data Mining for Predicting Flight Delays". *ACM Transactions on Intelligent Systems and Technology*, 8(1): 1-20.
- Bellman, R. E. (2015). *Adaptive control processes: a guided tour*. Princeton University, London.
- Bergstra, J. S., Bardenet, R., Bengio, Y. ve Kégl, B. (2011). "Algorithms for hyper-parameter optimization". *Advances in neural information processing systems*, 24: 2546-2554.

- Bergstra, J. ve Bengio, Y. (2012). Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research*, 13: 281-305.
- Bostrom, N. (2020). *Süper Zeka* (2. Baskı). Koç Üniversitesi Yayınları, İstanbul.
- Bradley, A. P. (1996). "The use of area under the roc curve in the evaluation of machine learning algorithms". *Pattern Recognition Society*, 30(7): 1145-1159.
- Bramer, M. (2016). *Principles of Data Mining*. Springer, UK.
- Branco, P., Torgo, L. ve Ribeiro, R. (2015). "A survey of predictive modelling under imbalanced distributions". *arXiv preprint*, 1-48.
- Breiman, L. (1996) "Bagging predictors". *Machine learning*, 24(2), 123-140
- Breiman, L. (1998). "Arcing classifier (with discussion and a rejoinder by the author)". *The annals of statistics*, 26(3): 801-849.
- Breiman, L. (2001). "Random Forests". *Machine Learning*, 45(1): 5-32.
- Britto, R., Dresner, M. ve Voltes, A. (2012). "The impact of flight delays on passenger demand and societal welfare". *Transportation Research Part E: Logistics and Transportation Review*, 48(2): 460-469.
- Brooks, P. W. (1967). "The Development of Air Transport". *Journal of Transport Economics and Policy*, 1(2): 164-183.
- Bulut, F. (2017). "Örnek tabanlı sınıflandırıcı topluluklarıyla yeni bir klinik karar destek sistemi". *Journal of the Faculty of Engineering and Architecture of Gazi University*, 32(1): 65-76.
- Cao, K., Wei, C., Gaidon, A., Arechiga, N. ve Ma, T. (2019). "Learning imbalanced datasets with label-distribution-aware margin loss". *Advances in Neural Information Processing Systems*, 1567-1578.
- Chakrabarty, N., Kundu, T., Dandapat, S., Sarkar, A. ve Kole, D. K. (2019). "Flight Arrival Delay Prediction Using Gradient Boosting Classifier". *Emerging Technologies in Data Mining and Information Security*, 813: 651-659.
- Chandramouleeswaran, K. R., Krzemien, D., Burns, K. ve Tran, H. T. (2018). "Machine Learning Prediction of Airport Delays in the US Air Transportation Network". *Aviation Technology, Integration, and Operations Conference*. 25-29 Temmuz 2018, Atlanta, Georgia, 1-10.
- Chawla, N. V., Bowyer, K., Hall, L. O. ve Kegelmeyer, W. P. (2002). "SMOTE: Synthetic Minority Over-sampling Technique". *Journal of Artificial Intelligence Research*, 16(1): 321-357. doi:10.1613/jair.953

- Chawla, N. V., Hall, L. O. ve Joshi, A. (2005). "Wrapper-based computation and evaluation of sampling methods for imbalanced datasets". *Proceedings of the 1st international workshop on Utility-based data mining*. 21 Ağustos 2005, Chicago, 24-33.
- Chen, J. ve Li, M. (2019). "Chained Predictions of Flight Delay Using Machine Learning". *AIAA SciTech Forum*. 7-11 Ocak 2019, San Diego, 1-25.
- Chen, T., ve Guestrin, C. (2016). "XGBoost: A scalable tree boosting system". *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 13-17 Ağustos 2016, San Francisco, 785-794.
- Chen, X., Yu, H., Cao, K., Zhou, J., Wei, T. ve Hu, S. (2020). "Uncertainty-Aware Flight Scheduling for Airport Throughput and Flight Delay Optimization". *IEEE Transactions on Aerospace and Electronic Systems*, 56(2): 853-862.
- Choi, S., Kim, Y. J., Simon, B. ve Mavris, D. (2017). Cost-sensitive Prediction of Airline Delays Using Machine Learning. *36th Digital Avionics Systems Conference (DASC)* (s. 1-8). IEEE.
- Choi, S., Kim, Y. K., Briceno, S. ve Mavris, D. (2016). Prediction of Weather-induced Airline Delays Based on Machine Learning Algorithms. *35th Digital Avionics Systems Conference (DASC)*, 16-21 Eylül 2017, St. Petersburg: 1-6.
- Chowdhury, M., ve Sadek, A. W. (2012). "Advantages and Limitations of Artificial Intelligence" . *Artificial Intelligence Applications*, 6: 1-8.
- Clemen, R. ve Winkler, R. (1986). "Combining economic forecasts". *Journal of Business & Economic Statistics*, 4(1): 39-46.
- Coates, A. ve Ng, A. Y. (2011). "The importance of encoding versus training with sparse coding and vector quantization". *International Conference on Machine Learning*. 28 Haziran-2 Temmuz 2011, Bellevue, 1-8.
- Coates, A., Lee, H. ve Ng, A. Y. (2011). "An Analysis of Single-Layer Networks in Unsupervised Feature Learning". *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 11-13 Nisan 2011, Fort Lauderdale, 215-223.
- Cohen, J. (1960). "A coefficient of agreement for nominal scales". *A coefficient of agreement for nominal scales*, 20(1): 37-46.
- Cook, A. ve Tanner, G. (2009). "The challenge of managing airline delay costs" . *Conference on Air Traffic Management (ATM) Economic*, 10 Eylül 2009, University of Belgrade, 1-10.



