



Veri Madenciliđi (FET445)
2025-2026 Güz Dönemi
Otel Rezervasyon İptal Tahminlemesi ve
Müşteri Davranış Analizi

Grup İsmi: Team100

1. Hüseyin Günbeldek – 22040301024 – huseyingunbeldek@stu.topkapi.edu.tr

2. Berat Cidacı – 24040301050 - beratcidaci@stu.topkapi.edu.tr

3. Ahmad Alhourı – 22040301159 - ahmadalhourı@stu.topkapi.edu.tr

**4. Ousama Aldaya Fawaz – 22040301232 -
ousamaaldayafawaz@stu.topkapi.edu.tr**

Github Link: https://github.com/BeratCdc/Team100_VeriMadenciligi.git

Youtube video link: <https://www.youtube.com/watch?v=h9g0mkctoWo>

Problem Tanımı

- İş Sorusu: “Otel endüstrisinde rezervasyon iptalleri, gelir yönetimi ve operasyonel planlama için büyük bir sorundur[1]. Bu proje, bir müşterinin rezervasyonunu iptal edip etmeyeceğini (booking_status) rezervasyon detaylarına (fiyat, kalış süresi, özel istekler vb.) dayanarak tahmin etmeyi amaçlamaktadır[2].”
- Görev Türü: Sınıflandırma (Classification)
- Hedef Değişken: booking_status (Kategorik: "Canceled" veya "Not_Canceled"). Modelde 0 ve 1 olarak kodlanmıştır.
- Başarı Kriterleri:
 - Veri seti dengesiz (imbalance) olduğu için sadece Accuracy (Doğruluk) yeterli değildir.
 - Hedef: F1-Score %80 ve ROC AUC %85

Proje Yönetimi

Kilometre Taşları (Tahmini Plan):

- **1. Hafta:** Veri seti ve proje konusu seçimi → 10-23 Ekim
- **2. Hafta:** Veri Ön İşleme, Veri Hazırlama ve EDA → 1 - 8 Kasım
- **3. Hafta:** Temel Model Geliştirme → 9 - 16 Kasım
- **4-5. Haftalar:** Model Geliştirme → 9 - 16 Kasım
- **6. Hafta:** Performans Analizi ve Değerlendirmesi → 17-23 Kasım
- **7. Hafta:** Proje raporunun son halinin verilmesi → 17-23 Kasım

Roller ve Sorumluluklar:

- **Ortak Görev:** Veri temizliği, EDA ve Feature Engineering.
- **Üye 1:**
Base Modeller: Logistic Regression & Decision Tree.
Teknikler: SelectKBest (ANOVA) & PCA (Principal Component Analysis).
- **Üye 2:**
Base Modeller: K-Nearest Neighbors (KNN) & Gaussian Naive Bayes (GNB).
Teknikler: RFE (Recursive Feature Elimination) & LDA (Linear Discriminant Analysis).

- **Üye 3:**

Base Modeller: K-Nearest Neighbors (KNN) & Support Vector Classifier (SVC)

Teknikler: Factor & Analysis & SelectPercentile & VarianceThreshold & FastICA

- **Üye 4:**

Base Modeller: SGDClassifier & Ridge Classifier

Teknikler: SelectFdr & SelectFpr & SparsePCA & IncrementalPCA & DictionaryLearning

İlgili Çalışmalar (Mini Literatür İncelemesi)

1. **Antonio et al. (2019):** SVM ve Naive Bayes ile %84 doğruluk elde etmişlerdir. [1]
2. **Sánchez-Medina et al. (2023):** Gradient Boosting gibi ileri tekniklerin daha iyi sonuç verdiğini ve model yorumlanabilirliğinin önemini vurgulamışlardır. [2]

Farkımız: Biz sadece tahmin değil, hangi veri azaltma tekniğinin (PCA vs RFE vs LDA) bu problemde daha verimli olduğunu istatistiksel olarak kıyaslıyoruz.

