



**İSTANBUL TOPKAPI ÜNİVERSİTESİ**  
**MÜHENDİSLİK FAKÜLTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ**

**-VERİ MADENCİLİĞİ (1.DÖNEM)-**  
**UÇUŞ GECİKME TAHMİNİ PROJE RAPORU**  
**-GRUP ADI: TÜRBÜLANS (TURBULENCE)-**

**Öğrenci Adı : EMİNE GÜNEŞ**

**Öğrenci No : 22040101036**

**E-mail : eminegunes@stu.topkapi.edu.tr**

**Öğrenci Adı : OĞUZHAN ÖZDEMİR**

**Öğrenci No : 22040101017**

**E-mail : oguzhanozdemir2@stu.topkapi.edu.tr**

**Öğrenci Adı : MUHAMMET ENES İNAL**

**Öğrenci No : 22040101023**

**E-mail : muhammetenesinal@stu.topkapi.edu.tr**

**Öğrenci Adı : MUHAMMED MERT ORUÇ**

**Öğrenci No : 22040101035**

**E-mail : muhammedmertoruc@stu.topkapi.edu.tr**

**Öğrenci Adı : ZEKERİYA DENİZ UĞURLU**

**Öğrenci No : 22040101034**

**E-mail: zekeriyaadenizugurlu@stu.topkapi.edu.tr**

**GİTHUB REPO LİNK: <https://github.com/Oguzhan1511/Turbulence>**

## 1) PROBLEM TANIMI

**İş/Bilimsel Soru:** Projemiz, Amerika'daki 2015 yılına ait mevcut uçuş verileri kullanarak, bir uçuşun varış noktasına planlanan saatten 15 dakikadan fazla gecikmeli olarak varıp varamayacağını önceden yüksek doğrulukla tahmin edebilir mi? sorusuna cevap aramaktadır. Uçuş sektöründe rötarlar hem uçuş şirketlerini finansal olarak zarara sokup hem de yolcu memnuniyetsizliğine yol açmaktadır. Bu proje, uçuşların gerçekleşmeden önce gecikme olup olmayacağını tahmin ederek havayolu şirketleri için planlama ve yolcular için bilgilendirme sağlamayı amaçlamaktadır.

**Görev Türü :** İkili Sınıflandırma (Binary Classification)

**Hedef Değişken :** 'IS\_DELAYED' Uçuşun 15 dakikadan fazla geç kalıp kalmadığını gösteren ikili (0 veya 1) değişkendir. '0 (negatif)' zamanında veya erken varış ( 'ARRIVAL\_DELAY' <= 15 dakika). '1 (pozitif)' gecikmeli varış ('ARRIVAL\_DELAY' > 15 dakika).

**Başarı Kriterleri :** 'ROC AUC >= 0.70' = Modelin sınıfları ayırt etme yeteneği

'Doğruluk (Accuracy) >= %80 '

'F1-Score >= 0.20' = Gecikmeli uçuşların doğru tespiti ile yanlış tespiti arasındaki dengeyi ölçme

## 2) PROJE YÖNETİMİ

**1.BÖLÜM (21-30 Ekim) :** Literatür taraması, Kaggle 'Flight Delays' veri setinin seçimi ve proje konusunun netleştirilmesi.

**2.BÖLÜM (1-10 Kasım) :** Verilerin tanınması (EDA) ve görselleştirmesi. Eksik verilerin analizi. Yüksek miktardaki 'null' verilen olduğu sütunların tespit edilmesi ve veri temizleme yöntemlerinin belirlenmesi.

**3.BÖLÜM (10-17 Kasım) :** Veri ön işleme (Data Preprocessing) pipeline'ının kurulması. Kategorik değişkenler, havalimanları için Target Encoding ve havayolları için One-Hot Encoding işlemlerinin uygulanması.

**4.BÖLÜM (17-30 Kasım) :** Basit 'Baseline' modellerin (Dummy, Decision Tree, Gaussian Naive Bayes vs.) kurulması. Basit modeller üzerinde denenmek üzere bazı feature selection ve dimension reduction yöntemlerinin uygulanması.

**5.BÖLÜM (1-30 Aralık) :** Gelişmiş modellerin planlanması ve uygulanması. Gelişmiş modellerden en iyi sonuçların alınabilmesi için feature selection ve dimension reduction yöntemlerinin uygulanması. Hiper-parametre optimizasyonlarının planlanması ve uygulanması.

**6.BÖLÜM (1-7 Ocak) :** Gelişmiş modellerden çıkarılan sonuçların performans analizlerinin yapılması. Modellerin tanımlanan performans metriklerine göre karşılaştırılması. Confusion Matrix analizi ile hata türlerinin incelenmesi.

**7.BÖLÜM (7-16 Ocak) :** Final proje raporunun derlenmesi ve sunum hazırlaması.

## **Roller ve Sorumluluklar:**

**Zekeriya Deniz Uğurlu (Veri Entegrasyonu ve Ön İşleme) :** 'flights.csv', 'airlines.csv' ve 'airports.csv' veri setlerinin birleştirilmesi. Eksik veri analizi ve temizleme stratejilerinin yapılması. Encoding işlemlerinin uygulanması. 'preprocess\_pipeline' ve 'encode\_features' fonksiyonlarının geliştirilmesi.

**Emine Güneş (Veri Analizi) :** Verilerin incelenmesi. Veri setindeki kategorik ve sayısal değişkenlerin görselleştirilmesi. Histogramlar ve dağılım grafiklerinin oluşturulması. Sayısal değişkenler ile hedef değişken arasındaki korelasyon matrisinin çıkarılması.

**Muhammet Enes İnal (Temel Base Model İşlemleri) :** Projenin başarısını ölçmek için bir referans noktası oluşturulması. Karmaşık olmayan, hızlı çalışan ve yorumlanabilir modellerin kurulması ve eğitilmesi. Bu modellerin ham veri üzerindeki ilk Accuracy ve ROC-AUC gibi skorlarının raporlanması.

**Muhammed Mert Oruç (Özellik Seçimi İşlemleri) :** Yüksek boyutlu veri setinde, oluşturulan base modellere en çok katkı sağlayan niteliklerin istatistiksel yöntemlerle belirlenmesi ve gürültülü verilerin ayıklanması. İşlemlerden sonra baseline model ile test edilip performans etkilerinin test edilmesi.

**Oğuzhan Özdemir (Boyut İndirgeme İşlemleri) :** Özellikle encoding işlemleri sonrası artan sütun sayılarının yönetilmesi ve hesaplama maliyetlerinin düşürülmesi. Belirlenen boyut indirgeme yöntemleri ile veri setinin varyansını koruyan daha az sayıda bileşene indirgemesi. İndirgenmiş veri seti ile Baseline modellerinin tekrar test edilerek işlem süresi ve başarı farkının analiz edilmesi.