- Cutler, A., Cutler, R. D. ve Stevens, J. R. (2011). "Ensemble machine learning". Zhang, C. ve Ma, Y. (Ed.), *Random Forests*. Springer, Boston, 157-175.
- Daley, B. (2016). *Air transport and the environment*. Routledge, New York.
- Daoud, E. (2019). "Comparison between XGBoost, LightGBM and CatBoost Using a Home Credit Dataset". *Computer Science*, 13(1): 6-10.
- Das, S., Dey, A., Pal, A. ve Roy, N. (2015). "Applications of Artificial Intelligence in Machine Learning: Review and Prospect". *International Journal of Computer Applications*, 115(9): 31-41.
- Davis, J. ve Goadrich, M. (2006). "The Relationship Between Precision-Recall and ROC Curves". *ACM International Conference Proceeding Series*, Kasım 2006, Markham Ontario Canada, 233-240.
- Delgado, R. ve Tibau, X. (2019). Why Cohen's Kappa should be avoided as performance measure in classification. *Plos One*, 14(9): 1-26.
- Dietterich, T. G. (2000). "Ensemble Methods in Machine Learning". *International workshop on multiple classifier systems*, 21-23 Temmuz, Cagliari, İtalya, 1-15.
- Ding, Y. (2017). Predicting flight delay based on multiple linear regression. *IOP Conference Series: Earth and Environmental Science*, 4-7 Eylül 2017, Zvenigorod, 1-7.
- Domhan, T., Springenberg, J. T. ve Hutter, F. (2015). "Speeding up Automatic Hyperparameter Optimization of Deep Neural Networks by Extrapolation of Learning Curves". *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 25-31 Temmuz 2015, Buenos Aires, Argentina, 1-9.
- Domingos, P. (2017). *Master Algoritma Yapay Öğrenme Hayatımızı Nasıl Değiştirecek?* Paloma, İstanbul.
- Dorogush, A. V., Ershov, V. ve Gulin, A. (2018). CatBoost: gradient boosting with categorical features support. *arXiv preprint arXiv:1810*, 1-7.
- Dou, X. (2020). Flight Arrival Delay Prediction And Analysis Using Ensemble Learning. *4th Information Technology, Networking, Electronic and Automation Control Conference*, 12-14 Temmuz, Chongqing, Çin, 836-840.
- Elele, O. (2009). *Küresel Krize Karşı Alınan Önlemler ve Krizin Gidişatı*. Sermaye Piyasası Aracı Kuruluşları Birliği, İstanbul
- Elmas, Ç. (2007). *Yapay Zeka Uygulamaları*. Seçkin, Ankara.
- Ertel, W. (2011). *Introduction to Artificial Intelligence*. Springer, Weingarten.
- Ertel, W. (2017). *Introduction to Artificial Intelligence (2. Baskı)*. Springer, Weingarten.

- Eun, Y., Hwang, I. ve Bang, H. (2010). "Optimal Arrival Flight Sequencing and Scheduling Using Discrete Airborne Delays". *IEEE Transactions on Intelligent Transportation Systems*, 11(2): 359-373.
- Ferri, C., Hernández-Orallo, J. ve Modroi, R. (2009). An experimental comparison of performance measures for classification. *Pattern Recognition Letters*, 30(1): 27-38.
- Feurer, M., Springenberg, J. T. ve Hutter, F. (2015). "Initializing bayesian hyperparameter optimization via meta-learning". *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 25-30 Ocak 2015, Austin, Texas, 1-8).
- Freund, Y. ve Schapire, R. E. (1996). "Experiments with a new boosting algorithm," *Machine Learning: Proceedings of the Thirteenth International Conference*, 3-6 Temmuz 1996 , Bari, İtalya, 148-156.
- Friedman, J. H. (2001). "Greedy function approximation: a gradient boosting machine". *Annals of statistics*, 29: 1189-1232.
- Gangawar, V. (2012). "An overview of classification algorithms for imbalanced datasets". *International Journal of Emerging Technology and Advanced Engineering*, 2.: 42-47.
- Gareth, J., Witten, D., Hastie, T. ve Robert, T. (2013). *An Introduction to Statistical Learning*. New York: Springer. Springer, New York.
- Geman, S., Bienenstock, E. ve Doursat, R. (1992). Neural Networks and the Bias/Variance Dilemma. *Neural Computation*, 4(1): 1-58.
- Geron, A. (2019). *Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow*. O'Reilly, Sebastopol.
- Ghawi, R. ve Pfeffer, J. (2019). "Efficient Hyperparameter Tuning with Grid Search for Text Categorization using kNN Approach with BM25 Similarity". *Open Computer Science*, 1(9):160-180.
- Goodfellow, I., Bengio ve Courville, A. (2016). *Deep Learning*. MIT Press, Cambridge.
- Graupe, D. (2007). *Principles of Artificial Neural Networks* (2. Baskı). Singapore: World Scientific.
- Gui, G. ve Yang, J. (2020). "Flight Delay Prediction Based on Aviation Big Data and Machine Learning". *Transactions on Vehicular Technology*, 69(1): 140-150.
- Gürsakan, N. (2017). *Makine Öğrenmesi ve Derin Öğrenme*. Dora, Bursa.
- Güvercin, M., Ferhatosmanoglu, N. ve Gedik, B. (2020). Forecasting Flight Delays Using Clustered Models Based on Airport Networks. *IEEE Transactions on Intelligent Transportation Systems*, 1-11.