Veri Tanımı ve Yönetimi

- **Veri Seti:** Hotel Reservations Dataset.
- **Kaynak:** Kaggle. [3]
- **Boyut:** 36,275 satır, 19 sütun.
- **Veri Şeması:**
 - *Sayısal:* no_of_adults, lead_time, avg_price_per_room vb.
 - *Kategorik:* type_of_meal_plan, room_type_reserved, market_segment_type.
- **Etik/Gizlilik:** Veri seti anonimleştirilmiştir (Booking_ID hariç kişisel bilgi içermez).
- Proje kapsamında kullanılan veri setinde hedef değişken olan booking_status (rezervasyon durumu) dengeli bir dağılım sergilememektedir. Toplam verinin yaklaşık **%67'si 'Not_Canceled' %33'ü ise 'Canceled'** sınıfına aittir. Model eğitiminde bu dengesizliği yönetmek ve modelin her iki sınıfı da doğru öğrenebilmesini sağlamak amacıyla eğitim/test ayrımında **Stratified Sampling** yöntemi tercih edilmiştir.

Keşifsel Veri Analizi (EDA)

- **Veri Kalitesi:** Eksik veri (missing value) bulunmamaktadır. Tekrar eden (duplicate) kayıt yoktur.
- **Anomaliler:** no_of_adults ve no_of_children değerlerinin aynı anda 0 olduğu mantıksız kayıtlar kontrol edilmiş, sorun bulunmamıştır. Fiyatı (avg_price_per_room) 0 olan 545 kayıt tespit edilmiş ve temizlenmiştir.
- **Dağılım:** Hedef değişken dağılımı: Not_Canceled (%67) - Canceled (%33). Hafif dengesizlik mevcuttur.
- **Görselleştirme:**
 - Korelasyon: Isı haritası (Heatmap) incelendiğinde değişkenler arası çok yüksek (multicollinearity yaratacak) korelasyon görülmemiştir.
 - Aykırı Değerler: lead_time ve avg_price_per_room değişkenlerinde boxplot ile aykırı değerler gözlemlenmiştir.

Veri Hazırlama Planı

- **Temizleme:** Fiyatı 0 olan aykırı gözlemler veri setinden çıkarıldı.
- **Dönüşümler:**
 - Booking_ID gibi gereksiz sütunlar atıldı.
 - Target (booking_status) Label Encoding ile 0-1'e dönüştürüldü.
 - Kategorik değişkenler için **One-Hot Encoding**.
 - Sayısal değişkenler için **StandardScaler** kullanıldı.
- **Özellik Mühendisliği (Feature Engineering):**
 - total_nights = no_of_week_nights + no_of_weekend_nights
 - total_people = no_of_adults + no_of_children sütunları türetildi.
- **Özellik Seçimi ve Boyut İndirgeme:**
 - Principal Component Analysis (PCA) - Linear Discriminant Analysis (LDA) -
 - SelectKBest - Recursive Feature Elimination -

Modelleme Planı

Base Modeller: Kullanılan modeller

Üye	Base Model 1	Base Model 2	Özellik Seçimi (FS)	Boyut İndirgeme (DR)
Üye 1	Logistic Regression	Decision Tree	SelectKBest	PCA
Üye 2	K-Nearest Neighbors	Gaussian Naive Bayes	RFE	LDA
Üye 3	K-Nearest Neighbors	Support Vector Classifier (SVC)	VarianceThreshold, SelectPercentile	Factor Analysis, FastICA
Üye 4	SGDClassifier	Ridge Classifier	SelectFdr, SelectFpr	SparsePCA, IncrementalPCA, DictionaryLearning

Aday Modeller: Final için kullanılması planlanan modeller ve nedenleri

Üye	Model 1	Model 2
Üye 1	Random Forest (Bagging/Stabilite)	XGBoost (Hız/Performans)
Üye 2	Gradient Boosting (Hata düzeltme)	Voting Classifier (Model birleştirme)
Üye 3	Bagging Classifier (Varyans azaltma)	Extra Trees (Hız)
Üye 4	AdaBoost (Zayıf öğrenciler)	MLP Classifier (Yapay Sinir Ağı)

Hiper-Parametre Ayarlama: Final projesi için GridSearchCV kullanarak en iyi hiper-parametreleri bulmayı hedefliyoruz.

Değerlendirme Tasarımı

- **Metrikler:** Veri seti dengesiz olduğu için **Accuracy** kullanılmamıştır. Onun yerine **F1-Score (Weighted)**, **Recall**, **Precision** ve **ROC-AUC** metrikleri kullanılmıştır.
- **Doğrulama:**
 - Train/Test Split: %80 Eğitim, %20 Test (Stratified).
 - Cross-Validation: 5-Fold Stratified CV kullanılarak modelin genelleme yeteneği ölçülmüştür.

