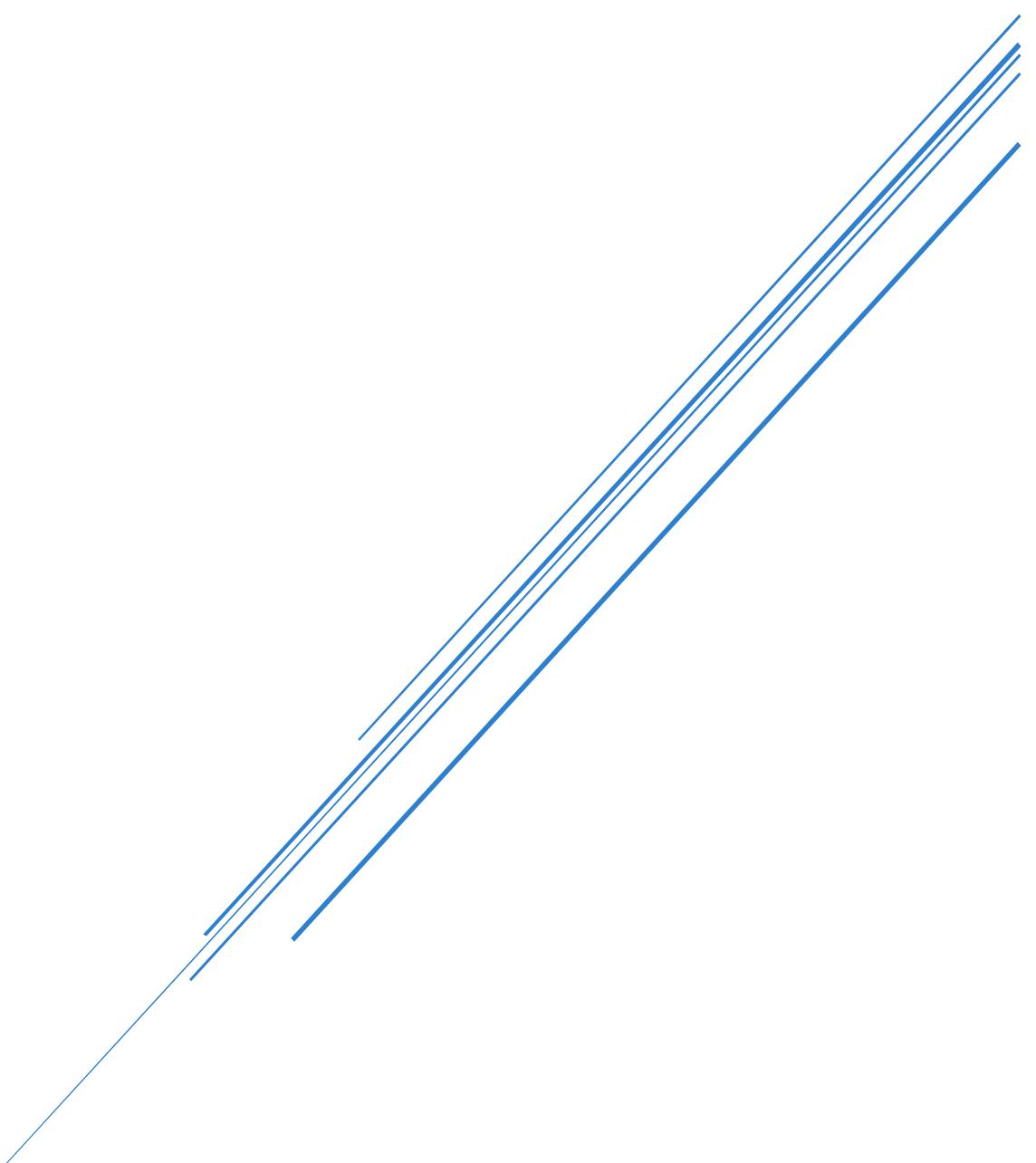


5ML



Esra Aydın
LOTUS AI



İçindekiler

1. COVID-19 Verisi Üzerinden Regresyon Uygulaması	3
1. Veri Seti	3
2. Görselleştirme ve Ön İşleme.....	3
3. Modelleme.....	4
4. Hiperparametre Optimizasyonu (GridSearchCV)	4
5. Yeni Veri Üzerinde Tahmin	4
6. H2O AutoML ile Regresyon	5
7. H2O AutoML ile Model Açıklamaları	5
8. Hiperparametre Optimizasyonu (GridSearchCV) with H2O	7
2. Meme Kanseri Verisi Üzerinde Sınıflandırma Modeli Geliştirme	7
1. Veri Seti	7
2. Eksik Değer Analizi	8
3. Label Encoding ve KNNImputer.....	8
4. Görselleştirme	8
5. Feature Encoding (Öznitelik Kodlama).....	9
6. Özellik Seçimi: Feature Importance (Önemli Değişkenler).....	9
7. Hedef Değişken Dağılımı (Sınıf Dengesizliği)	9
8. Dengesizlik Giderme: SMOTE Uygulaması.....	10
9. Modelleme ve Değerlendirme	10
10. Modeller	10
11. Model Performans Karşılaştırması	10
12. Grid Search ile En İyi Modelin Hiperparametre Ayarı.....	10
13. En İyi Modelin Test Verisi Üzerindeki Performansı	11
14. ROC Eğrisi ve AUC Skoru	11
3. Öğrenci Performansı Verisi Üzerinde Kümeleme Modeli Geliştirme	12
1. Veri Seti	12
2. Eksik Verilerin İncelenmesi ve Temizlenmesi	12
3. Sınav Hazırlık Etkisi: Boxplot Görselleştirmesi	12
4. Korelasyon Analizi	12
5. Notların Dağılımı: Histogram	13
6. Kategorik Değişkenlerin Sayısallaştırılması	13
7. Ölçeklendirme	14
8. Kümeleme (K-Means)	14
9.Kümelere Göre Ortalama Ders Başarıları	14
10. Hiyerarşik Kümeleme	15



11. DBSCAN Kümeleme	16
12. H2O AutoML KMeans Sonuçlarının İncelenmesi	16
4. Öğrenci Performansı Faktörü Verisi Üzerinde Birlikteklilik Kural Çıkarımı	17
1. Veri Seti.....	17
2. Eksik Veri İşlemleri.....	17
3.Temel Görselleştirmeler	17
3.Başarı Seviyesi Oluşturma	19
4.Apriori Birlikteklilik Analizi	19
5.Network Grafiği ile Kural Görselleştirmesi	20
6.Tüm Değişken Dönüşümleri	21
5. Öğrenci Performansı Verisi Üzerinde Anomali Tespiti	22
1.Veri Seti	22
2.Anomali Tespiti (Outlier Detection)	22
3.Isolation Forest ile Anomali Tespiti.....	22
4.Local Outlier Factor (LOF) ile Anomali Tespiti	23
5.Özel Durumlara Dayalı Anomali Analizi	24