- Haines, M. M., Stansfeld, S. A., Job, R. F., Berglund, B. ve Head, J. (2001). "Chronic aircraft noise exposure, stress responses, mental health and cognitive performance in school children". *Cambridge University Press*, 31(2): 265-277.
- He, H. ve Garcia, E. (2009). "Learning from imbalanced data". *IEEE Transactions on knowledge and data engineering*, 21(9): 1263-1284.
- He, H. ve Ma, Y. (2013). *Imbalanced Learning*. Wiley-IEEE Press, New Jersey.
- Helme, M. P. (1992). "Reducing air traffic delay in a space-time network". *International Conference on Systems, Man, and Cybernetics*, 18-21 Ekim 1992, Chicago, 236-242.
- Hulse, J. V., Khoshgoftaar, T. M. ve Napolitano, A. (2007). "Experimental Perspectives on Learning from Imbalanced Data". *International Conference on Machine Learning*. 20-24 Temmuz 2007, Oregon, 935-942.
- IATA. (2019). *Airlines Financial Monitor*.
- Ibragimov, B. ve Gusev, G. (2019). "Minimal Variance Sampling in Stochastic Gradient Boosting". *Advances in Neural Information Processing Systems*, 8-14 Aralık 2019, Vancouver, 15061-15071.
- İşler, H. (2018). Makine Elemanları. *Ders Notları*. İzmir: Ege Üniversitesi, Ege Meslek Yüksek okulu.
- Jakubiak, M. (2015). "Environmental impact of air transport - case study of Krakow Airport". *Logistyka*. 5-5 Mart 2015, Kielce, 276-283.
- James, G., Witten, D., Hastie, T. ve Tibshirani, R. (2017). *An Introduction to Statistical Learning with Applications in R*. Springer, New York.
- Jo, T. ve Japkowicz, N. (2004). "Class imbalances versus small disjuncts". *ACM SIGKDD Explorations Newsletter*, 6(1): 40-49.
- Jungai, T. ve Hongjun, X. (2012). "Optimizing Arrival Flight Delay Scheduling Based on Simulated Annealing Algorithm". *Physics Procedia*, 33: 348-353.
- Kantardzic, M. (2003). *Data Mining: Concepts, Models, Methods, and Algorithms*. Wiley-IEEE Press, NJ.
- Kaya, A. ve Kendirli, S. (2018). "Havayolu Şirketlerinde Finansal Bir Araç Olarak Hedging Yönteminin Kullanılması". *MANAS Sosyal Araştırmalar Dergisi*, 7(4): 113-126.
- Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W. ve Liu, T.-Y. (2017). "LightGBM: a highly efficient gradient boosting decision tree". *Advances in neural information processing systems*. 31st Conference on Neural Information Processing Systems. 4-9 Aralık 2017, Long Beach CA, 3146-3154.

- Kim, Y. J., Choi, S., Briceno, S. ve Mavris, D. (2016). "A Deep Learning Approach to Flight Delay Prediction". *IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*. 25-30 Eylül 2016, CA, 1-6.
- Kimnander, M. (2020). *Predicting Profitability of New Customers Using Gradient Boosting Tree Models*. Yüksek Lisans Tezi. University of Skövde, Skövde.
- Kohavi, R. (1995). "A study of cross-validation and bootstrap for accuracy estimation and model selection". *International Joint Conference on AI Palais de Congres*, 20-25 Ağustos 1995 Quebec, 1137-1145.
- Kohavi, R. ve Provost, F. (1998). "Glossary of Terms -Special Issue on Applications of Machine Learning and the Knowledge Discovery Process". *Machine Learning*, 30: 271-274.
- Kubat, M. (2017). *An Introduction to Machine Learning*. Springer, FL.
- Kuhn, N. ve Jamadagni, N. (2017). "Application of Machine Learning Algorithms to Predict Flight Arrival Delays". 1-6.
- Kwok, S. W. ve Carter C. (1990). "Multiple Decision Trees". *Machine Intelligence and Pattern Recognition*, 9: 327-335.
- Lan, S., Clarke, J. ve Barnhart, C. (2006). "Planning for Robust Airline Operations: Optimizing Aircraft Routings and Flight Departure Times to Minimize Passenger Disruptions". *Transportation Science*, 40(1): 15-28.
- Landis, J. ve Koch, G. (1977). "The measurement of observer agreement for categorical data". *Biometrics*, 33(1): 159-174.
- Li, J. (2018). "Monthly housing rent forecast bast on LightGBM". *International Journal of Intelligent Information and Management Science*, 7(6): 58-65.
- Liaw, A. ve Wiener, M. (2002). "Classification And Regression By Random Forest". *R News*, 2(3): 18-22.
- Liu, F., Sun, J., Liu, M., Yang, J. ve Gui, G. (2020). Generalized flight delay prediction method using gradient boosting decision tree. *91st Vehicular Technology Conference*. 25-28 Mayıs 2020, Antwerp, 1-5.
- Loh, W.-Y. (2011). "Classification and regression trees". *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(1): 14-23.
- Mangortey, E. (2019). *Predicting The Occurrence Of Ground Delay Programs and Their Impact on Airport amd Flight Operations*. Yüksek Lisans Tezi. Georgia Institute of Technology, Georgia.

- Mani, I. ve Zhang, J. (2003). "kNN approach to unbalanced data distributions: a case study involving information extraction". *Proceedings of workshop on learning from imbalanced datasets*, 126: 1-7.
- Manna, S., Biswas, S., Kundu, R., Rakshit, S., Gupta, P. ve Barman, S. (2017). "A Statistical Approach to Predict Flight Delay Using Gradient Boosted Decision Tree". *International Conference on Computational Intelligence in Data Science (ICCIDS)*. 2-3 Temmuz 2017, Chennai, 1-5.
- Mantovani, R. G., Horváth, T., Cerri, T., Vanschoren, R. ve Carvalho, A. C. (2016). Hyperparameter tuning of a decision tree induction algorithm. *5th Brazilian Conference on Intelligent Systems (BRACIS)*. 9-12 Ekim 2016, Recife, 37-42.
- Markovic, D., Hauf, T., Röhner, P. ve Spehr, U. (2008). "A statistical study of the weather impact on punctuality at Frankfurt airport". *Meteorological Applications*, 15(2): 293-303.
- McCarthy, N., Karzand, M. ve Lecue, F. (2019). "Amsterdam to Dublin Eventually Delayed? LSTM and Transfer Learning for Predicting Delays of Low Cost Airlines". *The Thirty-First AAAI Conference on Innovative Applications of Artificial Intelligence*. 4-9 Şubat 2019, San Francisco, 9541-9546.
- McCormack, R. L. (1989). "Imperialism, Air Transport and Colonial Development: Kenya, 1920-46". *Journal of Imperial and Commonwealth History*, 17(3): 374-395.
- Meel, P., Singhal, M., Tanwar, M. ve Saini, N. (2020). "Predicting Flight Delays with Error Calculation using Machine Learned Classifiers". *7th International Conference on Signal Processing and Integrated Networks*. 27-28 Şubat 2020, Delhi, 71-76.
- Menardi, G. ve Torelli, N. (2014). Training and assessing classification rules with imbalanced data. *Data Mining and Knowledge Discovery*, 28(1): 92-122.
- Metz, C. E. (1978). "Basic principles of ROC analysis". *Seminars in Nuclear Medicine*, 8(4): 283-298.
- Micci-Barreca, D. (2001). "A preprocessing scheme for high-cardinality categorical attributes in classification and prediction problems". *ACM SIGKDD Explorations Newsletter*. 3(1): 27-32.
- Mogha, G., Ahlawat, K. ve Singh, A. P. (2017). "Performance Analysis of Machine Learning Techniques on Big Data Using Apache Spark". *International Conference on Recent Developments in Science, Engineering and Technology*. 29-30 Temmuz 2017, Qingdao, 17-26.