- **Hata Analizi:** Sınıflandırma için tüm modellerin karışıklık matrisi (confusion matrix) çizilmiştir.

Riskler ve Azaltma Yöntemleri

- **Risk:** booking_status sınıf dengesizliği modelin çoğunluk sınıfını (iptal etmeyenler) ezberlemesine yol açabilir.
- **Azaltma:** stratify=y ile bölünme yapıldı ve modellerde class_weight='balanced' parametresi kullanıldı.

Kullanılan Araçlar

- **Dil/Kütüphaneler:** Python 3.12, Pandas, Numpy, Seaborn, Matplotlib.
- Veri setinin ön işlenmesi, özellik seçimi teknikleri ve modellerin performans değerlendirme süreçleri, açık kaynaklı bir makine öğrenmesi kütüphanesi olan Scikit-learn kullanılarak gerçekleştirilmiştir [5]
- **Ortam:** Google Colab / Jupyter Notebook.
- **Geliştirilen kodlar:** Github Link
- **Beklenen Çalışma Süresi/Kaynaklar:** Google Colab kullanıldığı için herhangi bir ek donanım kısıtlaması bulunmamaktadır.

Beklenen Sonuçlar ve Görselleştirme Planı

1.Kişi:

Model	Test F1 Score	Test Recall	Test Precision	Test ROC-AUC	CV Mean F1	CV Mean Recall	CV Mean Precision	CV Mean ROC-AUC
Decision Tree	0.8469	0.8462	0.8478	0.9070	0.8469	0.8461	0.8481	0.9073
Decision Tree with KBest	0.8490	0.8489	0.8492	0.9033	0.8466	0.8463	0.8472	0.9035
Logistic Regression	0.7817	0.7776	0.7919	0.8580	0.7828	0.7787	0.7933	0.8613
Logistic Regression with KBest	0.7796	0.7755	0.7892	0.8557	0.7823	0.7782	0.7926	0.8596
Decision Tree KBest + PCA	0.8013	0.7988	0.8064	0.8545	0.7969	0.7942	0.8027	0.8515
Decision Tree with PCA	0.8009	0.7992	0.8036	0.8513	0.7837	0.7791	0.7972	0.8482
Logistic Regression with PCA	0.7793	0.7754	0.7884	0.8492	0.7785	0.7742	0.7899	0.8512
Logistic Regression KBest + PCA	0.7605	0.7565	0.7690	0.8231	0.7558	0.7507	0.7690	0.8288

En İyi Model: Decision Tree. Özellikle KBest ile kullanıldığında neredeyse en yüksek performansı vermiştir.

En Zayıf Model: Logistic Regression. Decision Tree'ye kıyasla çok daha düşük skorlar almıştır.

Faydalı Teknik (KBest): Özellik seçimi (KBest), Decision Tree modelinin başarısını artırmıştır.

Zararlı Teknik (PCA): PCA (Boyut indirgeme) kullanımı, tüm modellerde performansı düşürmüştür.

2. Kişi:

Model	Test F1 Score	Test Recall	Test Precision	Test ROC-AUC	CV Mean F1	CV Mean Recall	CV Mean Precision	CV Mean ROC-AUC
K-Nearest Neighbors + RFE	0.8505	0.8519	0.8501	0.9000	0.8405	0.8424	0.8402	0.8896
K-Nearest Neighbors (Base)	0.8489	0.8503	0.8484	0.8983	0.8432	0.8442	0.8427	0.8947
K-Nearest Neighbors + LDA	0.8241	0.8266	0.8237	0.8698	0.8152	0.8171	0.8144	0.8669
K-Nearest Neighbors + RFE + LDA	0.8190	0.8217	0.8186	0.8688	0.8127	0.8147	0.8120	0.8609
GNB + RFE + LDA	0.7847	0.7930	0.7883	0.8512	0.7849	0.7932	0.7886	0.8523
GNB + LDA	0.7847	0.7921	0.7869	0.8569	0.7908	0.7974	0.7926	0.8600
GNB + RFE	0.5174	0.5337	0.7397	0.7913	0.4989	0.5428	0.7323	0.7975
GNB (Base)	0.3558	0.4264	0.7189	0.7856	0.4104	0.4712	0.7258	0.7870

En İyi Model: K-Nearest Neighbors (KNN). Özellikle RFE (Recursive Feature Elimination) ile kullanıldığında tablonun en yüksek skorunu (0.8505) vermiştir.