1.COVID-19 Verisi Üzerinden Regresyon Uygulaması

1. Veri Seti

Bu çalışmada kullanılan veri seti, COVID-19 pandemisine ait günlük **vaka** ve **ölüm** sayılarını içermektedir. Veri seti, 720 günlük gözlemden oluşmaktadır ve şu değişkenleri içermektedir:

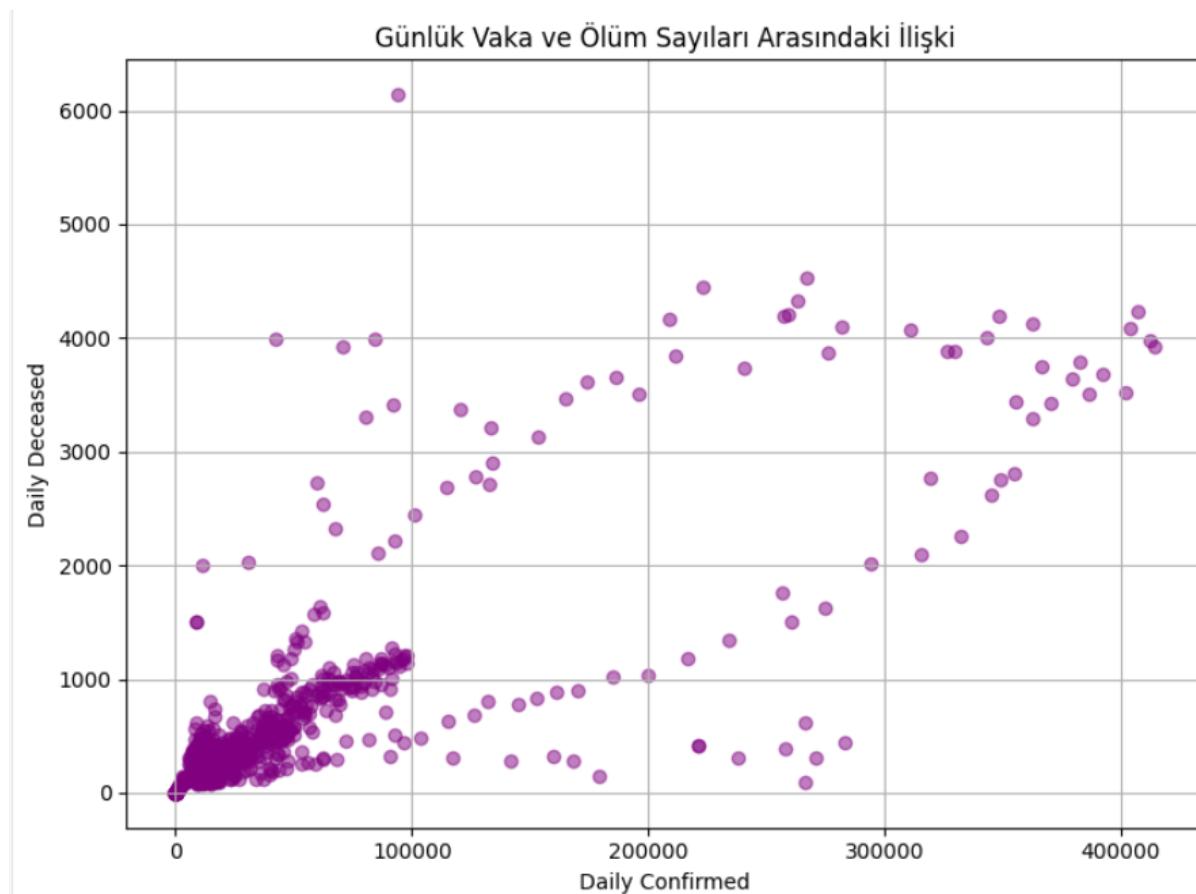
DEĞİŞKEN ADI	AÇIKLAMA
DATE	<i>Ham tarih bilgisi (örneğin: 30 January 2020)</i>
DATE_YMD	<i>ISO formatında tarih (YYYY-MM-DD)</i>
DAILY CONFIRMED	<i>O gün bildirilen yeni vaka sayısı</i>
DAILY DECEASED	<i>O gün bildirilen ölüm sayısı</i>

Veri setinde eksik değer (NA) bulunmamaktadır.

2. Görselleştirme ve Ön İşleme

Veri öncelikle zaman serisi analizi için işlenmiş ve aşağıdaki gibi görselleştirilmiştir:

Günlük Vaka ve Ölüm Sayıları:



Vaka ve Ölüm Sayısı Arasındaki İlişki:

Ardından, tarih sütunu kullanılarak gün, ay, yıl, hafta, hafta içi günü gibi ek zaman bileşenleri türetilmiş ve kümülatif toplamlar hesaplanmıştır.



Ayrıca, günlük ölüm oranı (Fatality Rate) aşağıdaki şekilde hesaplanmıştır:

$$\text{Fatality Rate} = \text{Daily Deceased} / (\text{Daily Confirmed} + \epsilon)$$

(ϵ : Sıfıra bölmeyi önlemek için küçük bir sayı)

3. Modelleme

Amaç:

Günlük vaka sayısı ve tarihsel bilgiler kullanılarak günlük ölüm sayısını (Daily Deceased) tahmin etmek.

Temel Modeller:

- Linear Regression
- K-Nearest Neighbors Regressor
- Random Forest Regressor
- XGBoost Regressor

Model performansları MAE, RMSE ve R² metrikleri ile değerlendirilmiştir.