- Mohamed, H. M., Al-Tabbakh, S. M. ve El-Zahed, H. (2018). "Machine Learning Techniques for analysis of Egyptian Flight Delay". *J. Sci. Res. Sci.*, 35: 390-399.
- Moreira, L., Dantas, C., Oliveira, L., Soares, J. ve Ogasawara, E. (2018). "On Evaluating Data Preprocessing Methods for Machine Learning Models for Flight Delays". *International Joint Conference on Neural Networks (IJCNN)*. 8-13 Temmuz 2018, Rio, 1-8.
- Morgan, J. N. ve Sonquist, J. A. (1963). "Problems in the analysis of survey data, and a proposal". *Journal of the American statistical association*, 58(302): 415-434.
- Morrell, S., Taylor, R. ve Lyle, D. (2008). "A review of health effects of aircraft noise" *Australian and New Zealand Journal of Public Health*, 21(2): 221-236.
- Mustapha, I. B., Shamsuddin, S. M. ve Hasan, S. (2019). "A Preliminary Study on Learning Challenges in Machine Learning-based Flight Delay Prediction". *International Journal of Innovative Computing*, 9(1): 1-5.
- Müller, A. ve Guido, S. (2017). *Introduction to Machine Learning with Python*. O'Reilly, Sebastopol.
- Nilsson, N. (2009). *The Quest for Artificial Intelligence a History of Ideas and Achievements*. Cambridge University Press, Cambridge.
- Nilsson, N. J. (1998). *Artificial intelligence: a new synthesis*. Morgan Kaufmann Publishers, Burlington.
- Oded, M. ve Lior, R. (2005). *Data Mining and Knowledge Discovery Handbook*. Springer, Londra.
- Oğuzlar, A. (2003). "Veri Ön İşleme". *Erciyes Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*. (21): 67-76.
- Oxford Economics. (2019). *How Robots Change the World*. Oxford.
- Oza, S., Sharma, S., Sangoi, h., Raut, R. ve Kotak, V. C. (2015). Flight Delay Prediction System Using Weighted Multiple Linear Regression". *International Journal of Engineering and Computer Science*, 4(4), 11765-11773.
- Pal, M. (2005). "Random forest classifier for remote sensing classification". *International Journal of Remote Sensing*, 26: 217-222.
- Patgiri, R., Hussain, S. ve Nongmeikapam, A. (2020). "Empirical Study on Airline Delay Analysis and Prediction". *EAI Endorsed Transactions on Scalable Information Systems*, 1-8.
- Pepper, C. B., Nascarella, M. A. ve Kendall, R. J. (2003). "A review of the effects to aircraft noise on wildlife and humans, current control mechanisms, and the need for further study". *Environmental Management*, 32(4): 418-432.

- Pinto, N., Doukhan, D. ve DiCarlo, J. J. (2009). "A high-throughput screening approach to discovering good forms of biologically inspired visual representation". *PLoS Comput Biol*, 5(11): 1-12.
- Poole, D., Machworth, A. ve Goebel, R. (1998). *Computational intelligence a logical approach*. Oxford University Press, New York.
- Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. ve Gulin, A. (2019). CatBoost: unbiased boosting with categorical features. 1-23.
- Provost, F. J., Fawcett, T. ve Kohavi, R. (1998). The case against accuracy estimation for comparing induction algorithms. *15th Int. Conf. on Machine Learning*, 1998, San Francisco, 445-453.
- Punmiya, R. ve Choe, S. (2019). "Energy theft detection using gradient boosting theft detector with feature engineering-based preprocessing". *Transactions on Smart Grid*, 10(2): 2326-2329.
- PwC. (2017). *Sizing the prize What's the real value of AI for your business and how can you capitalise?* PwC.
- PwC. (2018). *Will Robots Really Steal Our Jobs?* PwC, UK.
- Ravi, D., Wong, C., Deligianni, F., Berthelot, M., Andreu-Perez, J., Lo, J. ve Yang, G. Z. (2017). "Deep Learning for Health Informatics". *IEEE Journal of Biomedical and Health Informatics*, 21(1): 4-21.
- Rebollo, J. J. ve Balakrishnan, H. (2014). "Characterization and prediction of air traffic delays". *Transportation Research Part C: Emerging Technologies*, 44: 231-241.
- Rich, E. (1983). *Artificial Intelligence*. McGraw-Hill, NY.
- Roiger, R. (2017). *Data Mining A Tutorial-Based*. Addison Wesley, Boston.
- Rokach, L. ve Maimon, O. (2014). *Data Mining with Decision Trees* (2. Baskı). World Scientific Publishing Co. Pte. Ltd, Singapore.
- Russell, S., ve Norvig, P. (2009). *Artificial Intelligence: A Modern Approach*. Prentice Hall, NJ.
- Sahin, E. K. (2020). "Comparative analysis of gradient boosting algorithms for landslide susceptibility mapping". *Geocarto*, 35: 1-25.
- Salford Systems. (2014). *Random forests for beginners*. Salford Systems.
- Salunkhe, U. R. ve Suresh, N. M. (2016). "Classifier Ensemble Design for Imbalanced Data Classification: A Hybrid Approach". *International Conference on Computational Modeling and Security*. 11-13 Şubat 2016, Bengaluru, 725-732.
- Schapire, R. E. (1990). "The strength of weak learnability". *Machine Learning*, 5(2): 197-227.