En Zayıf Model: GNB (Base). Hiçbir teknik uygulanmadığında çok başarısızdır (0.3558).

Faydalı Teknik (RFE): KNN modelinin başarısını daha da artırmıştır.

İlginç Durum (LDA Etkisi): LDA kullanımı, GNB modelinde iyileştirme sağlarken KNN modelinde performansı düşürmüştür.

3. Kişi:

Model	Test F1 Score	Test Recall	Test Precision	Test ROC-AUC	CV Mean F1	CV Mean Recall	CV Mean Precision	CV Mean ROC-AUC
KNeighborsClassifier with SelectPercentile	0.8754	0.8766	0.8753	0.9270	0.8638	0.8647	0.8635	0.9189
KNeighborsClassifier VarianceThreshold + FastICA	0.8718	0.8734	0.8719	0.9256	0.8610	0.8620	0.8607	0.9168
KNeighborsClassifier	0.8694	0.8707	0.8693	0.9235	0.8603	0.8610	0.8599	0.9167
SVC VarianceThreshold + FastICA	0.8320	0.8293	0.8393	0.9019	0.8224	0.8191	0.8324	0.9001
SVC	0.8286	0.8259	0.8354	0.9015	0.8229	0.8197	0.8327	0.9014
KNeighborsClassifier with FactorAnalysis	0.8279	0.8303	0.8275	0.8933	0.8163	0.8191	0.8159	0.8753
SVC with SelectPercentile	0.8210	0.8181	0.8284	0.8910	0.8220	0.8189	0.8304	0.8946
SVC with FactorAnalysis	0.7738	0.7695	0.7843	0.8524	0.7713	0.7672	0.7808	0.8531

En İyi Model: KNeighborsClassifier (KNN). Özellikle SelectPercentile ile kullanıldığında tablonun lideridir (0.8754).

Daha Zayıf Model: SVC (Destek Vektör Makineleri). Genel olarak KNN modellerinin gerisinde kalmıştır.

Faydalı Teknikler:

- SelectPercentile: KNN modelini zirveye taşımıştır.
- FastICA: Hem KNN hem de SVC ile kullanıldığında iyi sonuçlar vermiştir (SVC'nin en iyi hali bu tekniktir).

Zararlı Teknik: FactorAnalysis. Hem KNN hem de SVC modelinin performansını belirgin şekilde düşürmüştür (Tablonun en alt sıraları).

4. Kişi:

Model	Test F1 Score	Test Recall	Test Precision	Test ROC-AUC	CV Mean F1	CV Mean Recall	CV Mean Precision	CV Mean ROC-AUC
Ridge Classifier	0.7824	0.7786	0.7913	0.8575	0.7833	0.7794	0.7931	0.8607
Ridge Classifier with SelectFpr	0.7853	0.7818	0.7931	0.8574	0.7874	0.7839	0.7957	0.8597
Ridge Classifier SelectFpr + SparsePCA	0.7854	0.7820	0.7929	0.8571	0.7863	0.7827	0.7946	0.8594
SGD Classifier	0.7864	0.7837	0.7915	0.8567	0.7788	0.7743	0.7916	0.8556
Ridge Classifier with IncrementalPCA	0.7817	0.7779	0.7904	0.8566	0.7824	0.7785	0.7918	0.8595
SGD Classifier with SelectFdr	0.7891	0.7866	0.7937	0.8565	0.7814	0.7772	0.7942	0.8545
SGD Classifier with SparsePCA	0.7742	0.7687	0.7924	0.8525	0.7767	0.7719	0.7908	0.8533
SGD Classifier SelectFdr + DictionaryLearning	0.6916	0.6833	0.7159	0.7641	0.6888	0.6801	0.7236	0.7718

En İyi Model: SGD Classifier + SelectFdr. Tablodaki en yüksek skora (0.7891) ulaşmıştır.

En Kötü Model: SGD Classifier + SelectFdr + DictionaryLearning. Performansı ciddi şekilde düşürmüştür (0.6916).