## **Çıktılar:**

**GitHub Repo Link:** <https://github.com/Oguzhan1511/Turbulence>

**Final Proje Raporu:** FET445\_[OgrenciNumarası]\_[Turbulence]\_FinalReport.pdf

**Kod Dosyaları:** FET445\_[22040101034]\_[Turbulence]\_1.ipynb

FET445\_[22040101035]\_[Turbulence]\_2.ipynb

FET445\_[22040101036]\_[Turbulence]\_3.ipynb

FET445\_[22040101017]\_[Turbulence]\_4.ipynb

FET445\_[22040101023]\_[Turbulence]\_5.ipynb

**Veri Seti:** 'flights.csv', 'airlines.csv' ve 'airports.csv' tabloları.

Veri seti linki (Kaggle) : <https://www.kaggle.com/datasets/usdot/flight-delays>

**Sunum:** Proje bulgularını özetleyen sunum dosyası (.pptx)

### 3) İlgili Çalışmalar

Abhishek Banerjee tarafından yapılan çalışma uçuş gecikmesi tahmini odaklıdır. Bizim projemizde olduğu gibi uçuşun geç kalıp kalmayacağını tahmin etmeye çalışmıştır ama bizim projemizden farkı bu çalışmadan uçağın kalkış noktasından geç kalkmayı tahmin etmektedir. Linear regression, ridge, random forest, decision tree, boosted linear modellerini kullanmıştır ve hepsinin doğru tahmin oranlarını karşılaştırmıştır. Veri setindeki tüm verileri kullanmak yerine bazı havalimanlarının uçuşlarını kullanmıştır.

Link:

<https://www.kaggle.com/code/abhishek211119/2015-flight-delays-and-cancellation-prediction>

Manasi Chhibber tarafından yapılan çalışmada yine uçağın geç kalıp kalmayacağı hesaplanmıştır. Tek model olarak Decision Tree kullanılmıştır ve kodda doğruluk oranını %99.8 olarak gösterilmiştir. Bizden farklı olarak Training ve Testing verilerini 70 30 olarak ayırmıştır. Data leakage engellemek için ekstra bir şey yapmamıştır.

Link: <https://www.kaggle.com/code/manasichhibber/flight-delay-predictions>.

Levaniz tarafından yapılan çalışma diğer iki çalışmaya göre daha detaylıdır. Model olarak random forest kullanılmıştır. Date leakage önlenmesi için gereken kolonları silmiştir. Doğruluk (accuracy) metriğini kullanarak verileri karşılaştırmıştır. Veri kullanımını azaltmak için bazı yerlerde memory kısıtlamıştır.

Link: <https://www.kaggle.com/code/levaniz/machine-learning-analysis-of-flights-data>.

Bizim çalışmamızda ise hepsinin eksiklikleri bulunarak en iyi hale getirilmeye çalışılmıştır. Öncelikle data leakage engellenerek modelin gerçek sonuçlar verecek şekilde öğrenmesi sağlanmıştır. Farklı base modeller denenmiş ve en iyi doğruluk oranına sahip modeli seçilmiştir. Veri setinin bazı bölümleri değil hepsi kullanılarak hiçbir hafıza kısıtlanmamıştır. Aynı veri seti üzerinden yapılmış diğer çalışmalara göre farkımız feature selection, feature engineering, dimension reduction kullanmamızdır. Aynı modellerde farklı parametreler kullanılarak yine en iyi sonucu verecek parametreler bulunmuştur.

### 4)Veri Tanıtımı ve Yönetimi

#### 1-Veri Seti

**Adı:** 2015 Uçuş Gecikmeleri ve İptalleri (2015 Flight Delays and Cancellations)

**Kaynak:** Kaggle

**Bağlantı:** <https://www.kaggle.com/datasets/usdot/flight-delays>

**Lisans/Kullanım Hakları:** Bu veriler ABD Hükümeti tarafından üretilmiştir. Bu yüzden telif hakları kısıtlaması yoktur akademik ve ticari kullanılabilir.

## 2-Veri Şeması

Veri seti 3 ana dosyadan oluşmaktadır ve birbirleriyle ilişkilidir.

### 1. flights.csv ( Ana Veri Tablo)

#### Değişkenler:

- Zaman Bilgileri: YEAR, MONTH, DAY, DAY\_OF\_WEEK (Tam sayı- Integer)
- Uçuş Bilgileri: AIRLINE(String), FLIGHT\_NUMBER(Integer), TAIL\_NUMBER(String)
- Lokasyon: ORIGIN\_AIRPORT, DESTINATION\_AIRPORT(String)
- Zamanlama: SCHEDULED\_DEPARTURE, DEPARTURE\_TIME, DEPARTURE\_DELAY(Float-Int)
- Hedef Değişken: ARRIVAL\_DELAY >15 dakika ise "1" (Gecikti) değilse "0" (gecikmedi) anlamına gelir.
- İptal Durumu: CANCELLED (0 veya 1 Binary)
- Birimler: Zamanlar HHMM formatında, gecikmeler dakika cinsinden
- Beklenen Aralıklar: Gecikmeler negatif(erken varış) olabilir veya pozitif (geç kalma) olabilir.

### 2. airlines.csv (Havayolu Tablosu)

IATA\_CODE: Havayolunun 2 haneli kod kısaltmasıdır( Örneğin; AA;AS...)

AIRLINE: Havayolunun tam adıdır(Örneğin; American Airlines)

### 3 .airports.csv (Havaalanı Tablosu)

IATA\_CODE: 3 haneli havaalanı kodu(Örneğin; AJX,LAX)

AIRPORT,CITY,COUNTRY,STATE: Lokasyon bilgileri (String)

LATITUDE, LONGITUDE: Coğrafi konumlar (Float-Harita görselleştirmesi için)

## 3- Boyut(Size ve Scale)

- Satırlar(Rows): flights.csv dosyasında yaklaşık 5.8 milyon satır vardır.
- Sütunlar(Columns): flights.csv dosyasında 31 adet sütun vardır.
- Beklenen Sınıf Dengesi(Class Balance): Veri seti dengesizdir

Uçuşlar çoğunlukla zamanında kalkar(%80)

Gecikme yaşayan uçuşlar %20 dağılımındadır

#### 4-Veri Eriřim Planı(Data Access Plan)

Elde Etme: Veriler kaggle platformundan elde edilecektir.

Depolama: Proje geliştirme aşamasında veriler diskte depolanacaktır.

Güncelleme: Veri grubu 2015 yılına aittir herhangi bir aktif veri akışı olmadığı için güncellemeye ihtiyaç yoktur.

#### 5-Etik, Gizlilik ve Önyargı

-Gizlilik(Privacy): Veri seti anonimdir. Etik açıdan kullanımı güvenlidir.

-Rıza: ABD Ulaştırma Bakanlığı'nın yayınladığı OpenData olduğu için bireysel rızaya gerek yoktur.

