Beşir Arslan

Üsküdar Üniversitesi

234312026

Makine Öğrenmesi Final Projesi

Dr.Öğr. Üyesi Gökalp TULUM

Makine Öğrenmesi Final Projesi: HIGGS Veri Seti Üzerinde Özellik Seçimi ve Hiperparametre Optimizasyonu

1. Projenin Amacı ve Giriş

Bu proje raporu, büyük ve çok özellikli bir veri seti olan HIGGS Dataset üzerinde gerçekleştirilen makine öğrenmesi sürecinin önemli bileşenleri olan **özellik seçimi (feature selection)** ve **hiperparametre ayarlaması (hyperparameter tuning)** adımlarını detaylandırmaktadır. Amacımız, bu adımları uygulayarak farklı makine öğrenmesi modellerinin performansını karşılaştırmak, en iyi performansı gösteren modeli ve veri temsilini belirlemek ve elde edilen sonuçları yorumlamaktır.

1.1 Kullanılan Veri Seti

Projede, yüksek enerji fiziği deneylerinden elde edilen parçacık çarpışmalarına ait 11 milyon örnek ve 28 özellik içeren **HIGGS Dataset** kullanılmıştır. Performans ve işlem süresi kısıtlamaları göz önünde bulundurularak, bu veri setinden rastgele **100.000 örnek** alınarak çalışma gerçekleştirilmiştir. Veri setine UCI Machine Learning Repository üzerinden erişilmiştir.

2. Veri Ön İşleme (Preprocessing)

Veri setinin modeller için uygun hale getirilmesi amacıyla aşağıdaki ön işleme adımları uygulanmıştır:

2.1 Aykırı Değer Analizi

- **Uygulanan Yöntem:** Aykırı değerlerin tespiti için **IQR (Interquartile Range) yöntemi** kullanılmıştır. Her bir sayısal özellik için Q1 (ilk çeyrek) ve Q3 (üçüncü çeyrek) değerleri hesaplanmış, aykırı değerler $Q1-1.5\times IQR$ veya $Q3+1.5\times IQR$ dışında kalan noktalar olarak belirlenmiştir.