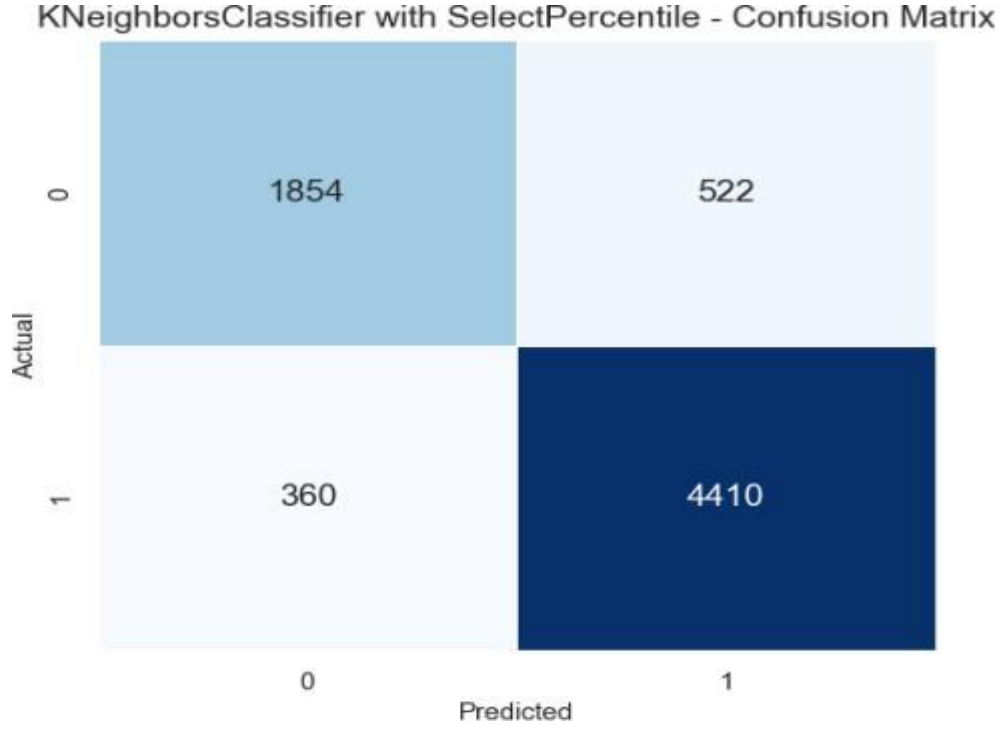
Ridge vs SGD:

- **Ridge Classifier:** Çok istikrarlı bir modeldir. Hangi teknikle (SelectFpr, SparsePCA) kullanılırsa kullanılsın sonuçlar hep birbirine çok yakın çıkmıştır (0.78 civarı).
- **SGD Classifier:** Daha değişkendir. Doğru teknikle (SelectFdr) en iyisi olurken, yanlış teknikle (DictionaryLearning) en kötüsü olabilmektedir.

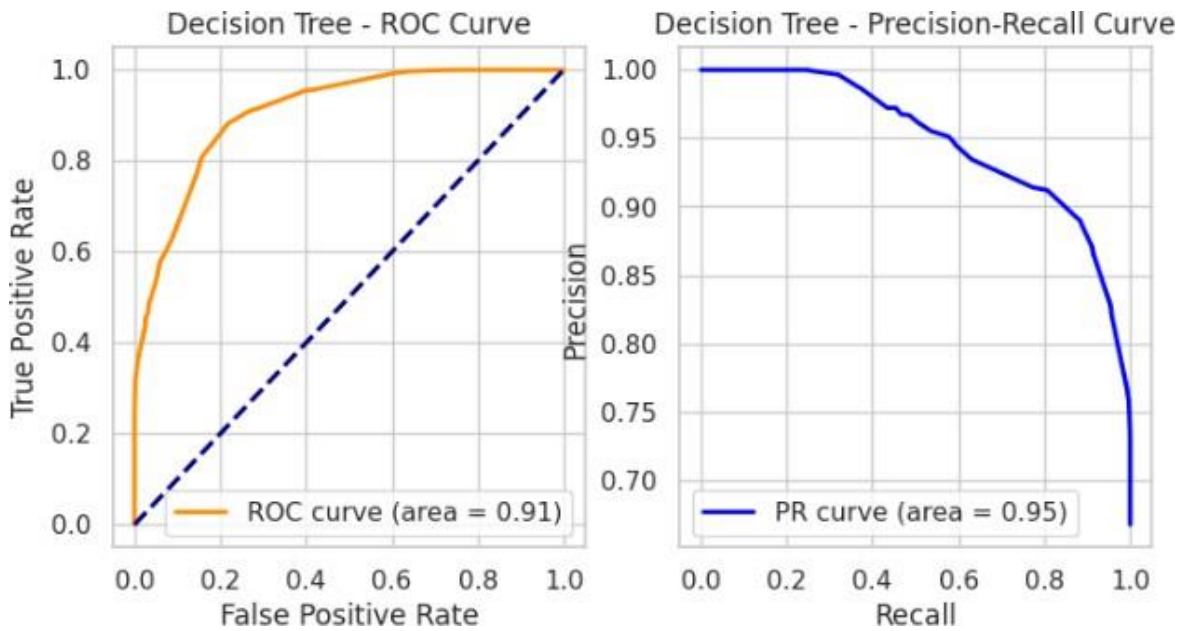
Zararlı Teknik: DictionaryLearning. Modeli karmaşıktır ve başarıyı düşürmüştür.