### 5) Keşifsel Veri Analizi (Exploratory Data Analysis)

**Veri Kalitesi Kontroller :** Veri setinin güvenilirliğini sağlamak amacıyla temizlik işlemleri uygulanmıştır. İlk olarak 'CANCELLATION\_REASON', 'AIR\_SYSTEM\_DELAY' ve 'WEATHER\_DELAY' gibi sütunların veri tanıma (EDA) işlemlerinden sonra %80'in üzerinde eksik veri içerdiği tespit edilmiş ve bu sütunlar veri setinden temizlenmiştir.

'ARRIVAL\_DELAY', 'DEPARTURE\_DELAY', 'ACTUAL\_ELAPSED\_TIME', 'WHEELS\_OFF' ve 'TAXI\_IN/OUT' gibi değişkenler sızıntı riski (data\_leakage) taşıdığı gerekçesiyle veri setinden temizlenmiştir.

Ayrıca, iptal edilen (CANCELLED = 1) ve yönlendirilen (DIVERTED = 1) gibi uçuşlar filtrelenerek veri tutarlılığı sağlanmıştır.

**Dağılımlar ve Denge :** Hedef değişken analizi sonucunda, veri setinde sınıf dengesizliği olduğu görülmüştür. Gecikmeyen uçuşların sayısı geciken uçuşlara kıyasla oldukça fazladır. Sayısal değişkenlerin 'DISTANCE' ve 'SCHEDULED\_TIME' özelliklerinin sağa çarpık bir dağılım sergilediği histogramlar aracılığıyla gözlemlenmiştir.

**Özellik-Hedef İlişkileri :** Sayısal değişkenler arasındaki ilişkileri anlamak için kategorik değişkenler(Havayolu ve Havalimanı) ile hedef değişken arasındaki ilişkiyi modele yansıtabilmek için, bu değişkenlerin ortalama gecikme oranlarına göre kodlanması yani Target Encoding stratejisi benimsenmiştir. 'SCHEDULED\_DEPARTURE' gibi zaman bilgileri, günün bölümlerine (Sabah, öğle, akşam vs.) ayrılarak kategorik hale getirilmiş ve hedef üzerindeki etkisi daha belirgin hale getirilmiştir.

**Görselleştirme Planı :** Verinin özelliklerini ortaya koymak için aşağıdaki grafikler kullanılmıştır:

**Countplot :** Hedef değişkenin(IS\_DELAYED) sınıf dağılımını ve dengesizliğini göstermek için.

**Histogram & KDE:** Uçuş mesafesi ve sürelerinin dağılımını ve çarpıklığını analiz etmek için.

**Heatmap (Isı Haritası) :** Değişkenler arası korelasyon matrisini görselleştirmek ve çoklu bağlantı riskini değerlendirmek için.

## 6) Veri Hazırlama Planı

**Temizleme** : Analiz sonucunda tespit edilen veri sızıntısı kaynakları ('DEPARTURE\_TIME' , 'ARRIVAL\_DELAY' , 'TAXI\_IN' , 'TAXI\_OUT' vs.) eğitim setinden temizlenmiştir. Gürültü yaratan ve yüksek kardinaliteli 'FLIGHT\_NUMBER' ve 'TAIL\_NUMBER' gibi sütunlar silinmiştir. Analiz odağını sadece gerçekleşen uçuşlar olarak daraltmak adına 'CANCELLED' ve 'DIVERTED' gibi gerçekleşmeyen veya yönlendirilen uçuş kayıtları veri setinden temizlenmiştir.

**Imputasyon Stratejisi** : Eğitim sırasında mevcut olan fakat test setinde görülmeyen havalimanı kodları için Target Encoding işlemi yaparken 'Global Ortalama (mean)' değeri atanarak, modelin yeni verilere karşı hata üretmesi veya yanlış tahmin yapmasının önüne geçilmiştir.

**Dönüşümler** : 300' den fazla benzersiz değeri olan 'ORIGIN\_AIRPORT' ve 'DESTINATION\_AIRPORT' değişkenleri için, her havalimanının 'ortalama gecikme oranı' hesaplanarak sayısal bir değere dönüştürülmüştür. Bu sayede yüzlerce yeni sütun oluşturulmadan kategorik bilgi korunmuştur.

'AIRLINE' (14 sınıf) ve 'TIME\_OF\_DAY' (5 sınıf) gibi düşük kardinalite değişkenler için One-Hot Encoding uygulanarak modelin bu kategoriler arasındaki farkı daha iyi öğrenmesi sağlanmıştır.

**Özellik Mühendisliği**: Ham olan 'SCHEDULED\_DEPARTURE' verisi, sayısal bir değerden anlamlı zaman dilimlerine dönüştürülmüştür. Saat bilgisi belirli aralıklar kullanılarak 'Sabah', 'Öğle' , 'Akşam', 'Gece' ve 'Gece Yarısı' olmak üzere 5 farklı kategoriye ('TIME\_OF\_DAY') ayrılmış ve modelin zaman etkenini daha iyi öğrenmesi sağlanmıştır.

'flights.csv' tablosu, 'airlines.csv' ve 'airports.csv' tabloları ile birleştirilerek, sadece kodlar yerine havayolu ve havalimanı karakteristiklerinin de analize dahil edilmesi sağlanmıştır.

**Özellik Seçimi ve Boyut İndirgeme** : Değişkenler arasındaki çoklu bağlantıyı çözmek için Korelasyon Matrisi kullanılmış ve hedef değişkenle (IS\_DELAYED) ilişkisi en düşük olan öznelilikler belirlenmiştir.

Özellikle One-Hot Encoding sonrası artan sütun sayısını yönetmek ve hesaplama maliyetini düşürmek için, veri setindeki varyansın büyük kısmını koruyan PCA yöntemi uygulanarak veri daha düşük boyutlu bir uzaya taşınmıştır.

## 7) Modelleme Planı

**Muhammed Mert Oruç**: Dummy Classifier : Modelin hiçbir şey öğrenmeden sadece çoğunluk sınıfını tahmin ederek minimum başarı alt sınırını belirlemek için kullanılmıştır.

Ridge Classifier : One-Hot Encoding sonrası artan sütun sayısı ve çoklu bağlantı riskiyle başa çıkmak için doğrusal model olarak seçilmiştir.

**Zekeriya Deniz Uğurlu**: Decision Tree : Veri setindeki doğrusal olmayan ilişkileri yakalayabilen basit bir yapı sunduğu için tercih edilmiştir.

Linear SVC : Büyük ve seyrek veri setlerinde, karmaşık hesaplamalara girmeden hızlı ve etkili bir ayırım düzlemi çizbildiği için seçilmiştir.

**Oğuzhan Özdemir:** Logistic Regression: İkili sınıflandırma problemlerinde endüstri standardı olması ve her bir sütunun üzerindeki etkisini olasılıksal olarak yorumlar.

Gaussian Naive Bayes: Özelliklerin birbirinden bağımsız olduğu varsayımıyla çalışan, eğitim süresi çok kısa olan ve büyük veri setlerinde hızlı olan modeldir.