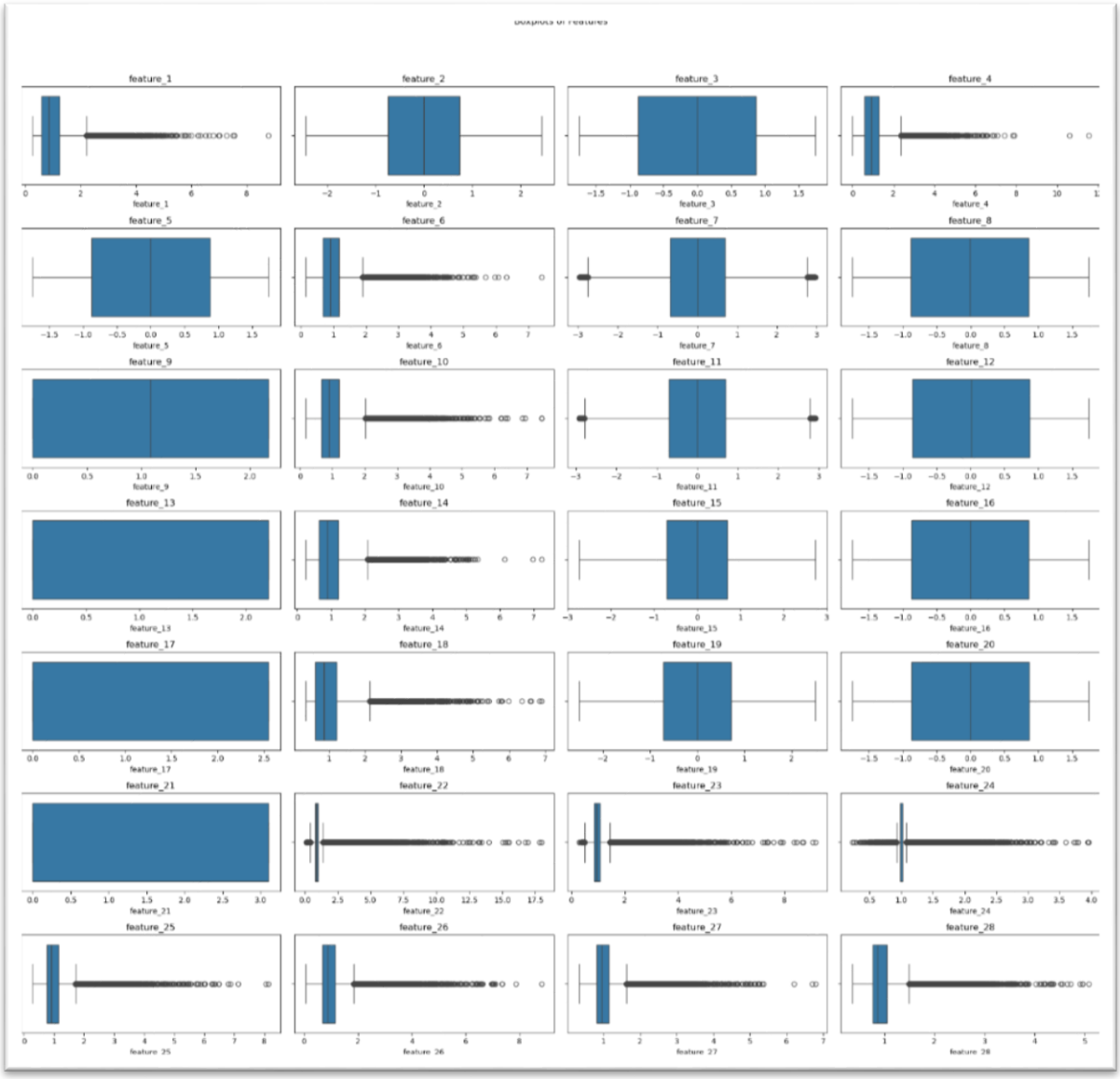


Fig 1. Kutu Grafik Aykırı Değerlerin Görselleştirilmesi

- **Tespit Edilen Aykırı Değerler:**

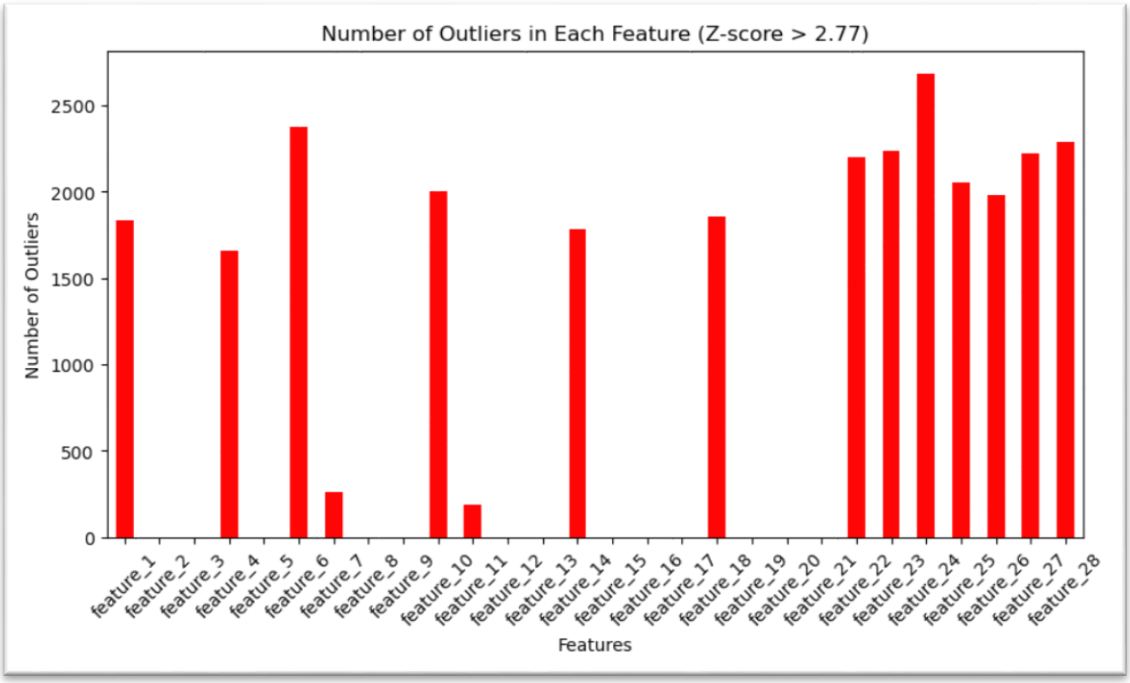


Fig 2. Aykırı Değerlerin Sayıları

- **Uygulanan İşlem:**
    - Tespit edilen aykırı değerlerin veri setinin genel dağılımını bozmasını engellemek ve model performansını stabilize etmek amacıyla **Z-skoru** hesaplanmış ve ardından **Winsorizing tekniği** uygulanmıştır. Bu yöntemle, belirli bir Z-skoru eşiğinin dışındaki tüm aykırı değerler, veri setinin kabul edilebilir alt ve üst sınırlarına (genellikle belirli yüzdelik dilimlere) baskılanmıştır. Bu yaklaşım, aşırı değerlerin etkisini azaltırken veri kaybını önlemiş ve veri dağılımını daha tutarlı hale getirmiştir