- Schultz, M., ve Reitmann, S. (2019). "Machine learning approach to predict aircraft boarding". *Transportation Research Part C*, 98: 391-408.
- Sen, R., Kandasamy, K., ve Shakkottai, S. (2018). "Multi-fidelity black-box optimization with hierarchical partitions". *International Conference on Machine Learning*, 10-15 Temmuz 2018, Stockholm, 4538-4547.
- Shannon, C. E. (1948). "A mathematical theory of communication". *The Bell System Technical Journal*, 27(3): 379-423.
- Shepherd, B., Shingal, A. ve Raj, A. (2016). *Value of Air Cargo: Air Transport and Global Value Chains*. IATA.
- Skiena, S. S. (2017). *The Data Science Design Manual*. Springer, NY.
- Smith, B. ve Linden, G. (2017). "Two Decades of Recommender Systems at Amazon.com". *IEEE Internet Computing*, 21(3): 12-18.
- Snoek, J., Larochelle, H. ve Adams, R. P. (2012). "Practical bayesian optimization of machine learning algorithms". *Advances in neural information processing systems*. 3-8 Aralık 2012, Lake tahoe, 2951-2959.
- Suchetana, B., Rajagopalan, B. ve Silverstein, J. (2017). "Assessment of wastewater treatment facility compliance with decreasing ammonia discharge limits using a regression tree model". *Science of the Total Environment*, 598: 249-257.
- Sun, Y., Wang, X. ve Tang, X. (2014). "Deep Learning Face Representation from Predicting 10,000 Classes". *Computer vision and pattern recognition*, 23-28 Temmuz 2014, OH, 1891-1898.
- Sutton, R. S. ve Barto, A. G. (2011). *Reinforcement Learning: An Introduction*. The MIT, London.
- Takeichi, N., Kaida, R., Shimomura, A. ve Yamauchi, T. (2017). "Prediction of Delay due to Air Traffic Control by Machine Learning". *AIAA Modeling and Simulation Technologies Conference*. -13 Ocak 2017, Grapevine, 1-7.
- Tanyıldızı, E. ve Demirtaş, F. (2019). "Hiper Parametre Optimizasyonu". *1st International Informatics and Software Engineering Conference (UBMYK)*. 6-7 Kasım 2019, Ankara, 1-5.
- Thiagarajan, B., Srinivasan, L., Sharma, A. V., Sreekanthan, D. ve Vijayaraghavan, V. (2017). "A Machine Learning Approach for Prediction of On-time Performance of Flights". *36th Digital Avionics Systems Conference (DASC)*. 17-21 Eylül 2017, St. Petersburg, 1-6.



- Transportation Research Board. (2002). *Future Flight A Review of the Small Aircraft Transportation System Concept*. National Academy Press, Washington.
- Tu, Y., Ball, M. O. ve Jank, W. S. (2008). "Estimating flight departure delay distributions—a statistical approach with long-term trend and short-term pattern". *Journal of the American Statistical Association*, 103(481): 112-125.
- Türk, A. M. ve Ozkan, K. (2015). "Pre-Processing Methods for Imbalanced Data Set of Wilted Tree". *ICAINSETE*, 25-28 Ağustos 2015, Vienna, 267-275.
- Van der Aalst, W. M., Rubin, V., Verbeek, H. M., Van Dongen, B. F. ve Günther, C. (2010). "Process mining: a two-step approach to balance between underfitting and overfitting". *Software & Systems Modeling*, 9(1): 87-111.
- Venkatesh, V., Arya, A., Agarwal, P., S, L. ve Balana, S. (2017). "Iterative Machine and Deep Learning Approach for Aviation Delay Prediction". *4th IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics (UPCON)*. 26-28 Kasım 2017, Mathura, 562-567.
- Vermeulen, A. F. (2020). *Industrial Machine Learning Using Artificial Intelligence as a Transformational Disruptor*. Apress, Berkeley.
- Visa, S. ve Salescu, A. (2005). "Issues in Mining Imbalanced Data Sets - A Review Paper". *Proceedings of the sixteen midwest artificial intelligence and cognitive science conference*. 16-17 Nisan 2005, Ohio, 67-73.
- Wang, P. T., Schaefer, L. A. ve Wojcik, L. A. (2003). "Flight connections and their impacts on delay propagation". *Digital Avionics Systems Conference*. 12-16 Kasım 2003, Indianapolis, 5-9.
- Wang, J., Ma, Y., Zhang, L., Gao, R. X., & Wu, D. (2018). "Deep learning for smart manufacturing: Methods and applications". *Journal of Manufacturing Systems*. 48(C):144-156.
- Watts, J. D., Powell, S. L., Lawrence, R. L. ve Hilker, T. (2011). "Improved Classification of Conservation Tillage Adoption Using High Temporal And Synthetic Satellite Imagery". *Remote Sensing of Environment*. 115: 66-75.
- Weiss, G. M. ve Provost, F. (2003). "Learning When Training Data are Costly: The Effect of Class Distribution on Tree Induction". *Journal of Artificial Intelligence Research*, 19: 315-354.
- Witten, I. H. ve Frank, E. (2002). "Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations". *Acm Sigmod Record*, 31(1): 76-88.