**Emine Güneş:** Dummy Classifier: Sınıf dengesizliğinin model başarısını yanıltıp yanıltmadığını kontrol etmek ve rastgele tahmin başarısını raporlamak için eklenmiştir.

Gaussian Naive Bayes: Hesaplama maliyeti düşük olduğu için ve karmaşık modellerden önce verinin genel dağılımına dayalı basit tahminleme performansını görüntülemek amacıyla seçilmiştir.

**Muhammet Enes İnal:** Logistic Regression: Modelin çıktılarını (0 veya 1) olasılık değeri olarak verebilmesi ve basit yapısı nedeniyle base model olarak seçilmiştir.

Decision Tree: Veri ön işleme gerektirmemesi ve özneliliklerin hiyerarşik önemini basitçe ortaya koyduğu için seçilmiştir.

### **Aday Modeller:**

**Oğuzhan Özdemir:** LightGBM : 5 milyon satırlık büyük veri setimizde, yaprak odaklı büyüme stratejisi sayesinde diğer boosting algoritmalarına nazaran hızlı eğitildiği ve yüksek başarı sağlayacağı, Base model olarak seçilen Logistic Regression'un aksine doğrusal olmayan örüntüleri de yakalayabileceği düşünüldüğü için aday model olarak seçilmiştir.

Extra Trees Classifier : RandomForest'a kıyasla bölünme noktalarını rastgele seçerek modelin aşırı öğrenme riskini daha fazla düşürdüğü ve parazitli verilerde daha sağlam çalışacağı düşünüldüğü için seçilmiştir.

**Zekeriya Deniz Uğurlu:** Random Forest : Çok sayıda karar ağacını birleştirerek, hataları minimize etmesi ve hem kategorik hem de sayısal verilerle güçlü performans vermesi düşünüldüğü için seçilmiştir. DecisionTree Classiffier modelinin en büyük sorunu olan yüksek varyans ve ezberleme problemini çözeceği için aday olarak görülmüştür.

SVC(RBF Kernel ile) : Veriyi daha yüksek boyutlu bir uzaya taşıyarak, doğrusal olarak ayrılamayan karmaşık sınıf sınırları tespit etme yeteneği nedeniyle ve Linear SVC base modelin sadece düz bir çizgi ile ayırabildiği sınıfları, RBF çekirdeği ile eğrisel ve çok boyutlu sınırlar çizerek ayırır. Bu sayede Linear SVC'den çok daha üstün ayırım gücüne sahiptir. Bu yüzden aday olarak seçilmiştir.

**Emine Güneş:** Naive Bayes: Hiper-parametre optimizasyonu ile basit bir modelin maksimum kapasitesini test edebileceği için seçilmiştir. Base model olarak seçilen DummyClassifier ve Gaussian NB'nin varsayılan ayarlarından daha hassas bir olasılık hesabı yapmaktadır.

CatBoost: Özellikle "Airline" ve "Airport" gibi kategorik değişkenleri One-Hot Encoding işlemine tabi tutmadan kendi içinde işleyebilmesi ve veri setimizin büyük problemi olan sınıf dengesizliğine karşı dirençli olması sebebiyle en güçlü adaylardan biridir.

**Muhammed Mert Oruç:** QDA : Sınıfların kovaryans matrislerini ayrı ayrı öğrenerek doğrusal olmayan karar sınırları çizebilmesi, onu basit linear modellerden ayıran en önemli özelliğidir. Ayrıca RidgeClassifier base modeli doğrusal bir model olduğundan dolayı veriyi düz bir çizgiyle ayırmaya çalışır. QDA ise verinin dağılımına göre eğrisel sınırlar çizer ve eksik öğrenme riskini ortadan kaldırır. Bu sebeple aday olarak kullanılabilir.

ADA Boost: Önceki modellerin yanlış sınıflandırdığı zor uçuş örneklerine daha fazla ağırlık vererek, modelin zayıf yönlerini iteratif olarak güçlendirdiği için aday olarak seçilmiştir.

**Muhammed Enes İnal:** XGBoost: İçerdiği regülarizasyon terimleri ile aşırı öğrenmeyi engellemesi nedeniyle aday olarak düşünülmüştür. Decision Tree modelinin aksine, ağaçları bağımsız değil sıralı kurar ve hatayı optimize ederek ilerler. Logistic Regression'a göre çok daha esnek bir yapı sunduğu için ana model olarak seçilebilir.

Stacking Classifier: Tek bir modelin başarısına güvenmek yerine, XGBoost, RandomForest ve KNN gibi farklı modellerin tahminlerini birleştirip genelleştirme yeteneği en yüksek modeli oluşturmak için seçilmiştir.

## 8) Değerlendirme Tasarımı

**Kullanılan Metrikler:** Ekibimiz geliştirdiği tüm modellerin performansını karşılaştırmak için birden fazla metrik kullanmıştır. F1, Precision, Recall, ROC-AUC, PR-AUC, MAE gibi karşılaştırma metrikleri kullanılmıştır. Bu metrikler 6 base modelde denenerek karşılaştırılmıştır.

### Zekeriya Deniz Uğurlu (PROJE-1):

<u>Model</u>	<u>Accuracy</u>	<u>F1 Score</u>	<u>Precision</u>	<u>Recall</u>	<u>ROC AUC</u>	<u>PR AUC</u>	<u>MAE (Hata)</u>
Decision Tree (Normal)	60.45	36.88	0.26	0.65	66.48	30.12	0.40
Decision Tree (RFE)	61.26	36.86	0.26	0.63	66.50	30.05	0.39
Decision Tree (PCA)	60.34	35.82	0.25	0.62	65.06	27.61	0.40
Linear SVM (Normal)	58.82	35.12	0.24	0.62	63.56	26.07	0.41
Linear SVM (PCA)	58.10	34.70	0.24	0.62	62.93	25.61	0.42
Linear SVM (RFE)	58.59	34.82	0.24	0.62	63.11	25.45	0.41

**Emine Güneş (PROJE-3):**

Model	accuracy	F1 Score	precision	recall	ROC AUC	PR AUC	MAE
<b>Dummy Classifier</b>	82.09	0.00	0.00	0.00	50.00	17.91	17.91
<b>Dummy Classifier +CA</b>	82.09	0.00	0.00	0.00	50.00	17.91	17.91
<b>Dummy Classifier +LDA</b>	82.09	0.00	0.00	0.00	50.00	17.91	17.91
<b>Gaussian NB</b>	72.87	0.25	0.25	0.26	61.42	24.07	33.84
<b>Gaussian NB +CA</b>	72.36	0.26	0.25	0.28	61.48	24.12	33.90
<b>Gaussian NB+LDA</b>	82.09	0.00	0.00	0.00	63.55	26.12	28.44