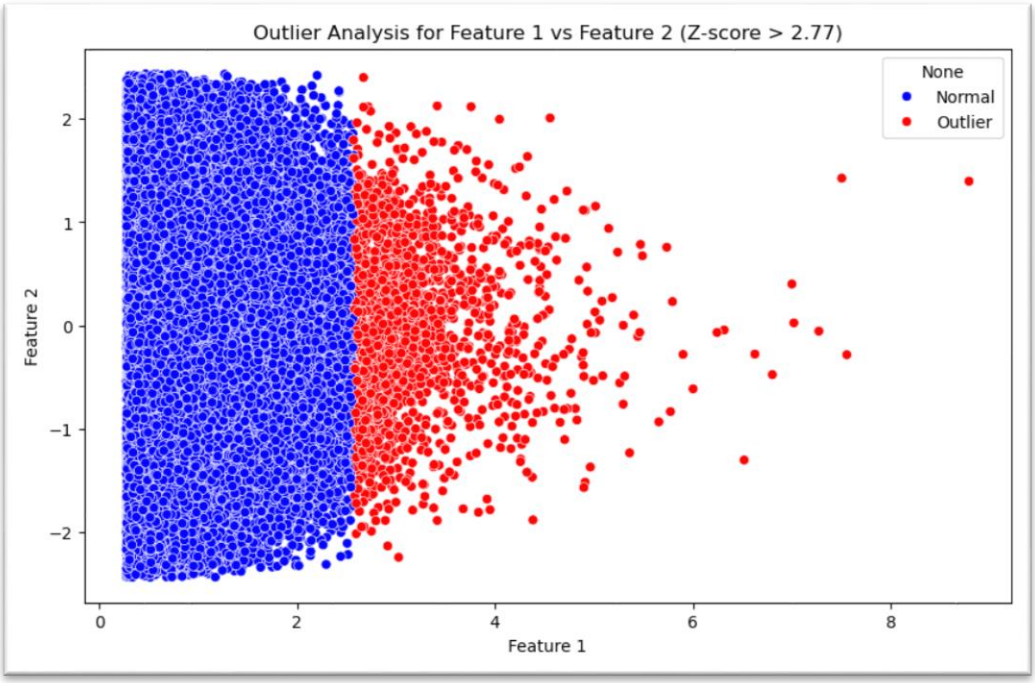


Fig 3. Aykırı değer analizi Z skor tabanlı

- **Görselleştirmeler:**
    - Feature_1 ve feature_2 kolonları için baskılama işlemi sonrası 2.7 olarak belirlenen eşik noktasında birikme gözlenmiştir.