- Wong, J. ve Tsai, S. (2012). "A survival model for flight delay propagation". *Journal of Air Transport Management*, 23: 5-11.
- World Bank Group. (2016). *Air Transport*. WBG.
- Wu, J., Toscano-Palmerin, S., Frazier, P. I. ve Wilson, A. G. (2020). "Practical Multi-fidelity Bayesian Optimization for Hyperparameter Tuning". *Uncertainty in Artificial Intelligence Conference*. 4-6 Ağustos 2020, Toronoto, 788-798.
- Xiao, Y., Zhao, Y., Wu, G. ve Jing, Y. (2020). "Study on Delay Propagation Relations Among Airports Based on Transfer Entropy". *IEEE Access*, 8: 39103-37103.
- Xu, N., Sherry, L. ve Laskey, K. B. (2008). "Multifactor Model for Predicting Delays at U.S. Airports". *Transportation Research Record*, 8: 62-71.
- Yalçın, O. (2016). "Havacılık, Hava Gücünün Doğuşu ve Birinci Dünya Savaşına Etkisi". *Ankara Üniversitesi Türk İnkılâp Tarihi Enstitüsü Atatürk Yolu Dergisi*, 59: 181-239.
- Yang, C., Marshall, Z. A., ve Mott, J. H. (2020). "A Novel Integration Platform to Reduce Flight Delays in the National Airspace System". *Systems and Information Engineering Design Symposium (SIEDS)*. 24 Nisan 2020, Zoom, 1-6.
- Yao, Y., Rosasco, L. ve Caponnetto, A. (2007). "On early stopping in gradient descent learning". *Constructive Approximation*, 26(2): 289-315.
- Yılmaz, A. (2017). *Yapay Zeka*. Kodlab, İstanbul.
- Ying, X. (2019). "An Overview of Overfitting and its Solutions". *IOP Conf. Series: Journal of Physics: Conf. Series*, 1168(2): 1-6.
- Yu, B., Guo, Z., Asian, S., Wang, H. ve Chen, G. (2019). "Flight delay prediction for commercial air transport: A deep learning approach". *Transportation Research Part E*, 125: 203-221.
- Yuan, Y. ve Shaw, M. J. (1995). "Induction of fuzzy decision trees". *Fuzzy Sets and systems*, 69(2): 125-139.
- Zadrozny, B., Langford, J. ve Abe, N. (2003). "Cost-sensitive learning by cost-proportionate example weighting". *Data Mining*. 19-22 Kasım 2003, Melbourne, 435-442.
- Zanin, M., Belkoura, S. ve Yanbo, Z. (2017). "Network analysis of Chinese air transport delay propagation". *Chinese Journal of Aeronautics*, 30(2): 491-499.
- Zhang, A., Lipton, Z. C., Li, M. ve Smola, A. J. (2020). *Dive into Deep Learning*. Unpublished draft.
- Zhang, J., Xu, X. H., Wang, F. ve Wei, D. X. (2010). "Airport delay performance evaluation based on fuzzy linear regression model". *Jiaotong Yunshu Gongcheng Xuebao*, 10(4): 109-114.

- Zhang, N. N., Ye, S. Z. ve Chien, T. (2018). "Imbalanced Data Classification Based on Hybrid Methods". *Proceedings of the 2nd International Conference on Big Data Research*. Kasım 2018, Weihai, 16-20.
- Zheng, A. (2015). *Evaluating Machine Learning Models A Beginner's Guide to Key Concepts and Pitfalls*. O'Reilly, Sebastopol.
- Akbulut, U., (2019). "İlk havayolu şirketi: 1914'te Florida'da kuruldu". <https://www.milliyet.com.tr/gundem/ilk-havayolu-sirketi-1914-te-florida-da-kuruldu-2820464> (erişim tarihi: 11.05.2020)
- Ankhaber, (2020). "Hava Kargo İçin 2019 Yılı Son 10 Yılın En Kötü Performanslı Yılı Oldu". <http://www.ankhaber.com.tr/finans/hava-kargo-icin-2019-yili-son-10-yilin-en-kotu-performansli-yili-h57655.html> (erişim tarihi: 11.05.2020)
- BBC, (2014). "Stephen Hawking warns artificial intelligence could end mankind". <http://www.bbc.com/news/technology-30290540> (erişim tarihi: 15.08.2019)
- BBC, (2015). "Gates: İnsanlık yapay zekadan kaygı duymalı". [https://www.bbc.com/turkce/haberler/2015/01/150130\\_gates\\_yapay\\_zeka](https://www.bbc.com/turkce/haberler/2015/01/150130_gates_yapay_zeka) (erişim tarihi: 17.08.2019)
- BBC, (2018). "2017 safest year for air travel as fatalities fall". <https://www.bbc.com/news/business-42538053> (erişim tarihi: 11.06.2020)
- BBC, (2019). "A quick guide to the US-China trade war". <https://www.bbc.com/news/business-45899310> (erişim tarihi: 11.06.2020)
- BBC, (2020a). "Coronavirus: Airlines set for 'worst' year on record". <https://www.bbc.com/news/business-52983756> (erişim tarihi: 12.07.2020)
- BBC, (2020b). "Coronavirus: Trump suspends travel from Europe to US". <https://www.bbc.com/news/world-us-canada-51846923> (erişim tarihi: 11.09.2020)
- Beard, J., (2019). "Global Aviation Challenges & Market Predictions Report 2020". <https://blog.v-hr.com/blog/global-aviation-challenges-market-predictions-report-2020#:~:text=Global%20challenges%20include%20the%20political,wider%20reach%20than%20ever%20before> (erişim tarihi: 11.06.2020)
- Bowen, J. ve Rodrigue, J., (2020). "The Geography of Transport Systems". [https://transportgeography.org/?page\\_id=1765#:~:text=Air%20transportation%20is%20the%20movement,that%20can%20sustain%20controlled%20flight](https://transportgeography.org/?page_id=1765#:~:text=Air%20transportation%20is%20the%20movement,that%20can%20sustain%20controlled%20flight). (erişim tarihi: 11.05.2020)