Linear Regression Metrics:

MAE = 256.5171
RMSE = 509.2809
R² = 0.6225

Random Forest Metrics:

MAE = 99.0928
RMSE = 210.7610
R² = 0.9354

KNN Regressor Metrics:

MAE = 109.1111
RMSE = 230.1486
R² = 0.9229

XGBoost Metrics:

MAE = 113.0018
RMSE = 264.1253
R² = 0.8985

4. Hiperparametre Optimizasyonu (GridSearchCV)

Random Forest modeli için Grid Search kullanılarak en iyi parametre kombinasyonu belirlenmiş ve modelin doğruluk oranı artırılmıştır.

```
Fitting 3 folds for each of 72 candidates, totalling 216 fits
En iyi parametreler: {'max_depth': 10, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 5, 'n_estimators': 200}
Optimize Edilmiş Random Forest Performans:
MAE: 47.5679
RMSE: 118.3365
R2: 0.9839
```

5. Yeni Veri Üzerinde Tahmin

Aşağıdaki gibi yeni bir veri girildiğinde model bu verilere karşılık günlük ölüm sayısı tahminini yapabilmektedir:

Daily Confirmed Cumulative Confirmed Day Month

5000 150000 10 7

Daily Confirmed Cumulative Confirmed Day Month

12000 160000 15 7

Yeni Veri İçin Tahminler (Daily Deceased):
[250.02618921 321.58339805]

Sonuç: 1

ilk satırda veri için model, yaklaşık 250 kişi günlük ölüm bekliyor. ikinci satırda veri için model, yaklaşık 322 kişi günlük ölüm bekliyor. Ne anlama gelir? Örneğin, 10 Temmuz'da 5000 günlük vaka ve toplam 150000 vaka olduğunda, model günlük ölüm sayısını yaklaşık 250 kişi olarak tahmin ediyor. 15 Temmuz'da ise vaka sayıları arttığı için (12000 günlük vaka ve 160000 toplam vaka) model, günlük ölüm sayısını daha yüksek, 322 kişi olarak tahmin ediyor.

6. H2O AutoML ile Regresyon

Ayrıca, H2O AutoML kullanılarak tüm regresyon algoritmaları otomatik şekilde denenmiş ve en iyi model belirlenmiştir.

Model Performans Metrikleri (Test Verisi):

MSE : 90202.3517

RMSE : 300.3371

MAE : 96.3923

R^2 : 0.8970

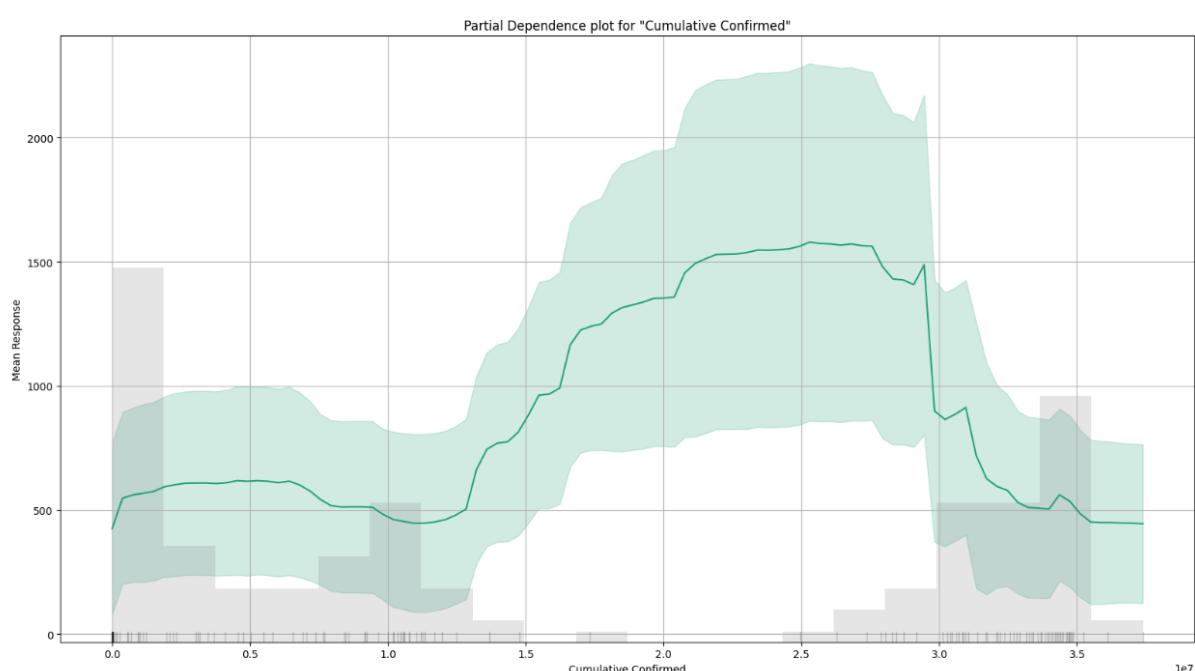
7. H2O AutoML ile Model Açıklamaları

1. Partial Dependence Plot (Özelliklerin Ortalama Etkisi)

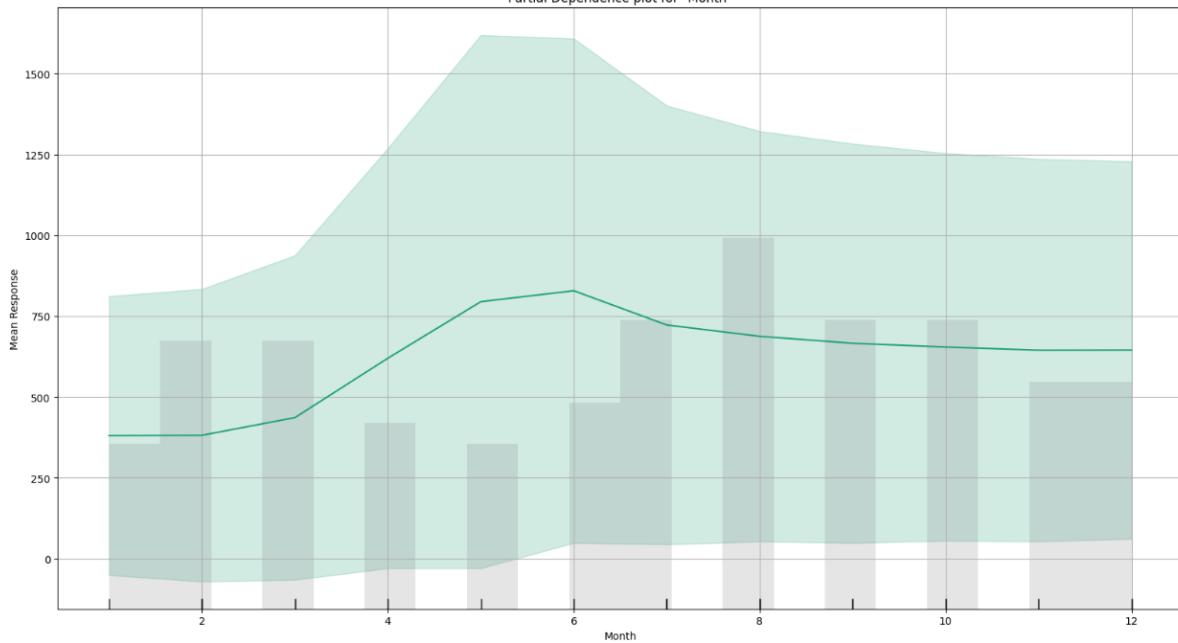
Partial Dependence Plot (PDP), bir özelliğin değerindeki değişim modelin tahmini üzerindeki ortalama etkisini gösterir. Diğer değişkenler sabit tutulurken, belirli bir değişkenin arttıkça ya da azaldıkça model çıkışının (örneğin Daily Deceased) nasıl değiştiği gözlemlenir.