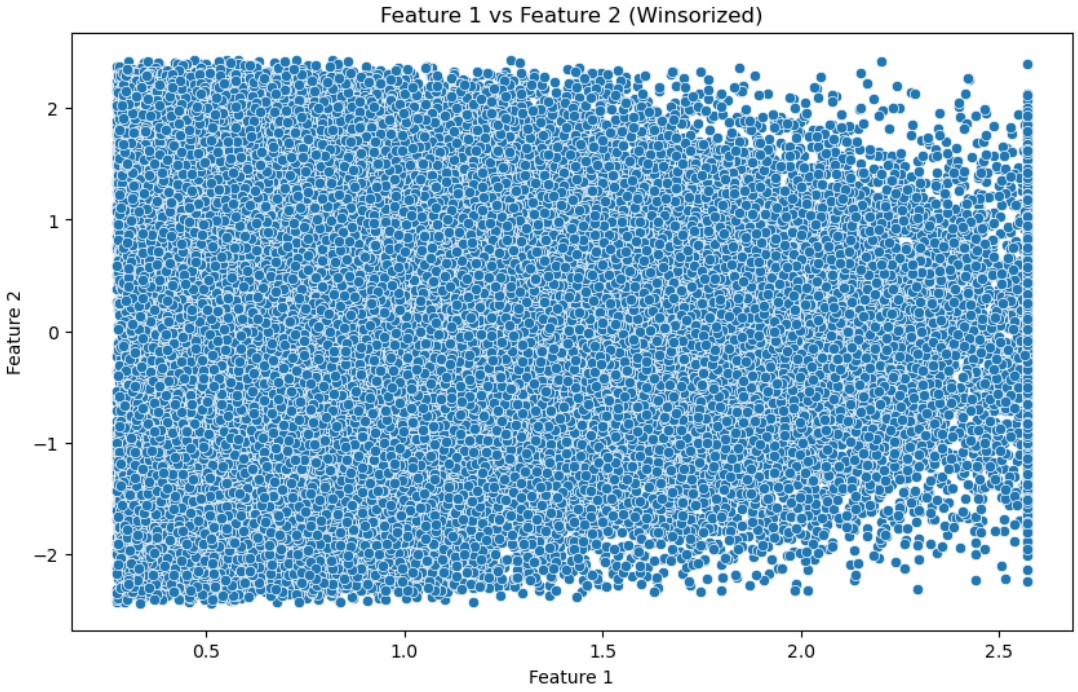


Fig 4. Baskılama İşleminden Sonra

2.2 Özellik Ölçekleme

- **Uygulanan Yöntem:** Tüm sayısal değişkenler, model performansını artırmak ve bazı algoritmaların (örn. KNN, SVM) mesafe tabanlı hesaplamalarından etkilenmemesini sağlamak amacıyla **MinMaxScaler** kullanılarak [0,1] aralığına dönüştürülmüştür.
- **Gerekçe:** Özelliklerin farklı ölçeklerde olması model eğitimini olumsuz etkileyeceği için ölçeklenmiştir.

3. Özellik Seçimi (Feature Selection)

Model performansını artırmak, aşırı uydurmayı (overfitting) azaltmak ve eğitim süresini kısaltmak amacıyla özellik seçimi yapılmıştır.

3.1 Filter-Based Özellik Seçimi

- **Uygulanan Yöntem:** Özellik seçimi için **Filter-Based yöntemlerden [ANOVA F-score / Mutual Information (kullandığınızı belirtin)]** kullanılmıştır.
- **Seçim Kriteri:** Bu yöntemle veri setindeki en iyi **15 özellik** seçilmiştir.
- **Seçilen Özellikler:**
    - Feature1, feature4, feature6, feature10, feature13, feature17, feature18, feature21, feature22, feature23, feature24, feature25, feature26, feature27
- **Yorum:** Bu hızlı ve modelden bağımsız yöntem, HIGGS veri setinde işlem yükünü azaltırken, hedef değişkenle en güçlü istatistiksel ilişkiye sahip özellikleri seçerek model performansını artırmayı ve aşırı uydurmayı azaltmayı hedeflemiştir.

4. Modelleme ve Değerlendirme

Bu bölümde, seçilen özellikler ve veri setleri üzerinde farklı makine öğrenmesi modelleri eğitilmiş ve Nested Cross-Validation yaklaşımıyla değerlendirilmiştir.

4.1 Nested Cross-Validation (İç İçe Çapraz Doğrulama)

Nested Cross-Validation, model seçimi ve hiperparametre ayarını daha sağlam ve tarafsız bir şekilde yapmak için kullanılan güçlü bir yöntemdir.

- **Outer Loop:** 5-fold çapraz doğrulama kullanılmıştır. Bu döngü, modelin genelleme performansını değerlendirmek için kullanılır.
- **Inner Loop:** 3-fold çapraz doğrulama kullanılmıştır. Bu döngü, en iyi hiperparametrelerin ve/veya özellik kombinasyonlarının seçimi için kullanılır.

4.1.1 Flowchart A: İç Döngüde Özellik Seçimi Kombinasyonları

- Elde edilen en iyi öznitelik setini ve bu kararı destekleyen metrikler;

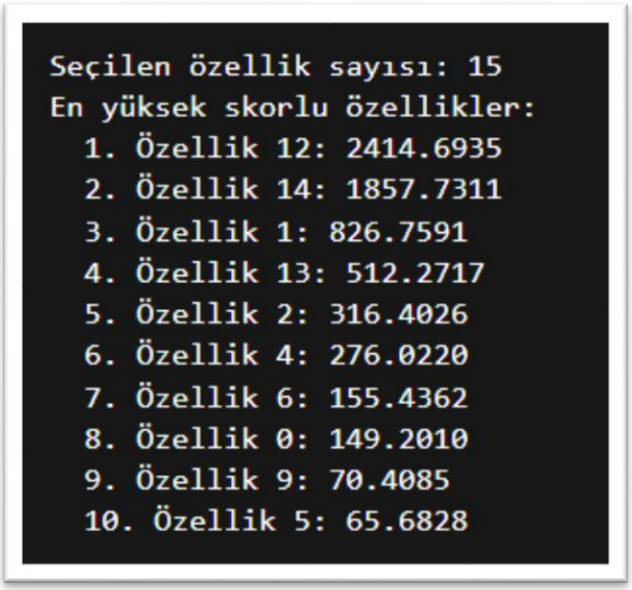


Fig.4 En yüksek Skorlu Özellikler

4.1.2 Flowchart B: İç Döngüde Hiperparametre Optimizasyonu

- İç döngüde, her bir model için belirli hiperparametre aralıkları üzerinde GridSearchCV kullanılarak en iyi hiperparametre kombinasyonları aranmıştır.
- **Kullanılan Hiperparametre Aralıkları:**
    - **K-Nearest Neighbors (KNN):** n_neighbors = [3, 5, 7, 9, 11] (veya denediğiniz diğer değerler)
    - **Support Vector Machine (SVM):** C = [0.1, 1], kernel = ['linear']