**Muhammed Mert Oruç (PROJE-2):**

Model	Accuracy	F1 Score	Precision	Recall	ROC AUC	PR AUC	MAE
<b>Dummy Classifier</b>	70.59%	0.18	0.18	0.18	49.98%	17.90%	29.41%
<b>Dummy Classifier+ANOVA F</b>	70.59%	0.18	0.18	0.18	49.98%	17.90%	29.41%
<b>Dummy Classifier +SVD</b>	70.59%	0.18	0.18	0.18	49.98%	17.90%	29.41%
<b>Ringe Classifier</b>	58.82%	0.35	0.24	0.62	63.56%	26.07%	41.18%
<b>Ringe Classifier+ANOVA F</b>	58.60%	0.35	0.24	0.62	63.30%	25.79%	41.40%
<b>Ringe Classifier + SVD</b>	58.09%	0.35	0.24	0.62	62.94%	25.63%	41.91%

**Oğuzhan Özdemir (PROJE-4):**

<b>Model</b>	<b>Accuracy</b>	<b>F1 Score</b>	<b>Precision</b>	<b>Recall</b>	<b>ROC AUC</b>	<b>PR AUC</b>	<b>MAE</b>
<b>Logistic Regrasyon</b>	%82.09	0.00	0.00	0.00	0.60	0.23	0.18
<b>Logistic Regrasyon + Mutual Information</b>	%82.09	0.00	0.00	0.00	0.64	0.25	0.18
<b>Logistic Regrasyon + MU + LDA</b>	%82.09	0.00	0.00	0.00	0.65	0.26	0.18
<b>Naive Bayes</b>	%75.04	0.25	0.25	0.25	0.51	0.18	0.21
<b>Naive Bayes + Mu</b>	%79.05	0.25	0.26	0.28	.053	0.21	0.21
<b>Naive Bayes + MU + LDA</b>	%82.09	0.25	0.26	0.28	0.54	0.23	0.21

**Muhammet Enes İnal (PROJE-5):**

<b>Model</b>	<b>Accur acy</b>	<b>F1 Score</b>	<b>Precisi on</b>	<b>Recal l</b>	<b>ROC AUC</b>	<b>PR AUC</b>	<b>MAE (Hata)</b>
<b>Logistic Reg.</b>	58.95	35.12	0.2449	62.04	63.57	26.07	0.4105
<b>Logistic Reg. (SelectFromModel)</b>	59.00	35.08	0.2448	61.85	63.48	25.96	0.4100
<b>Logistic Reg. (TruncatedSVD)</b>	58.28	34.70	0.2411	61.91	62.93	25.62	0.4172
<b>Logistic Reg. (Sel.FromModel + SVD)</b>	58.90	35.04	0.2444	61.91	63.43	25.90	0.4110
<b>Decision Tree</b>	66.89	39.97	0.2959	61.57	68.81	34.34	0.3311
<b>Decision Tree (SelectKBest - ANOVA F)</b>	60.64	36.85	0.2585	64.12	66.51	30.05	0.3936
<b>Decision Tree (FastICA)</b>	66.13	39.07	0.2882	60.65	67.16	31.08	0.3387
<b>Decision Tree (SelectKBest ANOVA F+ FastICA)</b>	60.39	36.71	0.2571	64.15	66.16	29.51	0.3961

**Desicion Tree Base Modeli:** Bu modele iki farklı strateji ile yaklaşılmıştır. Seçilen selection ve boyut indirgeme yöntemlerine göre modelin doğruluk skoru da değiştiği görülmüştür. Bunun sonucunda seçilen selection modellerinden RFE yönteminin daha iyi olduğu görülmüştür. Seçilen boyut indirgeme modellerine göre ise FastICA yönteminin daha başarılı olduğu görülmüştür.

**Dummy Classifier:** Dummy modellerinde stratejiler farklı olarak kullanılmıştır. Proje 3 de most\_frequent Proje 2 de ise stratified kullanmıştır. most\_frequent stratejisi her örneği çoğunluk sınıfı olarak tahmin ettiği için model 1 tahmini yapmaz. Bu yüzden F1, precision ve recall sıfır çıkar. Stratified stratejisi ise sınıf oranlarına göre rastgele tahmin ürettiğinden zaman zaman 1 sınıfını da tahmin eder. Böylece düşük de olsa F1, precision ve recall değerleri oluşur. Bu da stratified stratejisinin daha başarılı olduğunu göstermiştir.

**Gaussian NB:** Bu modelde farklı olarak proje 4'te var\_smoothing parametresi kullanılmıştır. var\_smoothing kullanıldığında, modelin olasılık tahminleri yumuşatıldığı için dağılımlar daha stabil hale gelmiştir, böylece bu yöntem daha başarılı olmuştur.

**Logistic Regresyon:** Bu modelde farklı parametreler kullanılmıştır. Proje 5 veri dengesizliğinin önüne geçmek için dengeleme parametresi kullanmıştır ve bunun sonucunda doğruluk skorunun düştüğü ama daha gerçekçi bir skor olduğu görülmüştür. Seçilen selection yöntemleri karşılaştırıldığında Mutual Information yöntemi daha başarılı olmuştur. Boyut indirgeme yöntemlerinde ise LDA yönteminin daha başarılı olduğu görülmüştür.

## 9) Riskler ve Azaltma Yöntemleri

**Veri Riskleri:** Proje kapsamında kullanılan uçuş veri seti yaklaşık 5.8 milyon satır içerdiğinden, çalıştırma sırasında yüksek RAM kullanımı oluşturmuş ve işlem süresini artırmıştır. Ayrıca veri setindeki ARRIVAL\_DELAY ve DEPARTURE\_DELAY sütunları, modelin tahmin etmeye çalıştığı “gecikme” bilgisini doğrudan içerdiği için data leakage riski yaratmıştır. Bu durum modelin gerçek performansını bozabileceğinden, preprocessing aşamasında bu sütunlar tamamen kaldırılmıştır.

Aynı şekilde ORIGIN\_AIRPORT ve DESTINATION\_AIRPORT sütunlarında çok sayıda benzersiz havalimanı kodu bulunduğu için, yüksek boyut problemi oluşmuştur. Bu nedenle bu kategorik değişkenler silinmiştir. Bu işlemlerle hem sızıntı engellenmiş hem de bellek kullanımı azaltılmıştır.

**Azaltıcı Yöntemler (Mitigations):** Azaltıcı yöntemler olarak ekip üyeleri çoğunlukla boyut indirmeleri kullanmıştır:

### Emine Güneş – LDA

LDA, gecikme durumunu (IS\_DELAYED) tahmin eden sınıflandırma modellerinde (Dummy ve Gaussian Naive Bayes) performansı değerlendirmek için kullanılmıştır. Bu yöntem sınıflar arasındaki ayrımı güçlendirdiği için, modelin gecikme tahminindeki ayrım gücünü gözlemlemeyi sağlamıştır.

### Oğuzhan Özdemir – LDA

LDA, Logistic Regression ve Naive Bayes modelleri üzerinde uygulanarak, farklı denetimli algoritmaların sınıf ayrımı yapan eksenlerde nasıl performans gösterdiği karşılaştırılmıştır.