Örneğin, Cumulative Confirmed değeri arttıkça modelin ölüm sayısı tahmininde de artış olması beklenebilir.

Yorumlama: Değişkenin hedef değişkene etkisi pozitif mi, negatif mi? Doğrusal mı, karmaşık mı?



Partial Dependence plot for "Month"

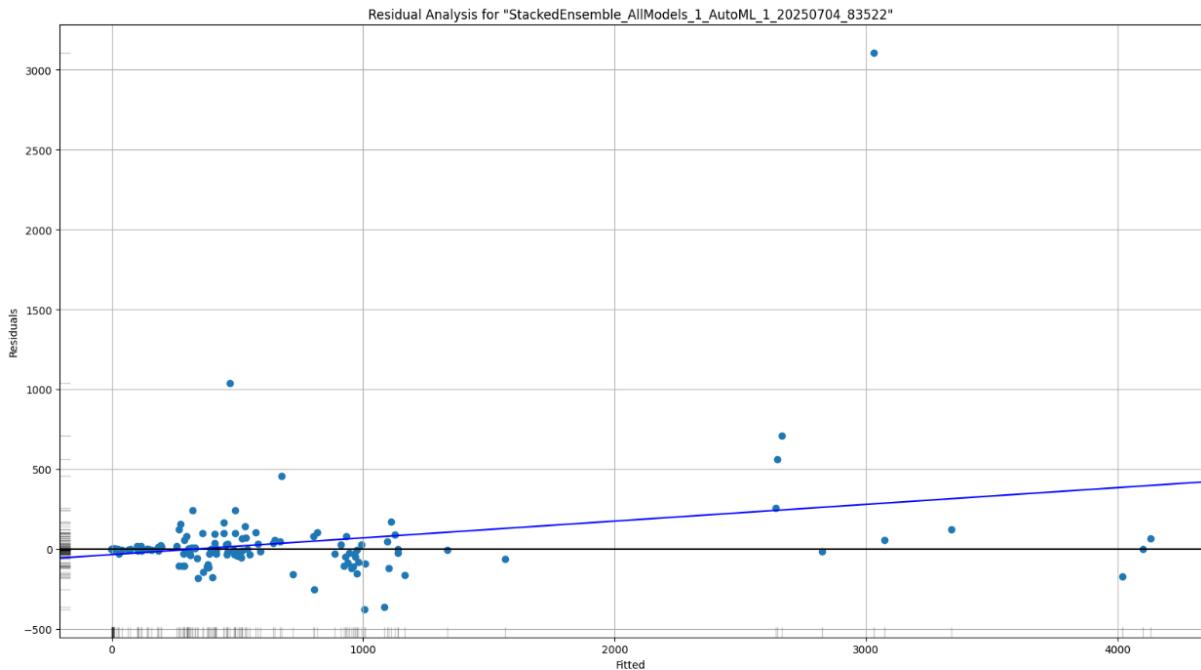


2. Residual Analysis (Model Hataları)

Residual Analysis, modelin tahmin ettiği değerlerle gerçek değerler arasındaki farkı (hataları) gösterir. Bu farkların dağılımı incelenerek modelin tutarlılığı, aykırı değer hassasiyeti ve genel doğruluğu hakkında fikir edinilir.

Hataların rastgele dağılması, modelin veri üzerinde sistematik bir hata yapmadığını gösterir. Aksi halde model belirli bölgelerde aşırı veya yetersiz tahmin yapıyor olabilir.

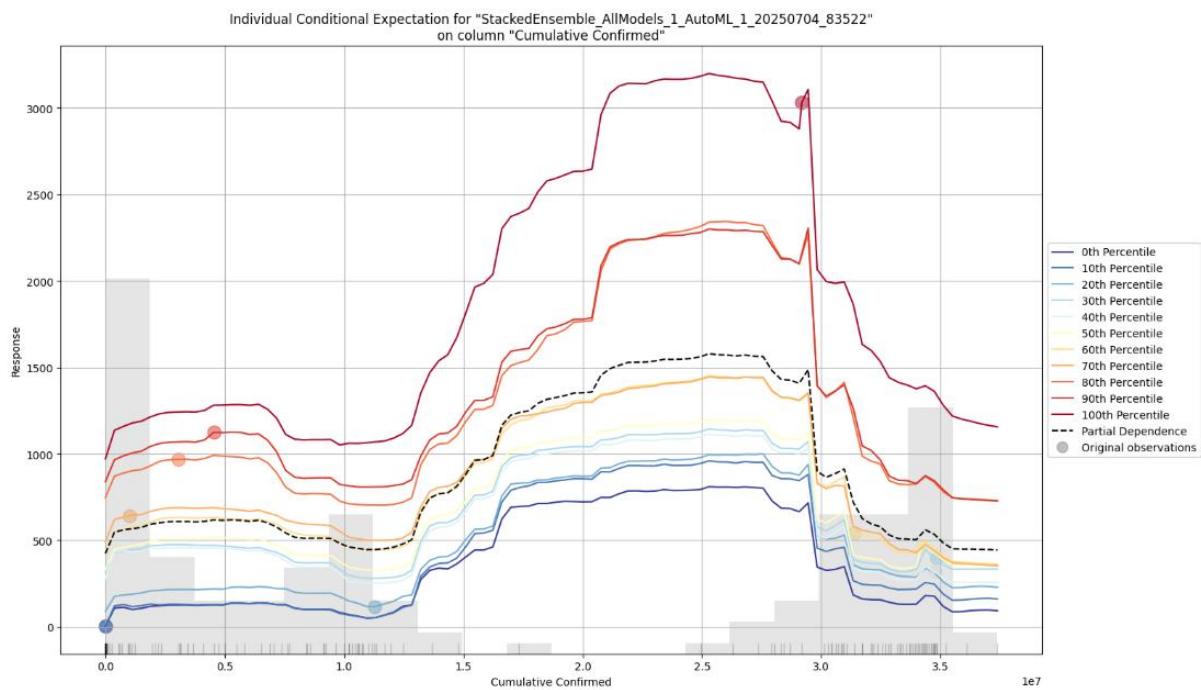
Yorumlama: İyi bir modelde artıklar (residuals) sıfır etrafında dengeli dağılmalıdır.