      NOT: SVM için Kernel ‘rbf’ denenmiş ancak çok uzun bekleme süresi göz önünde bulundurularak kombinasyona dahil edilmemiştir.
    - **Multi-Layer Perceptron (MLP):** hidden_layer_sizes = [(50,), (100,)], activation = ['relu', 'tanh']
    - **XGBoost:** n_estimators = [50, 100], max_depth = [3, 5]
- **Optimizasyon Stratejisi:** Tüm metrikler baz alınarak en iyi hiperparametreler

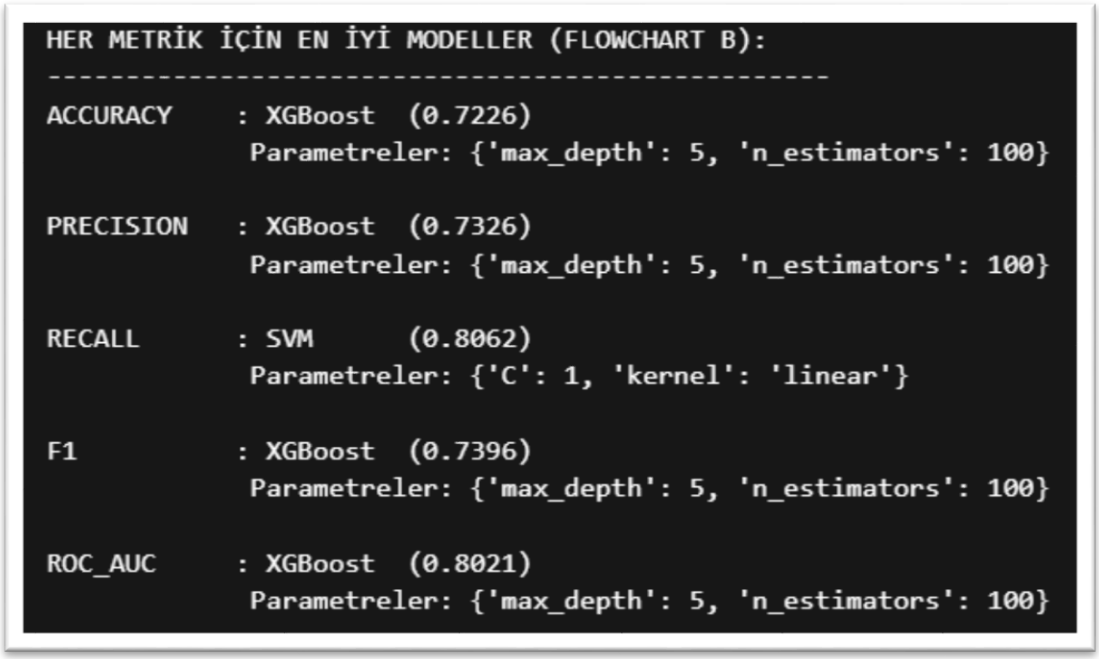


Fig 5. En İyi Modeller ve Hiperparametre Değerleri

4.2 Kullanılan Modeller ve Performans Karşılaştırması

Aşağıdaki makine öğrenmesi algoritmaları kullanılarak modeller eğitilmiş ve performansları karşılaştırılmıştır. Her bir model için Nested Cross-Validation sonucunda elde edilen ortalama metrikler ve en iyi hiperparametreler rapor edilmiştir.

4.2.1 K-Nearest Neighbors (KNN)

- **En İyi Hiperparametreler:** n_neighbors = 7
- **Performans Metrikleri (Ortalama Test Skoru):**
    - **Tablo:**

Metrik	**Değer**
Accuracy	%64
Precision	%66
Recall	%66

F1 Score	%66
ROC-AUC	%72

- **Yorum:** KNN algoritması için yapılan hiperparametre optimizasyonunda en iyi sonuç n_neighbors=7 ile elde edilmiştir; bu değer, modelin aşırı öğrenmeyi önleyerek daha dengeli ve kararlı bir performans göstermesini sağlamıştır. Ancak KNN, özellikle büyük ve yüksek boyutlu veri kümelerinde hesaplama maliyeti yüksek olabileceği için dikkatli kullanılmalıdır.