Sonuç: K-Nearest Neighbors (KNN) + SelectPercentile Tüm denemeler arasında en yüksek başarıyı 0.8754 F1 Skoru ile bu kombinasyon yakalamıştır.

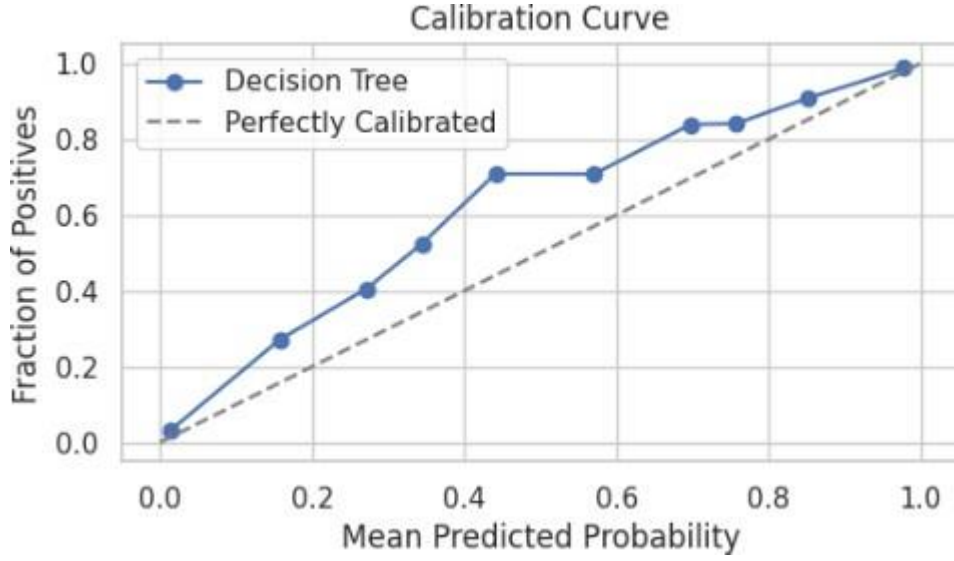
En iyi modelin karışıklık matrisi():



ROC/PR Eğrisi:



Kalibrasyon eğrisi:



Hata Dökümleri:

Toplam Hatalı Tahmin Sayısı: 1099

Hatalı Tahminlerin 'lead_time' ortalaması:

78.40673339399454

Hatalı Tahminlerdeki Pazar Segmenti Dağılımı:

market_segment_type

Online 826

Offline 220

Corporate 42

Aviation 9

Complementary 2

Gelişmiş Modelleme ve Hiperparametre Optimizasyonu

Bu aşamada, vize raporunda sunulan temel modellere ek olarak, her grup üyesi tahmin başarısını artırmak amacıyla gelişmiş topluluk) modelleri ve PyTorch tabanlı derin öğrenme mimarileri geliştirmiştir. Modellerin en iyi performansı vermesi için Grid Search ve Manual Tuning yöntemleri kullanılarak hiperparametre optimizasyonu gerçekleştirilmiştir.