3. Individual Conditional Expectation (ICE)

ICE grafiği, PDP'nin bireysel veri noktalarına uygulanmış hâlidir. Her bir satır için bir çizgi çizilir, böylece değişkenin etkisi örnek bazında gözlemlenebilir. PDP'nin genelleştirilmiş bir ortalamasını verirken, ICE daha detaylı ve bireysel bir analiz sunar.

Yorumlama: ICE grafiği, değişkenin bazı bireysel veri noktalarında farklı etkiler yarattığını gösterebilir. Farklı yönlerde hareket eden çizgiler, modelin heterojen tepkiler verdiği gösterebilir.



8. Hiperparametre Optimizasyonu (GridSearchCV) with H2O

Python'da scikit-learn kütüphanesinde kullanılan GridSearchCV, hiperparametrelerin tüm olası kombinasyonlarını deneyerek en iyi sonucu veren yapılandırmayı bulur. Aynı mantık, H2O AutoML platformunda da uygulanabilir. H2O'nun sunduğu H2OGridSearch sınıfı ile, kullanıcı belirli parametre aralıklarını tanımlayarak otomatik hiperparametre taraması yapabilir.

MSE : 86965.1311
RMSE : 294.8985
MAE : 105.4004
R² : 0.9007

2. Meme Kanseri Verisi Üzerinde Sınıflandırma Modeli Geliştirme

1. Veri Seti

- Veri kümesinde 341 gözlem ve 16 özellik (sütun) bulunmaktadır.
- Sayısal değişkenler: Age, Protein1, Protein2, Protein3, Protein4
- Kategorik değişkenler: Gender, Tumour_Stage, Histology, ER status, PR status, HER2 status, Surgery_type, Patient_Status

- Tarihsel değişkenler: Date_of_Surgery, Date_of_Last_Visit
- Kimlik bilgisi: Patient_ID

2. Eksik Değer Analizi

- Tüm değişkenlerde 7 eksik değer bulunmaktadır, Date_of_Last_Visit için 24, Patient_Status için 20 eksik değer vardır.
- Eksik veriler, değişkenin tipi dikkate alınarak farklı stratejilerle doldurulmuştur:
 - Sayısal değişkenler → medyan
 - Kategorik değişkenler → mod (en sık değer)
 - Tarihsel değişkenler → ileri doldurma (forward fill)

Eksik Verilerin İyileştirilmesi (İleri Teknikler)

3. Label Encoding ve KNNImputer

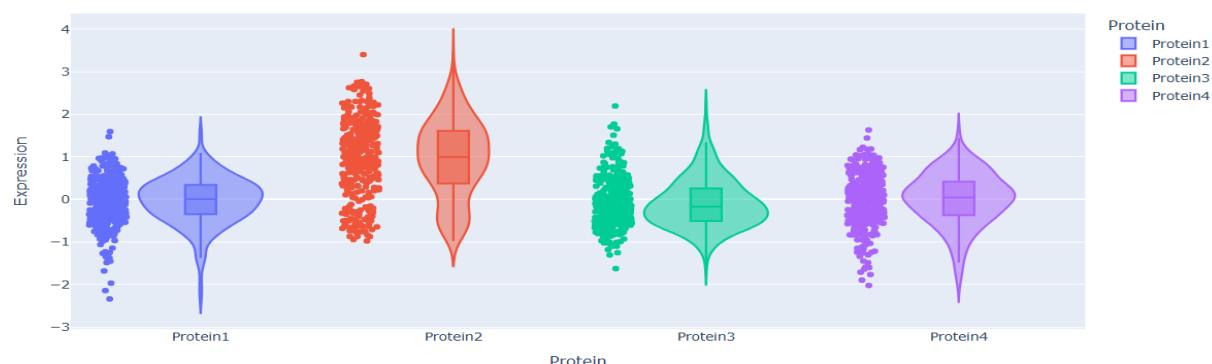
- Patient_Status değişkeni Alive → 0, Dead → 1 olarak kodlanmış ve eksik değerler KNNImputer (k-en yakın komşu) yöntemiyle doldurulmuştur.
- Böylece eksik sınıf bilgileri diğer özellikler dikkate alınarak tahmin edilmiştir.

4. Görselleştirme

Protein İfade Düzeylerinin Görselleştirilmesi

- Protein1–Protein4 sütunları, analiz ve görselleştirme için uzun formata dönüştürülmüştür (melt fonksiyonu ile).
- Plotly kütüphanesi kullanılarak violin plot oluşturulmuştur.
- Violin plot, proteinlerin ifade düzeylerinin dağılımını, yoğunluğunu ve olası uç değerlerini göstermektedir.
- Her protein için farklı renkler atanarak görselleştirme okunabilirliği artırılmıştır.
- Bu grafik, proteinin tüm örneklerdeki ifade varyasyonunu analiz etmek ve karşılaştırmak için kullanılmıştır.

Protein İfade Düzeylerinin Violin Plot'u (Plotly ile)