### Zekeriya Deniz Uğurlu – PCA

PCA, veri setindeki varyansı en iyi temsil eden bileşenleri çıkararak boyutu azaltmak

için kullanılmıştır. Gürültüyü azaltıp veri setini daha yönetilebilir hale getirerek hem hız hem de model stabilitesi açısından iyileşme sağlamayı amaçlamıştır.

#### **Muhammet Enes İnal – FastICA**

FastICA, veri setindeki bileşenleri istatistiksel olarak bağımsız hale getirerek yeni bir özellik temsili oluşturmak için kullanılmıştır. Böylece karmaşık yapılar daha ayrıştırılabilir forma getirilmiş ve gizli bağımsız bileşenlerin modeller üzerindeki etkisi değerlendirilmiştir.

#### **Muhammed Mert Oruç – Truncated SVD**

Truncated SVD, encoded özelliklerin çok yüksek boyutlu yapısını daha yönetilebilir hale getirmek için uygulanmış ve özellikle ilk 10 bileşen kullanılarak modelin hem hız hem de kaynak tüketimi optimize edilmiştir.

## **10) Kullanılan Araçlar**

**Python sürümü:** Python 3.11

**Ana Kütüphaneler:** Pandas, Numpy, Matplotlib, Seaborn, SkLearn, Warnings, GridSearchCV

**random seeds:** random\_state=42, random\_state=45

**Geliştirilen kodlar:** <https://github.com/Oguzhan1511/Turbulance>

## **11) Beklenen Sonuçlar ve Görselleştirme Planı**

### **Göstermeyi beklediğiniz tablolar/grafikler:**

Performans Karşılaştırma Tabloları: Geliştirilen tüm modellerin Train ve Test verileri üzerindeki Accuracy , F1-Score , Recall, ROC-AUC ve PR-AUC yan yana gösteren özet bir veri çerçevesi (DataFrame) sunulacaktır. Bu tablo hangi modelin en dengeli sonucu verdiğini gösterecektir.

Karışıklık Matrisi(Confusion Matrix): Sınıf dengesizliği nedeniyle modelin “Gecikme var” sınıfını ne kadar doğru tahmin net bir şekilde gösterir.

ROC Eğrileri: Modellerin pozitif sınıfı ayırt etme yeteneğini göstermek için ROC eğrileri çizdirilecek ve eğri altında kalan (AUC) üzerinden modeller arası sıralama yapmaktadır..

PCA Varyans Grafiği: Boyut indirgeme işlemi uygulanan modeller için, seçilen bileşen sayısının toplam varyansın ne kadar açıkladığını gösteren “Kümülatif Varyans Grafiği” sunularak veri kaybı oranı gösterilmektedir.

### **Yorumlanabilirlik Yaklaşımı(Interpretability Approach):**

Özellik Önem Düzeyleri: Ağaç Tabanlı modeller kullanıldığında, modelin karar verirken hangi değişkenlere daha fazla ağırlık verdiğini gösteren “Feature Importance” çubuk grafikleri gösterilecektir.

Katsayı Analizi: Lojistik Regresyon ve Ridge Classifier gibi doğrusal modeller için,özniteliklerin katsayıları incelenerek, hangi değişkenini gecikme olasılığını arttırdığı veya azalttığı yorumlanacaktır.

## 12)Referanslar

1-) U.S. Department of Transportation (DOT), "2015 Flight Delays and Cancellations," Kaggle, 2016. [Online]. Erişim adresi: <https://www.kaggle.com/datasets/usdot/flight-delays>.

2-)M. Chhibber, "Flight Delay Predictions," Kaggle, [Online]. Erişim adresi: <https://www.kaggle.com/code/manasichhibber/flight-delay-predictions>.

3-)Levaniz, "Machine Learning Analysis of Flights Data," Kaggle, [Online]. Erişim adresi: <https://www.kaggle.com/code/levaniz/machine-learning-analysis-of-flights-data>.

4-)Abhishek, "2015 Flight Delays and Cancellation Prediction," Kaggle, [Online]. Erişim adresi: <https://www.kaggle.com/code/abhishek211119/2015-flight-delays-and-cancellation-prediction>

# FİNAL RAPORU

## Gelişmiş Modeller:

**Emine Güneş: Gradient Boosting Classifier:** Genellikle karar ağaçlarını ardışık olarak ekleyerek, her yeni modelin bir önceki modelin yaptığı hataları düzeltmesi prensibiyle çalıştığı için seçilmiştir [1]. Hiper-parametre optimizasyonu sayesinde, veri setindeki karmaşık ve doğrusal olmayan ilişkileri, base modellere göre çok daha yüksek bir hassasiyet ve doğrulukla çözümleyebilmektedir.

**CatBoost:** Modelin hassas bir öğrenme gerçekleştirmek amacıyla bu model seçilmiştir [2]. Modelinin karmaşık yapısını verimize en uygun hale getirmek için gridSearchCV yöntemi kullanılmıştır. 2 katlı çapraz doğrulama ile her kombinasyonun başarısı test edilmiştir. Optimizasyon hedefi olarak, sınıf dengesizliğinden etkilenmeyen 'ROC-AUC' skoru kullanılmıştır.

**Muhammed Mert Oruç:** Projede ilk başta base model kullanırken şimdi ise modelin tahmin yeteneğini ve doğruluk oranını artırmak için daha güçlü model olan QDA ve ADABOOST kullandım. Bu modellerin performansını maksimize edecek en iyi yapılandırmayı bulmak için Hiper Parametre Optimizasyonu yöntemi uygulanmıştır.

**QDA:** Bu algoritmada optimize edilen temel parametre **reg\_param** parametresidir. Veri setindeki değişkenler arasında çoklu bağlantı sorunu olduğu için, kovaryans matrisi düzenleyerek modelin daha kararlı çalışmasını sağlamak adına bu değer 0.1 olarak ayarlanmıştır [3].

**AdaBoost:** Bu algoritmada ise modelin karmaşıklığı ile öğrenme hızı arasındaki dengeyi kurmak için, zayıf öğrencilerin birleştirilmesi prensibine dayalı olarak [4] **n\_estimators**(tahminci sayısı) ve **learning\_rate**(öğrenme oranı) parametreleri optimize edilmiştir. Veri setinin büyüklüğü ve işlem maliyeti göz önüne alınarak, aşırı öğrenmeyi engelleyen ve en yüksek doğruluğu sağlayan 50 tahminci ve 1.0 öğrenme oranı en uygun değerler olarak belirlenmiştir.

**Oğuzhan Özdemir:** Projede LightGBM ve Extra Tree Classifier kullanılmıştır.

**Extra Tree Classifier:** Bu model çok karmaşık ve büyük veri seti olduğu için ve varyansı düşürme yeteneğinden dolayı [5] GridSearchCV kullanılmamıştır bunun yerine manuel hiperparametre ayarlama yöntemi kullanılmıştır.