- Brandom, R., (2016). "Humanity and AI Will Be Inseparable". <https://www.theverge.com/a/verge-2021/humanity-and-ai-will-be-inseparable> (erişim tarihi: 07.03.2020)
- Brownlee, J., (2020). "A Gentle Introduction to the Fbeta-Measure for Machine Learning". [https://machinelearningmastery.com/fbeta-measure-for-machine-learning/#:~:text=The%20F2%2Dmeasure%20is%20calculated,\(4%20\\*%20Precision%20%2B%20Recall\)](https://machinelearningmastery.com/fbeta-measure-for-machine-learning/#:~:text=The%20F2%2Dmeasure%20is%20calculated,(4%20*%20Precision%20%2B%20Recall)) (erişim tarihi: 11.11.2020)
- Burt, J., (2018). "DHL gets logical – and logistical – about machine learning". <https://www.nextplatform.com/2018/04/18/dhl-gets-logical-and-logistical-about-machine-learning/> (erişim tarihi: 13.05.2019)
- Chen, T., (2015). "What is the difference between the R gbm (gradient boosting machine) and XGBoost (extreme gradient boosting)?" <https://www.quora.com/What-is-the-difference-between-the-R-gbm-gradient-boosting-machine-and-XGBoost-extreme-gradient-boosting/answer/Tianqi-Chen-1> (erişim tarihi: 13.10.2020)
- Chen B., (2020). "Early Stopping in Practice: an example with Keras and TensorFlow 2.0". <https://towardsdatascience.com/a-practical-introduction-to-early-stopping-in-machine-learning-550ac88bc8fd> (erişim tarihi: 05.12.2020)
- Cheng, L., (2016). "A gentle introduction to gradient boosting". [http://www.ccs.neu.edu/home/vip/teach/MLcourse/4\\_boosting/slides/gradient\\_boosting.pdf](http://www.ccs.neu.edu/home/vip/teach/MLcourse/4_boosting/slides/gradient_boosting.pdf) (erişim tarihi: 20.05.2020)
- CIRIUM, (2020). "Tracking the in-storage fleet and utilization in a time of uncertainty". <https://www.cirium.com/thoughtcloud/tracking-the-in-storage-fleet-at-a-time-of-uncertainty/> (erişim tarihi: 11.09.2020)
- Ciptadi A. (2019). "What Is Deep Learning, And How Is It Different From Machine Learning?". <https://bluehexagon.ai/blog/what-is-deep-learning-and-how-is-it-different-from-machine-learning/>. (erişim tarihi: 15.05.2019)
- Crawley, J., (2008). "Wasted fuel from U.S. flight delays costs billions". <https://www.reuters.com/article/us-airlines-delays/wasted-fuel-from-u-s-flight-delays-costs-billions-idUSN2250451020080522> (erişim tarihi: 15.06.2020)
- Department for International Trade, (2020). "Existing UK trade agreements with non-EU countries". <https://www.gov.uk/guidance/uk-trade-agreements-with-non-eu-countries> (erişim tarihi: 11.06.2020)
- FAA, (2020). "Flight Delay Information - Air Traffic Control System Command Center". <https://www.fly.faa.gov/flyfaa/usmap.jsp> (erişim tarihi: 12.10.2020)

- Gmelich, K., (2019). "Airbnb Says It Made a Profit Again in 2018". <https://www.bloomberg.com/news/articles/2019-01-15/airbnb-says-it-made-a-profit-again-in-2018-as-ipo-looms-large> (erişim tarihi: 07.03.2020)
- Grover, P., (2017). "*Gradient Boosting from scratch*". <https://medium.com/mlreview/gradient-boosting-from-scratch-1e317ae4587d> (erişim tarihi: 20.05.2020)
- Guru99, (2019) " Machine Learning Tutorial for Beginners: What is, Basics of ML" <https://www.guru99.com/machine-learning-tutorial.html> (erişim tarihi: 15.12.2019)
- Hürriyet, (2020). "*IATA: Hava yolu yolcu trafiği 2020'de Kovid-19 nedeniyle yüzde 66 azalacak*". <https://www.hurriyet.com.tr/ekonomi/iata-hava-yolu-yolcu-trafigi-2020de-kovid-19-nedeniyle-yuzde-66-azalacak-41623773> (erişim tarihi: 11.05.2020)
- IATA, (2020). "*Financial Monitor: Jet fuel prices will remain volatile in 2020*". <https://www.airlines.iata.org/news/financial-monitor-jet-fuel-prices-will-remain-volatile-in-2020> (erişim tarihi: 11.09.2020)
- IATA, (2020). "Industry Losses to Top \$84 Billion in 2020": <https://www.iata.org/en/pressroom/pr/2020-06-09-01/> (erişim tarihi: 11.09.2020)
- Johnson, K., (2018). "*DHL will invest \$300 million to quadruple robots in warehouses in 2019*". <https://venturebeat.com/2018/11/29/dhl-will-invest-300-million-to-quadruple-robots-in-warehouses-in-2019/> (erişim tarihi: 30.11.2020)
- LOJİPORT, (2019). "*Amazon, filosuna 15 uçak daha katacak*". <https://www.lojiport.com/:https://www.lojiport.com/amazon-filosuna-15-ucak-daha-katacak-105228h.htm> (erişim tarihi: 11.05.2020)
- Maslen, R., (2014). "*WORLD ROUTES: Critical Issues in Air Transport Today*". <https://www.routesonline.com/news/29/breaking-news/244463/world-routes-critical-issues-in-air-transport-today/> (erişim tarihi: 06.11.2020)
- Mazareanu, E., (2020a). "*Worldwide air freight traffic from 2004 to 2021*". <https://www.statista.com/statistics/564668/worldwide-air-cargo-traffic/> (erişim tarihi: 05.11.2020)
- Mazareanu, E., (2020b). "*Global air traffic - scheduled passengers 2004-2021*". <https://www.statista.com/statistics/564668/worldwide-air-cargo-traffic/> (erişim tarihi: 11.05.2020)
- Musk, E., (2017). "*Twitter*" <https://twitter.com/elonmusk/status/904638455761612800> (erişim tarihi: 01.06.2019)