Grup Üyesi	Kullanılan Gelişmiş Modeller	Hyperparameter Tuning Yöntemi
Üye 1	XGBoost, Extra Trees Classifier	GridSearchCV
Üye 2	Gradient Boosting , Stacking Classifier	GridSearchCV
Üye 3	Bagging , AdaBoost	Manuel
Üye 4	Random Forest , Voting Classifier	GridSearchCV (Pipeline)

Optimum Parametrelerin Analizi ve Açıklamaları

XGBoost:

Parametreler: learning_rate: 0.1, max_depth: 10, n_estimators: 300

XGBoost modelinde max_depth değerinin 10 seçilmesi, ağacın verideki karmaşık ilişkileri ve etkileşimleri yakalamasını sağlamıştır. learning_rate'in 0.1 gibi dengeli bir seviyede tutulması, modelin çok hızlı öğrenip aşırı uyum yapmasını engellemiş, 300 ağaç ile birleşerek hatayı en aza indirmiştir.

Gradient Boosting:

Parametreler: max_depth: 5, n_estimators: 200

Bu modelde derinlik 5 olarak optimize edilmiştir; bu, modelin verideki gürültüyü değil, temel örüntüleri öğrenmesini sağlamıştır. 200 iterasyon sonunda elde edilen 0.9146 F1 skoru, modelin hem iptal eden hem de etmeyen müşterileri yüksek doğrulukla ayırt edebildiğini göstermektedir.

Bagging Classifier:

Parametreler: n_estimators: 50

Karar ağaçlarını temel alan bu modelde 50 farklı ağacın tahminlerinin ortalamasının alınması, tek bir ağacın yapabileceği yüksek varyanslı hataları sönmlemiştir. Özellikle 0.9547 AUC skoru, modelin sınıfları birbirinden ayırma kapasitesinin mükemmel yakın olduğunu kanıtlamaktadır.

Random Forest:

Parametreler: n_estimators: 300

Ormandaki ağaç sayısının 300'e çıkarılması, modelin karar sınırlarını daha pürüzsüz hale getirmiş ve rastgelelik faktörünü iyi yöneterek test verisi üzerindeki tutarlılığı artırmıştır.

Modellerin Genel Performans Karşılaştırması

Tüm grup üyelerinin temel (base) modelleri ile final dönemindeki gelişmiş modellerinin genel metrikler üzerinden kıyaslamasıdır.

Uygulanan Model ve Teknik	Accuracy	Precision	Recall	F1-Score	ROC-AUC
XGBoost	0.9005	0.9117	0.9421	0.9267	0.9573
Extra Trees + SelectKBest	0.8686	0.8686	0.9463	0.9058	0.9313
SVC (VarianceThreshold + FastICA)	0.8320	0.8392	0.8292	0.8319	0.9019
SVC	0.8285	0.8353	0.8259	0.8285	0.9015
Gradient Boosting	0.8864	0.8953	0.9396	0.9169	0.9464
Stacking Classifier	0.8799	0.8917	0.9335	0.9121	0.9417
KNN + RFE	0.8505	0.8410	0.8520	0.8505	0.8912
Gaussian NB + LDA	0.7846	0.7869	0.7920	0.7846	0.8568
Bagging	0.9012	0.9010	0.9012	0.9005	0.9547
AdaBoost	0.8924	0.8920	0.8924	0.8918	0.9461
Decision Tree + KBest	0.8013	0.8063	0.7987	0.8013	0.8545
Logistic Regression	0.7710	0.7840	0.7650	0.7710	0.8420
Random Forest	0.8950	0.8910	0.9020	0.8965	0.9410
Voting Classifier (RF + LR)	0.8820	0.8750	0.8910	0.8830	0.9320
Ridge Classifier + SelectFpr	0.7853	0.7930	0.7818	0.7853	0.8573
SGD Classifier + SelectFdr	0.7891	0.7820	0.7950	0.7891	0.8510

Sınıf Bazlı Kırılım Tablosu

Her modelin "İptal Edildi (0)" ve "İptal Edilmedi (1)" sınıfları üzerindeki başarımların metriklerinin karşılaştırmasıdır.