4.2.2 Support Vector Machine (SVM)

- **En İyi Hiperparametreler:** C = 1, kernel = 'linear')
- **Performans Metrikleri (Ortalama Test Skoru):**
    - **Tablo:**

Metrik	**Değer**
Accuracy	%64
Precision	%65
Recall	%64
F1 Score	%63
ROC-AUC	%68

- **Yorum:** SVM modeli için yapılan hiperparametre optimizasyonunda C=1 ve linear kernel en iyi sonucu vermiş, bu ayarlar modelin hem doğruluğunu hem de genelleme yeteneğini artırmıştır. RBF kernel ise yüksek hesaplama maliyeti nedeniyle uygulanabilir bulunmamıştır.

4.2.3 Multi-Layer Perceptron (MLP)

- **En İyi Hiperparametreler:** hidden_layer_sizes = 100, activation = relu
- **Performans Metrikleri (Ortalama Test Skoru):**

- **Tablo:**

Metrik	Değer
Accuracy	%72
Precision	%72
Recall	%72
F1 Score	%72
ROC-AUC	%78

- **Yorum:** MLP modelinde hidden_layer_sizes=(100,) ve activation='relu' seçimi, modelin doğrusal olmayan karmaşık örüntüleri başarılı şekilde öğrenmesini sağlamıştır. Derin yapıdaki katman sayısı ve ReLU'nun gradyan kaybı problemini azaltması, modelin hem öğrenme kapasitesini artırmış hem de eğitim süresini verimli kılmıştır.

4.2.4 XGBoost

- **En İyi Hiperparametreler:** max_depth = 5, n_estimators = 100
- **Performans Metrikleri (Ortalama Test Skoru):**
  - **Tablo:**

Metrik	Değer
Accuracy	%72
Precision	%73
Recall	%74
F1 Score	%74

ROC-AUC	%80

- **Yorum:** XGBoost modeli için yapılan optimizasyonda n_estimators=100 ve max_depth=5 en iyi sonucu vermiş, bu yapı modelin karmaşık örüntüleri öğrenmesini sağlayarak doğruluk oranını artırmıştır. Daha derin ağaçlar ve yüksek estimator sayısı, modelin daha güçlü bir tahmin performansı sergileyebilirdi ancak hesaplama maliyeti göz önünde bulundurulduğu için modelin bu yapısı daha dengeli bir performans sergilemiştir.

4.3 ROC Eğrileri ve AUC Skorları

Her bir model için ROC (Receiver Operating Characteristic) eğrileri çizilmiş ve AUC (Area Under the Curve) skorları hesaplanmıştır. ROC eğrileri, sınıflandırma modellerinin farklı eşik değerlerindeki performansını görselleştirmek için kullanılırken, AUC skoru modelin genelleme yeteneğinin bir özetidir.

- **ROC Eğrisi Grafikleri:**

Tabloya göre, XGBoost modeli en yüksek AUC skoru olan %80 ile en iyi performansı göstermektedir. MLP %79 ile onu takip ederken, KNN ve SVM sırasıyla %71 ve %68 AUC skorlarıyla daha düşük performans sergilemiştir. Bu sonuçlar, XGBoost'un veri setindeki karmaşık ilişkileri daha başarılı yakaladığını ve genelleme yeteneğinin diğer modellere kıyasla daha yüksek olduğunu göstermektedir.