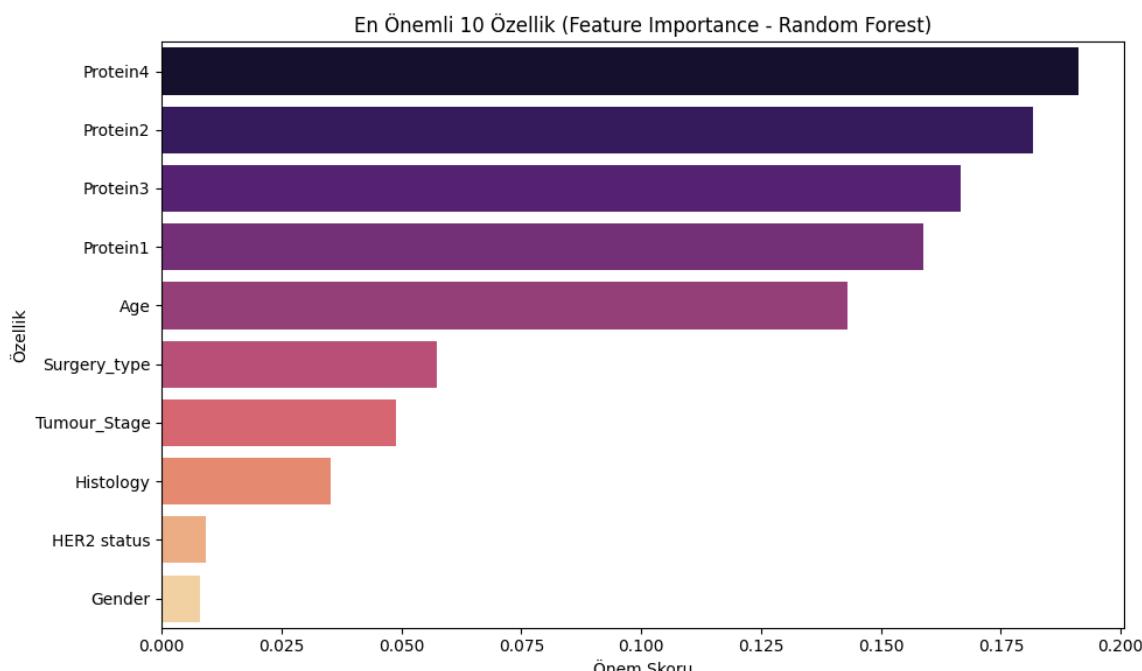
5. Feature Encoding (Öznitelik Kodlama)

- Kategorik değişkenler LabelEncoder kullanılarak sayısal hale getirilmiştir.
- Modelleme öncesi one-hot encoding yapılmıştır (özellikle X için).

6. Özellik Seçimi: Feature Importance (Önemli Değişkenler)

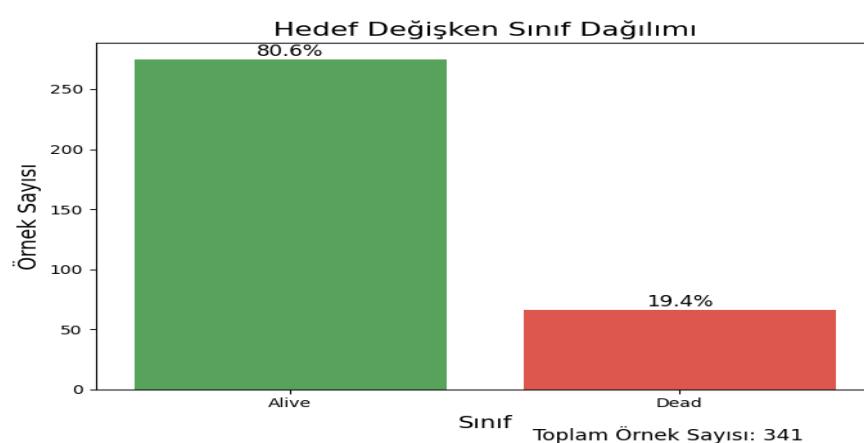
Random Forest ile Feature Importance

- Random Forest modeli eğitilerek değişken önem skorları hesaplanmıştır.
- Görselleştirme ile en önemli 10 değişken belirlenmiştir.
- Sıfırdan büyük öneme sahip değişkenler modelleme için seçilmiştir.



7. Hedef Değişken Dağılımı (Sınıf Dengesizliği)

- Hedef değişken (Patient_Status) dengesizdir:
 - Alive sınıfı çoğuluktadır.
- Sınıf dağılımı grafikle görselleştirilmiştir.



8. Dengesizlik Giderme: SMOTE Uygulaması

- Eğitim verisine SMOTE (Synthetic Minority Oversampling Technique) uygulanarak azınlık sınıfı örnekleri sentetik olarak üretilmiştir.
- Dengeli eğitim verisi elde edilmiştir.

9. Modelleme ve Değerlendirme

Değerlendirme Metrikleri

- Kullanılan metrikler: Accuracy, Precision, Recall, F1-score
- 10 katlı Stratified K-Fold cross-validation uygulanmıştır.

10. Modeller

Her model için üç varyasyon denenmiştir:

- Decision Tree (DT)
- K-Nearest Neighbors (KNN)
- Random Forest (RF)
- Support Vector Classifier (SVC)
- Logistic Regression (LR)
- Gaussian Naive Bayes (GNB)

11. Model Performans Karşılaştırması

- Tüm modellerin ortalama doğruluk, precision, recall ve f1 skorları karşılaştırılmıştır.
- En iyi performans gösteren model seçilmiştir.

10 Fold Cross-Validation Performance of Model Variants:

	Model	CV_Accuracy	CV_Precision	CV_Recall	CV_F1	Training_Time \
0	RF_v3	0.876956	0.881995	0.876948	0.876452	0.139729
1	RF_v2	0.837949	0.841042	0.837987	0.837531	0.096262
2	RF_v1	0.831131	0.836498	0.831494	0.830480	0.048702
3	SVC_v1	0.794239	0.801982	0.794156	0.792590	0.018581
4	SVC_v3	0.785307	0.809000	0.785173	0.779730	0.014873
5	DT_v1	0.773943	0.778816	0.774026	0.773031	0.007757
6	KNN_v2	0.737315	0.752156	0.737229	0.732943	0.004131
7	KNN_v1	0.730497	0.775471	0.730519	0.717248	0.004212
8	KNN_v3	0.684567	0.715464	0.684307	0.668123	0.003579
9	DT_v3	0.646089	0.657517	0.646212	0.638097	0.007653
10	LR_v3	0.620666	0.625030	0.620238	0.615618	0.004865
11	LR_v2	0.620666	0.625030	0.620238	0.615618	0.004749
12	LR_v1	0.604704	0.606757	0.604329	0.599896	0.004966
13	SVC_v2	0.604545	0.610558	0.604221	0.596075	0.025064
14	GNB_v2	0.593288	0.595541	0.592857	0.583622	0.003574
15	GNB_v1	0.593288	0.595541	0.592857	0.583622	0.003314
16	GNB_v3	0.593288	0.595541	0.592857	0.583622	0.003113
17	DT_v2	0.572992	0.588849	0.573052	0.546757	0.006170

12. Grid Search ile En İyi Modelin Hiperparametre Ayarı

- En iyi modelin parametreleri GridSearchCV ile optimize edilmiştir.
- Kullanılan skor fonksiyonu: macro F1-score