**LightGBM:** Bu model sütunlardaki özel karakterlere hassas olduğundan dolayı bunu ayarlamak için **lgb.LGBMClassifier** parametresi kullanılmıştır. [6]

İki modelde de metrik olarak accuracy, ROC AUC, PR AUC, MAE ve F1 Score kullanıldı. Veri seti dengesizliğinden dolayı PR AUC ve F1 score değerleri modellerin başarısı hakkında daha çok bilgi vermiştir.

**Muhammet Enes İnal:** Projede XGBoost ve Stacking Classifier modelleri kullanılmıştır.

**XGBoost:** Bu model özellikle dengesiz veri setlerinde daha başarılı çalışmaktadır [7]. scale\_pos\_weight parametresi dengesiz veri setini yönetmek için eklenmiştir. Bunun sayesinde gecikmeleri daha iyi yakalamıştır. n\_estimators=100 ve max\_depth=10 parametleri ile de aşırı öğrenmenin önüne geçmeyi hedeflenmiştir.

**Stacking Classifier:** Bu modelde parametre ayarlaması yerine algoritmik çeşitlilik kullanılmıştır. Doğrusal bir model (Logistic Regression), kural tabanlı bir model (Decision Tree) ve boosting tabanlı bir model (XGBoost) bir araya getirilerek [8] varyansın düşürülmesi hedeflenmiştir.

İki modelde de metrik olarak accuracy, ROC AUC, PR AUC, MAE ve F1 Score kullanıldı.

Metrikler sonucuna bakıldığında XGBoost en doğru ve stabil model çıkmıştır.

**Zekeriya Deniz Uğurlu:** Projede model çeşitliliğini sağlamak ve farklı algoritmaların başarısını kıyaslamak adına, bagging yöntemi RandomForestClassifier ve büyük veri setinde yüksek performans sergileyen boosting yöntemi olan HistGradientBoosting kullanılmıştır.

**RandomForestClassifier:** Bu model çok sayıda karar ağacını bir araya getirerek aşırı öğrenmeye karşı dirençli olması sebebiyle seçilmiştir [9]. Modelin performansını en üst düzeye çıkartmak için RandomizedSearchCV yöntemi ile hiperparametre optimizasyonu yapılmıştır. Veri setindeki dengesizliği yönetmek ve modelin gecikme durumlarını yakalama hassasiyetini artırmak için 'class\_weight' parametresi manuel olarak {0: 1, 1: 3} şeklinde ayarlanmıştır. Bu sayede F1 skorunun iyileştirilmesi hedeflenmiştir. Ayırt edicilik gücünü en yükseğe çıkarmak için optimizasyon metriği olarak 'ROC-AUC' kullanılmış ve işlem süresini kısaltmak adına parametre arama işlemi belirli bir sayıdaki örnek üzerinde gerçekleştirilmiştir.

**HistGradientBoosting:** Büyük veri setlerinde eğitim hızını ve büyük bir sorun yaratan bellek alanını en iyi şekilde kullanması amacıyla histogram tabanlı çalışan bu model tercih edilmiştir [6]. İşlem maliyetini düşürmek adına 200.000 örnek üzerinde RandomizedSearchCV yöntemiyle hiperparametre optimizasyonu yapılmış ve hedef metrik olarak 'f1' skoru kullanılmıştır. Sınıf dengesizliğini yönetmek adına 'class\_weight' parametresi {0: 1, 1: 2} olarak ayarlanmıştır. Modelin ham olasılık çıktıları üzerinde optimal eşik değeri ayarlaması yapılarak, doğruluk oranını belirli bir seviyenin üstünde tutarken aynı zamanda F1 skorunu en yükseğe taşıyan optimum eşik değeriyle tahminler oluşturulmuştur.

## Kullanılan Metrikler Karşılaştırılması:

### Emine Güneş :

Model	accuracy	F1 Score	precision	recall	ROC AUC	PR AUC	MAE
Gradient Boosting Classifier	82.09	0.03	0.50	0.02	67.88	32.34	27.50
Gradient Boosting Classifier +CA	82.27	0.05	0.61	0.03	67.85	32.40	27.46
Gradient Boosting Classifier NB +LDA	82.09	0.00	0.00	0.00	63.02	25.74	28.49
Cat Boost	82.20	0.03	0.64	0.01	67.76	33.00	27.62
Cat Boost +CA	82.30	0.04	0.72	0.02	67.78	33.14	27.60
Cat Boost +LDA	82.09	0.00	0.00	0.00	63.02	25.81	28.62

### Sınıf Bazlı Kırılım:

Model	% Precision	% Recall	% F1-Score
Gradient Boosting Classifier 0 (Gecikmedi)	82.30	99.62	90.13
Gradient Boosting Classifier 1 (Gecikti)	50.04	1.73	3.35
CatBoost 0 (Gecikmedi)	82.27	82.27	90.21
CatBoost 1 (Gecikti)	64.38	1.31	2.57

**Zekeriya Deniz Uğurlu :**

Model	accuracy	F1 Score	precision	recall	ROC AUC	PR AUC	MAE
RandomForestClassifier	81.70	0.41	0.48	0.35	74.78	43.56	18.30
RandomForestClassifier + PCA	75.66	0.30	0.31	0.30	65.80	28.87	24.34
RandomForestClassifier + RFE	80.26	0.42	0.44	0.41	73.48	43.45	19.74
HistGradientBoostingClassifier	79.33	0.39	0.41	0.37	72.23	37.86	22.83
HistGradientBoostingClassifier + PCA	79.47	0.26	0.36	0.20	66.73	28.87	22.64
HistGradientBoostingClassifier + RFE	79.27	0.38	0.41	0.36	71.66	26.27	21.18

**Sınıf Bazlı Kırılım:**

Model	Precision	Recall	F1-Score
RandomForestClassifier 0 (Gecikmedi)	0.87	0.92	0.89
RandomForestClassifier 1 (Gecikti)	0.49	0.36	0.41
HistGradientBoostingClassifier 0 (Gecikmedi)	0.87	0.88	0.88
HistGradientBoostingClassifier 1 (Gecikti)	0.41	0.38	0.39

**Muhammed Mert Oruç :**

Model	accuracy	F1-score	precision	recall	ROC- AUC	PR-AUC	MAE
QDA	74.13	0.24	0.25	0.23	61.29	24.48	25.86
QDA+A NOVA F	79.12	0.14	0.26	0.09	61.49	24.35	20.89
QDA+S VD	75.73	0.22	0.26	0.19	61.29	24.37	24.27
ADA BOOST	82.09	0.00	0.00	0.00	63.88	26.83	17.90
ADA BOOST +ANOV A F	82.08	0.00	0.00	0.00	63.48	26.57	17.92
ADA BOOST +SVD	82.08	0.00	0.17	0.00	62.93	25.72	17.93