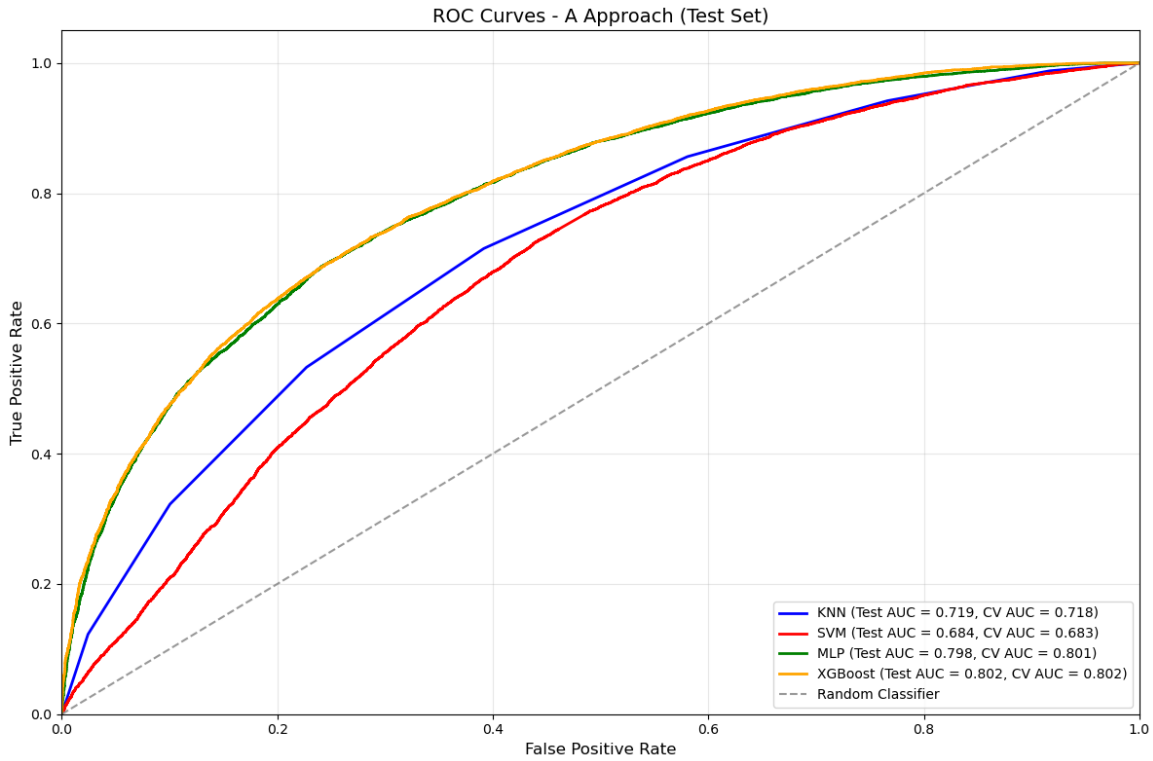


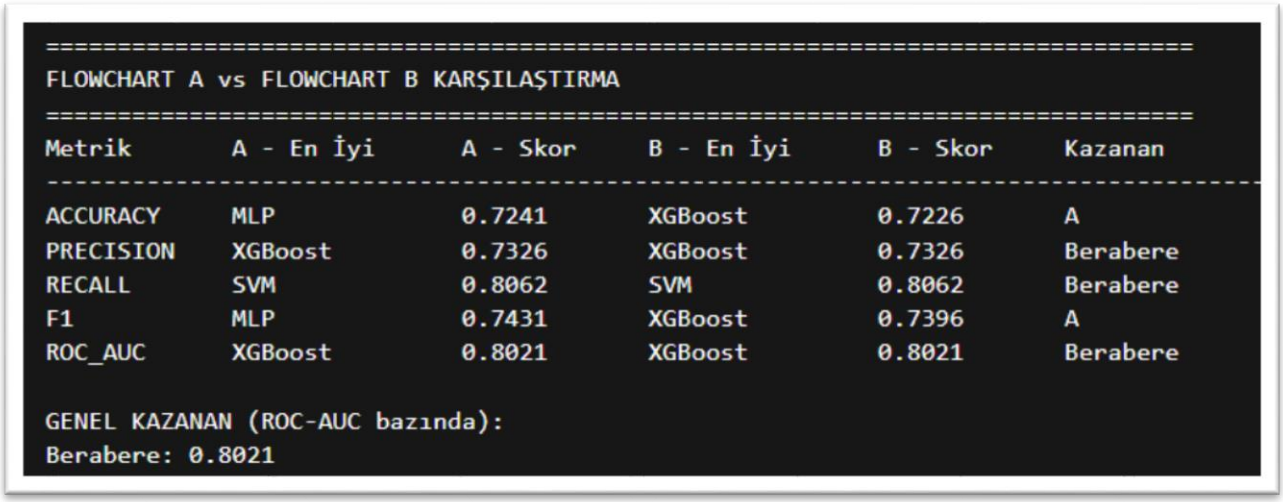
Fig 6. Tüm Modeller için ROC curve ve AUC skorları

5. Sonuçların Yorumlanması ve En Başarılı Model

Bu bölümde, elde edilen tüm sonuçlar bir araya getirilerek genel bir değerlendirme yapılacaktır.

5.1 Model Karşılaştırması ve Genel Değerlendirme

- Tüm modellerin performans metriklerini (Accuracy, Precision, Recall, F1-Score, ROC-AUC) içeren **özet bir tablo** oluşturuldu. Bu tablo, modeller arasındaki farkları daha net görmemizi sağlamaktadır.



```
================================================================================
FLOWCHART A vs FLOWCHART B KARŞILAŞTIRMA
================================================================================
Metrik       A - En İyi      A - Skor     B - En İyi      B - Skor     Kazanan
--------------------------------------------------------------------------------
ACCURACY     MLP             0.7241       XGBoost         0.7226       A
PRECISION    XGBoost         0.7326       XGBoost         0.7326       Berabere
RECALL       SVM             0.8062       SVM             0.8062       Berabere
F1           MLP             0.7431       XGBoost         0.7396       A
ROC_AUC      XGBoost         0.8021       XGBoost         0.8021       Berabere

GENEL KAZANAN (ROC-AUC bazında):
Berabere: 0.8021
```

Fig 7. FlowchartA ve FlowchartB karşılaştırma

- Tablo ve ROC eğrileri incelendiğinde, XGBoost modeli en yüksek ROC AUC skoruyla en iyi genel performansı sergilemiştir. Bu başarı, XGBoost'un gradyan artırma yöntemiyle karmaşık veri örüntülerini etkili şekilde yakalaması ve aşırı öğrenmeyi önleyici mekanizmaları sayesinde mümkün olmuştur. Ayrıca, veri setindeki değişkenlerin etkileşimlerini derinlemesine modelleyebilmesi XGBoost'u diğer modellere göre üstün kılmıştır.