13. En İyi Modelin Test Verisi Üzerindeki Performansı

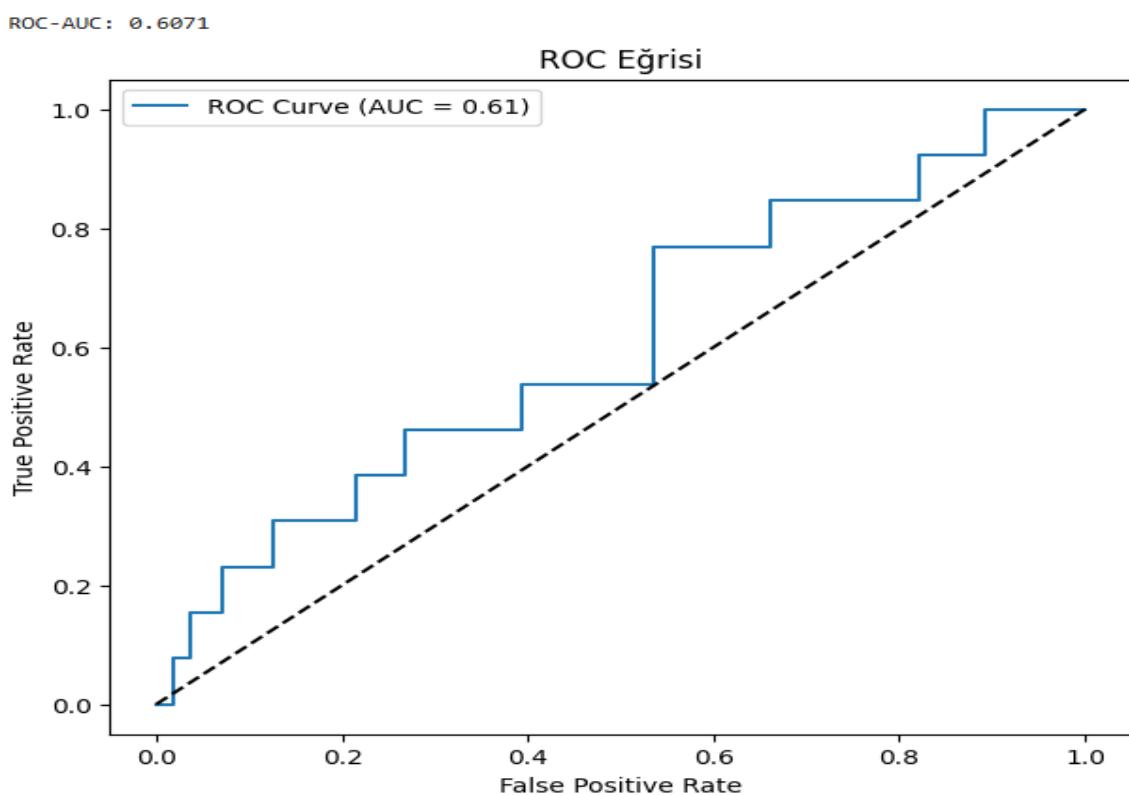
- Accuracy, Precision, Recall, F1-score gibi test metrikleri hesaplanmıştır.
- Confusion Matrix görselleştirilmiştir.

Best model based on CV F1: RF_v3

```
Performing Grid Search for Hyperparameter Optimization on the Best Classification Model...
Best Grid Search Parameters: {'max_depth': 15, 'min_samples_split': 4, 'n_estimators': 200}
Best CV Accuracy from Grid Search: 0.8992787461957903
```

14. ROC Eğrisi ve AUC Skoru

- Test verisi üzerinde performansı
- Modelin sınıflandırma başarısı ROC eğrisiyle analiz edilmiştir.
- ROC AUC değeri: modelin doğruluk-kararlılığı dengesini ölçer.



	precision	recall	f1-score	support
0.0	1.00	0.07	0.13	56
1.0	0.20	1.00	0.33	13
accuracy			0.25	69
macro avg	0.60	0.54	0.23	69
weighted avg	0.85	0.25	0.17	69

3. Öğrenci Performansı Verisi Üzerinde Kümeleme Modeli Geliştirme

1. Veri Seti

- Veri kümesi 12 sütundan ve 10.000 gözlemden oluşmaktadır.
- Değişkenler arasında öğrenci numarası (roll_no), cinsiyet (gender), etnik grup (race_ethnicity), ebeveyn eğitim seviyesi (parental_level_of_education), çeşitli ders notları, toplam not ve harf notu (grade) gibi bilgiler yer almaktadır.

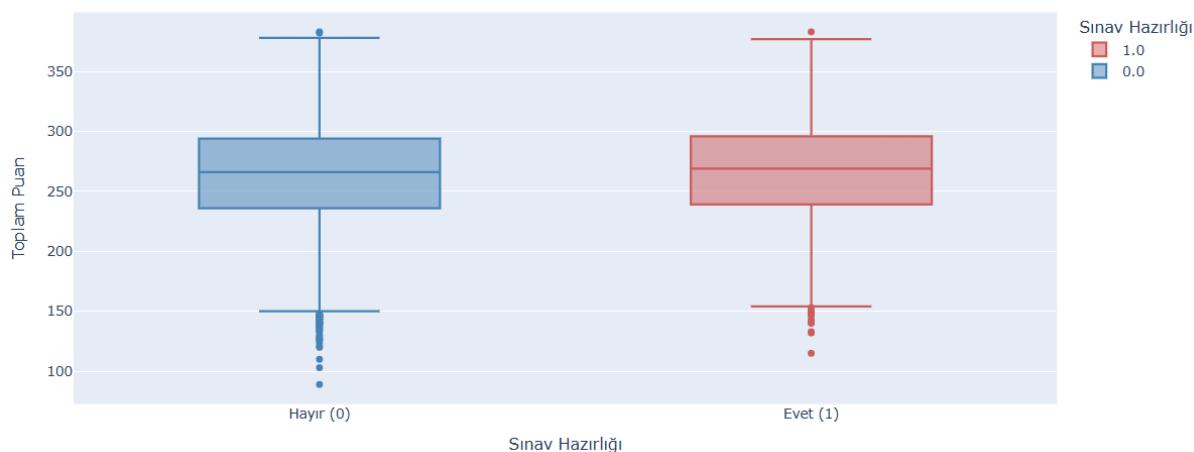
2. Eksik Verilerin İncelenmesi ve Temizlenmesi

- roll_no sütunu benzersiz tanımlayıcı olduğu için eksik olan satırlar doğrudan düşürülmüştür.
- Kategorik değişkenlerdeki eksik değerler mod (en sık tekrar eden değer) ile doldurulmuştur.
- Sayısal değişkenlerdeki eksikler ise ortalama değer ile doldurularak tamamlanmıştır.
- math_score sütunu bazı hatalı girişler içeriği için sayısal türe çevrilmiştir (to_numeric).