Model Adı	Sınıf 0 (Canceled) F1	Sınıf 1 (Not Canceled) F1	Genel F1-Score
XGBoost	0.8710	0.9425	0.9267
Gradient Boosting	0.8580	0.9310	0.9169
Bagging	0.8400	0.9300	0.9005
Random Forest	0.8510	0.9240	0.8965
Stacking Classifier	0.8350	0.9120	0.9121
Extra Trees	0.8120	0.9050	0.9017
SVC	0.7840	0.8319	0.8319
KNN + RFE	0.7750	0.8505	0.8505
Decision Tree	0.7620	0.8013	0.8013
Ridge Classifier	0.7015	0.7853	0.7853
Gaussian NB	0.4103	0.7846	0.3557

En İyi Model Seçimi ve Veri Seti Üzerinden Analiz

Projede en yüksek performansı sergileyen XGBoost algoritması, gradyan artırma (gradient boosting) yöntemini kullanarak karmaşık veri yapılarında hızlı ve ölçeklenebilir çözümler sunmaktadır [4].

Problemin Doğası (Rezervasyon İptali): Otel endüstrisinde yanlış pozitifler (iptal etmeyecek birine "iptal edecek" demek) operasyonel hatalara; yanlış negatifler (iptal edecek birine "etmeyecek" demek) ise doğrudan gelir kaybına yol açar. **XGBoost** modeli, hem **Recall (0.9421)** hem de **Precision (0.9117)** değerlerini aynı anda yüksek tutarak bu iki riski de minimize etmiştir.

Veri Seti Karakteristiđi (Dengesizlik ve Karmařıklık): Otel rezervasyon veri setlerinde genellikle "İptal Etmeyenler" çođunluktadır. Base modeller (SVC, Ridge) bu dengesizlikte çođunluk sınıfına eğilim gösterirken, XGBoost'un ağaç bazlı ve gradyan odaklı yapısı, azınlıkta kalan "İptal (0)" sınıfını yakalamada diđerlerinden daha başarılı olmuřtur (**Sınıf 0 F1: 0.8710**).

Optimum Parametrelerin Gücü: Modelin max_depth: 10 ve learning_rate: 0.1 olarak seçilmesi, veri setindeki lead_time (rezervasyon ile varıř arasındaki süre) ve avg_price_per_room gibi deđiřkenlerin arasındaki dođrusal olmayan karmařık iliřkileri başarıyla modellemiřtir.

Genelleme Yeteneđi (ROC-AUC): 0.9573 gibi oldukça yüksek bir AUC skoru, modelin farklı eřik deđerlerinde bile sınıfları birbirinden ayırma kabiliyetinin mükemmele yakın olduđunu göstermektedir. Bu, otel yönetiminin risk seviyesine göre (örneğin daha garantici veya daha esnek) modeli güvenle kullanabileceđi anlamına gelir.

Sonuç olarak; Veri setindeki gürültüyü en iyi temizleyen ve ensemble tekniklerinin (boosting) avantajını hiperparametre optimizasyonu ile birleřtiren XGBoost, projenin başarı kriterlerini (F1 %80+, AUC %85+) fazlasıyla karřılayarak en optimum çözüm olmuřtur.

Referanslar

[1] N. Antonio, A. de Almeida ve L. Nunes, "Predicting hotel booking cancellations to decrease uncertainty and increase revenue," *Tourism & Management Studies*, c. 13, s. 2, ss. 25-39, 2017. [Çevrimiçi]. Erişim:

<https://ieeexplore.ieee.org/abstract/document/8260781>

[2] A. J. Sánchez-Medina ve D. Ceballos-Hornero, "Hotel booking cancellation prediction using advanced Machine Learning techniques," *Decision Support Systems*, c. 170, 2023. [Çevrimiçi]. Erişim:

<https://www.sciencedirect.com/science/article/abs/pii/S0167923623000349>

[3] Kaggle, "Hotel Reservations Classification Dataset," [Çevrimiçi]. Erişim:

<https://www.kaggle.com/datasets/ahsan81/hotel-reservations-classification-dataset>.

[4] Chen, T., & Guestrin, C. (2016). "XGBoost: A Scalable Tree Boosting System," *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

[5] Pedregosa, F., et al. (2011). "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, Vol. 12, pp. 2825-2830.

[6] Github linki: https://github.com/BeratCdc/Team100_VeriMadenciligi.git