**Sınıf Bazlı Kırılım:**

Model	Precision	Recall	F1 Score
QDA 0 (gecikmedi)	0.84	0.85	0.84
QDA 1 (gecikti)	0.25	0.23	0.24
Ada Boost 0 (gecikmedi)	0.82	1.00	0.90
Ada Boost 1 (gecikti)	0.00	0.00	0.00

**Oğuzhan Özdemir:**

Model	accuracy	F1 Score	Recall	Precision	ROC AUC	PR AUC	MAE
LightGBM	82.39	0.07	0.24	0.65	70.95	36.72	28.42
LightGBM + Selection	82.31	0.45	0.24	0.65	69.84	34.92	29.94
LightGBM+ LDA	82.10	0.00	0.00	0.00	61.81	24.89	28.89
Extra Tree Classifier	81.23	0.25	0.002	0.80	68.07	31.83	0.18
Extra Tree Classifier+ Selection	<b>81.24</b>	<b>0.25</b>	<b>0.0015</b>	<b>0.75</b>	<b>68.08</b>	<b>31.85</b>	<b>0.18</b>
Extra Tree Classifier + LDA	72.20	0.21	0.0039	0.33	53.06	19.63	0.27

**Sınıf Bazlı Kırılım:**

Model	Precision	Recall	F1 Score
LightGBM 0 (gecikmedi)	0.82	1.00	0.90
LightGBM 1 (gecikti)	0.66	0.02	0.05
Extra Tree Classifier 0 (gecikmedi)	0.82	1.00	0.90
Extra Tree Classifier 1 (gecikti)	0.80	0.00	0.00

**Muhammet Enes İnal:**

Model	Accuracy (%)	Precision (%)	Recall (%)	ROC AUC (%)	PR AUC (%)	F1 Score	MAE
Stacking (Base)	67.72	29.23	56.46	68.52	33.25	0.38	0.32
Stacking + SelectFromModel	68.88	29.84	54.60	68.53	33.53	0.38	0.31
Stacking + PCA	58.98	24.80	63.53	64.70	27.70	0.35	0.41
XGBoost (Base)	70.05	30.37	52.05	68.21	33.31	0.38	0.30
XGBoost + SelectFromModel	68.79	29.65	54.13	68.07	33.34	0.38	0.31
XGBoost + PCA	66.80	26.56	48.39	63.83	27.32	0.38	0.33

**Sınıf Bazlı Kırılım:**

Model	Sınıf (%)	Precision (%)	Recall (%)	F1-Score (%)
XGBoost Base	0 (Gecikmedi)	87.61	73.97	80.22
XGBoost Base	1 (Gecikti)	30.37	52.05	38.36
Stacking Base	0 (Gecikmedi)	88.08	70.18	78.12
Stacking Base	1 (Gecikti)	29.23	56.46	38.52

### Grup üyelerinin en iyi ana modelleri:

Üye	Model	accuracy	F1 Score	precision	recall	ROC AUC	PR AUC	MAE
<b>Emine Güneş</b>	Gradient Boosting Classifier +CA	82.27	0.05	0.61	0.03	67.85	32.40	27.46
<b>Zekeriya Deniz Uğurlu</b>	RandomForestClassifier	81.70	0.41	0.48	0.35	74.78	43.56	18.30
<b>Muhammed Mert Oruç</b>	QDA+ANOVA F	79.12	0.14	0.26	0.09	61.49	24.35	20.89
<b>Oğuzhan Özdemir</b>	Extra Tree Classifier+ Selection	81.24	0.25	0.0015	0.75	68.08	31.85	18.01
<b>Muhammet Enes İnal</b>	Stacking + SelectFromModel	68.88	0.29	0.54	0.68	33.53	38.59	31.05

### En İyi Model Seçimi:

Grup üyeleri tarafından geliştirilen modeller, uçuş gecikme tahmini probleminin doğası ve kullanılan veri setinin dengesiz yapısı[10] göz önüne alınarak belirlenen başarı kriterleri çerçevesinde karşılaştırılmıştır.

Veri setimizde uçuşların büyük çoğunluğu zamanında gerçekleşmiştir. Bu tür dengesiz veri setlerinde modeller, sadece çoğunluk sınıfını yani gecikme olmamasını tahmin ederek yapay bir yüksek doğruluk elde edebilir.

Bir modelin başarısı sadece doğru bilmesi değil, "gecikmeli" ve "gecikmesiz" uçuşları birbirinden ne kadar net ayırabildiğiyle de ilgilidir.

Analiz sonuçlarına göre:

Muhammet Enes İnal %68.88 ve Muhammed Mert Oruç %79.12 tarafından geliştirilen modeller, öncelikli kriter olan %80 doğruluk barajının altında kaldıkları için elenmiştir.

Emine Güneş'in modeli %82.27 ile en yüksek doğruluk oranına ulaşsa da, 0.05 gibi çok düşük bir F1 skoru elde etmesi nedeniyle gecikme tespiti başarısız bulunmuştur.

Oğuzhan Özdemir'in modeli ise doğruluk %81.24 ve F1 skoru 0.25 kriterlerini sağlasa da, modelin ayırt ediciliğini gösteren ROC AUC değerinin 68.08'de kalması sebebiyle 70.00 barajını geçememiştir.

Zekeriya Deniz Uğurlu tarafından geliştirilen RandomForestClassifier modeli %81.70 doğruluk, 0.41 F1 skoru ve 74.78 ROC AUC değeri ile projenin belirlenen üç temel başarı kriterini de aynı anda sağlayan tek model olmuştur. Bu nedenle, hem kararlılık hem de ayırt edicilik açısından en dengeli performansı sunan RandomForestClassifier projenin ana modeli olarak seçilmiştir. [11],[12]

## Kaynakça:

[1] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.

[2] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: unbiased boosting with categorical features," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems (NeurIPS)*, 2018, pp. 6639–6649.

[3] E. Alpaydın, *Introduction to Machine Learning*, 4th ed. Cambridge, MA: MIT Press, 2020. (QDA ve temel sınıflandırıcılar için standart ders kitabı)

[4] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[5] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine Learning*, vol. 63, no. 1, pp. 3–42, 2006.

[6] G. Ke et al., "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," in *Advances in Neural Information Processing Systems (NIPS)*, 2017, vol. 30, pp. 3146–3154. (Not: HistGradientBoosting, LightGBM mimarisini temel aldığı için bu kaynak her ikisi için de geçerlidir).

[7] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.

[8] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, pp. 241–259, 1992.

[9] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[10]- A Deep Learning Approach to Flight Delay Prediction ( Veri Seti Dengesizliği)  
Kim, Y. J., Choi, S., Briceno, S., & Mavris, D. N. (2016)

[11]-An Ensemble-Based Approach for Flight Delay Prediction (Ensemble Model)  
Belcastro, L., Marozzo, D., & Trunfio, P. (2016)

[12]-Precision-Recall Curves for Classification with Imbalanced Data( F1 score)  
Saito, T., & Rehmsmeier, M. (2015)