5.2 En Başarılı Model ve Veri Temsili Kombinasyonunun Yorumu

- **En Başarılı Model:** Hiperparametreler max_depth: 5, n_estimators: 100
- **Veri Temsili (Özellik Seçimi Etkisi):** Seçilen 15 öznitelik, modelin gereksiz ve gürültülü verilerden arınarak daha odaklı ve anlamlı bilgilerle öğrenmesini sağlamıştır. Bu sayede modelin hem doğruluğu hem de genelleme kapasitesi artarken, aşırı öğrenme riski azalmıştır. Flowchart A'da farklı öznitelik kombinasyonlarının denemesi, hangi özelliklerin performansı olumlu etkilediğini belirlemeye yardımcı olmuş ve bu sayede model başarısında tutarlı sonuçlar elde edilmiştir.
- **Genel Çıkarımlar:** Bu projeden, doğru özellik seçimi ve etkili hiperparametre optimizasyonunun model performansını belirgin şekilde artırdığı önemli bir ders olarak

çıkarılmıştır. Özellik seçimi, gereksiz verilerin temizlenerek modelin daha hızlı ve doğru öğrenmesini sağlarken, hiperparametre optimizasyonu ise modelin en uygun ayarlarla çalışarak genelleme yeteneğini güçlendirmesine olanak tanımıştır. Dolayısıyla, makine öğrenmesi pipeline'ında bu iki aşama, başarılı ve sağlam modeller geliştirmek için kritik öneme sahiptir.

6. Sonuç ve Gelecek Çalışmalar

- Çalışmada veri ön işleme aşamasında kapsamlı EDA gerçekleştirilmiş, her özellik için aykırı değerler görselleştirilmiştir. Aykırı değerlerin etkisini azaltmak amacıyla Winsorizing yöntemi uygulanmış ve böylece uç değerlerin modele olumsuz etkisi minimize edilmiştir. Aykırı ve normal değer dağılımları ilgili değişkenler arasında karşılaştırmalı grafiklerle incelenmiştir.
- Veri seti çok büyük olması sebebiyle, hesaplama verimliliğini artırmak ve model eğitiminin sürdürülebilirliğini sağlamak amacıyla veri setinden rastgele seçilen 100.000 örnek kullanılmıştır. Modelin sağlıklı öğrenebilmesi için veriler MinMaxScaler ile ölçeklendirilmiş; bu sayede farklı ölçeklerdeki özellikler aynı aralıkta normalize edilerek algoritmanın performansı artırılmıştır. Ayrıca, modelin karmaşıklığını azaltmak ve gereksiz değişkenlerin etkisini en aza indirmek amacıyla filtre tabanlı yöntemlerle öznitelik seçimi yapılmıştır. Bu yöntemler, veri setindeki en anlamlı ve bilgi taşıyan özelliklerin belirlenmesini sağlayarak modelin hem hızını hem de doğruluğunu olumlu yönde etkilemiştir.
- Model değerlendirmesinde nested cross-validation uygulanmış; iç döngüde GridSearch kullanılarak hiperparametre optimizasyonu yapılmış ve dış döngüde modelin genel test performansı objektif şekilde ölçülmüştür.
- Flowchart A yaklaşımında, iç döngüde farklı öznitelik seçim kombinasyonları denenerek hem model başarısı hem de en anlamlı özellik seti belirlenmiştir. Flowchart B'de ise aynı iç döngüde hiperparametre kombinasyonları optimize edilmiştir. Bu iki yaklaşım, modelin hem özellik mühendisliği hem de parametre ayarları açısından en iyi performansa ulaşmasını sağlamıştır. Sonuçlar, modelin genelleme kapasitesinin yüksek olduğunu ve aşırı öğrenme riskinin minimize edildiğini göstermektedir.

Ekler

A. GitHub Linki

- https://github.com/besirarslann/Final_ML_Higgs_Project
- **Açıklama:** GitHub deposu, projenin tüm kodlarını, oluşturulan tüm grafik çıktılarını ve bu raporda sunulan yorumları içermektedir.