- Nahon, A., (2019). "*XGBoost, LightGBM or CatBoost — which boosting algorithm should I use?*" <https://medium.com/riskified-technology/XGBoost-lightgbm-or-catboost-which-boosting-algorithm-should-i-use-e7fda7bb36bc> (erişim tarihi: 25.05.2020)
- Narayanan, S., (2019). "*An Update About Face Recognition on Facebook*". <https://about.fb.com/news/2019/09/update-face-recognition/> (erişim tarihi: 01.07.2020)
- Nvidia, (2016). "*A visual representation of AI, machine learning, and deep learning*". <https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-in-telligence-machine-learning-deep-learning-ai/> (erişim tarihi: 30.11.2020)
- Ozker, S., (2019). "*Boosting algoritmaları nasıl çalışır?*". <https://medium.com/@sertacozker/boosting-algoritmalar%C4%B1-nas%C4%B1-%C3%A7al%C4%B1%C5%9F%C4%B1r-edac1174e971> (erişim tarihi: 20.05.2020)
- Peart, A., (2020). "*Homage to John McCarthy, the Father of Artificial Intelligence (AI)*". <https://www.artificial-solutions.com/blog/homage-to-john-mccarthy-the-father-of-artificial-intelligence> (erişim tarihi: 12.12.2020)
- Reed, T., (2020). "*These Are The Top Six Issues Facing The Revamped Airline Industry*". <https://www.forbes.com/sites/tedreed/2020/05/18/top-six-trends-as-crisis-reshapes-the-airline-industry/?sh=2c3afb9b5658> (erişim tarihi: 06.11.2020)
- Singh S., (2018). "*Understanding the Bias-Variance Tradeoff*". <https://towardsdatascience.com/understanding-the-bias-variance-tradeoff-165e6942b229> (erişim tarihi: 03.06.2020)
- Wong, S., (2016). "*Google Translate AI invents its own language to translate with*". <https://www.newscientist.com/article/2114748-google-translate-ai-invents-its-own-language-to-translate-with/> (erişim tarihi: 01.07.2020)
- World Bank, (2018). "*Air transport, passengers carried*". <https://data.worldbank.org/https://data.worldbank.org/indicator/IS.AIR.PSGR?end=2018&start=2004> (erişim tarihi: 11.05.2020)
- XGBoost, (2020). "*XGBoost Documentation*". <https://XGBoost.readthedocs.io/en/latest/> (erişim tarihi: 13.10.2020)
- Yandex, (2017). "*CatBoost is a high-performance open source library for gradient boosting on decision trees*". <https://catboost.ai/docs/concepts/about.html> (erişim tarihi: 05.02.2020)
- Yandex, (2017). "*Feature importance*". [https://catboost.ai/docs/concepts/fstr.html#fstr\\_\\_regular-feature-importance](https://catboost.ai/docs/concepts/fstr.html#fstr__regular-feature-importance) (erişim tarihi: 3.06.2020)

- Yandex, (2019). "*Missing values processing*". <https://catboost.ai/docs/concepts/algorithm-missing-values-processing.html> (erişim tarihi: 03.06.2020)
- Zhang, B., (2018). "*Airlines are making more money than ever — but they're facing a mountain of problems*". <https://www.businessinsider.com/airlines-biggest-business-problems-2018-4> (erişim tarihi: 06.11.2020)

## ÖZGEÇMİŞ

<b>Adı ve SOYADI</b>	Irmak DALDIR
<b>EĞİTİM DURUMU</b>	
<b>Mezun Olduğu Lise</b>	Bornova Anadolu Lisesi - Almanca
<b>Lisans Diploması</b>	İzmir Ekonomi Üniversitesi – Lojistik Yönetimi İzmir Ekonomi Üniversitesi – Uluslararası Ticaret ve Finans
<b>Yüksek Lisans Diploması</b>	Celal Bayar Üniversitesi - İşletme
<b>Tez/ Dönem Projesi Konusu</b>	Kalite fonksiyon göçerimi yaklaşımı ile müşteri memnuniyetinin analizi: Üçüncü parti lojistik hizmet sağlayıcılarında deneysel bir uygulama
<b>Doktora Diploması</b>	Akdeniz Üniversitesi - İşletme
<b>Tez/Dönem Projesi Konusu</b>	Havacılık Sektöründe Uçuş Gecikmelerinin Makine Öğrenmesi Temelli Analizi ve Tahmini
<b>Yabancı Dil / Diller</b>	İngilizce, Almanca
<b>BİLİMSEL FAALİYETLER</b>	
<p>DALDIR I., SOFYALIOĞLU Ç., (2019). "Üçüncü Parti Lojistik Sektöründe Depolama Faaliyetlerinin Stratejik Yönetimi: AHS ve KFG Yöntemlerinin Klasik ve Bulanık Yaklaşımlar Açısından Karşılaştırılması". <i>İşletme Araştırmaları Dergisi</i>, 11:917-936.</p> <p>DALDIR I., TOSUN Ö., (2018). "Bulanık WASPAS ile Yeşil Tedarikçi Seçimi". <i>Uludağ Üniversitesi Mühendislik Fakültesi Dergisi</i>,4: 193-208.</p> <p>DALDIR I., TOSUN Ö., (2017) "Comparison of the Port Authority's Efficiency in Turkey". <i>Journal of Management, Marketing and Logistics</i>, 4: 152-158.</p>	
<b>İŞ DENEYİMİ</b>	
<b>Çalıştığı Kurumlar</b>	Akdeniz Üniversitesi- Uygulamalı Bilimler Fakültesi