3. Sınav Hazırlık Etkisi: Boxplot Görselleştirmesi

- Öğrencilerin sınav hazırlık kursuna katıldı katılmama durumlarına göre toplam puan dağılımları boxplot ile görselleştirilmiştir.
- Bu grafik, kursa katılan ve katılmayan öğrencilerin başarı durumlarının karşılaştırılmasına olanak tanır.

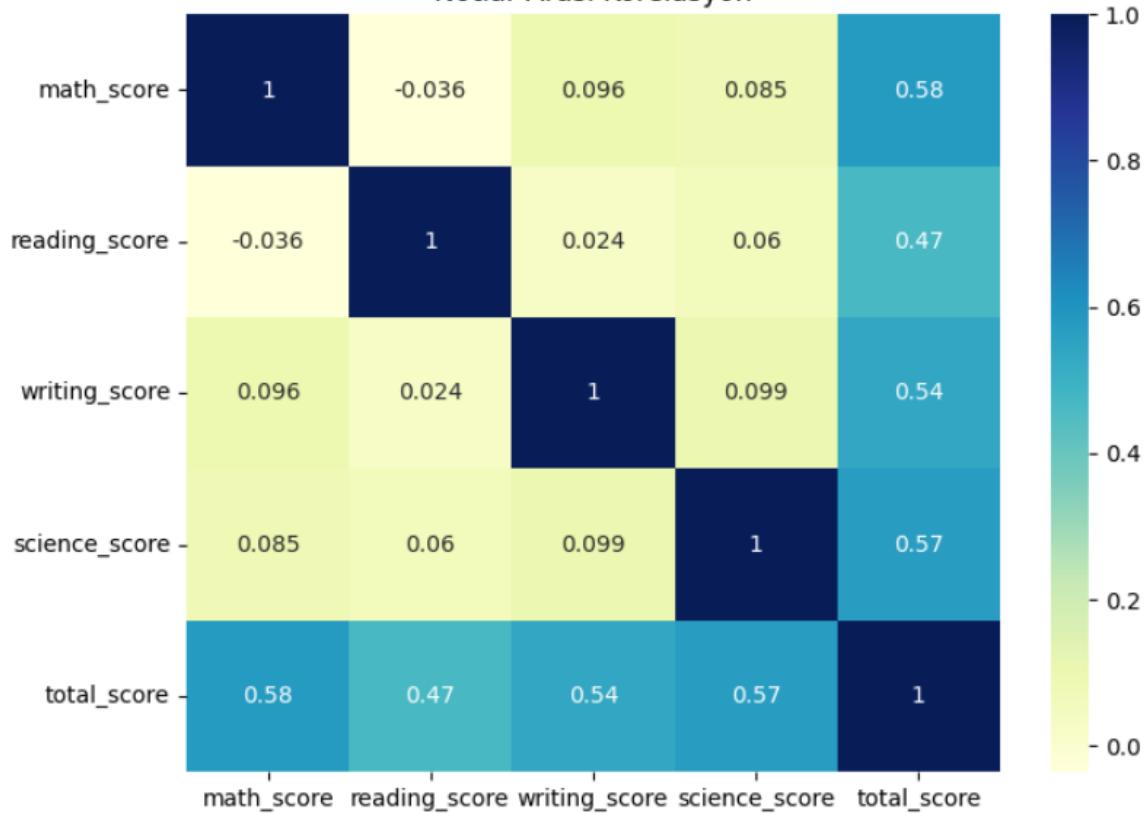
Sınav Hazırlığına Göre Toplam Puan Dağılımı



4. Korelasyon Analizi

- math_score, reading_score, writing_score, science_score ve total_score değişkenleri arasındaki ilişkiler korelasyon matrisi ile görselleştirilmiştir.
- Bu analiz, hangi dersler arasında pozitif ya da negatif güçlü ilişkiler olduğunu anlamaya yardımcı olur.

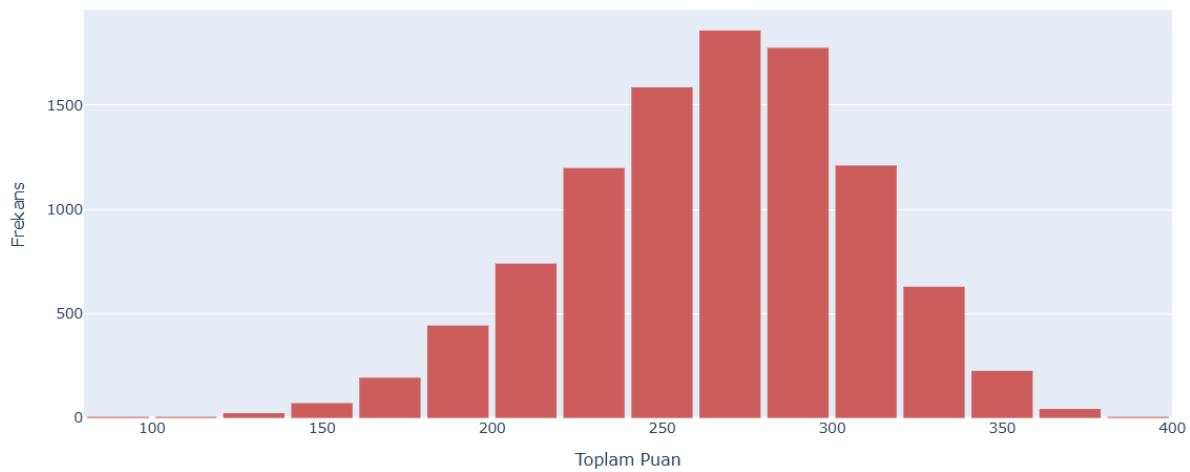
Notlar Arası Korelasyon



5. Notların Dağılımı: Histogram

- total_score değişkeni için histogram çizilerek notların frekans dağılımı incelenmiştir.
- Öğrencilerin genel başarı seviyesi bu grafik ile hızlıca anlaşılabilir.

[Toplam Puan Dağılımı \(Histogram\)](#)



6. Kategorik Değişkenlerin Sayısallaştırılması

- LabelEncoder ile gender, race_ethnicity, parental_level_of_education, ve grade sütunları sayısal formata çevrilmiştir.

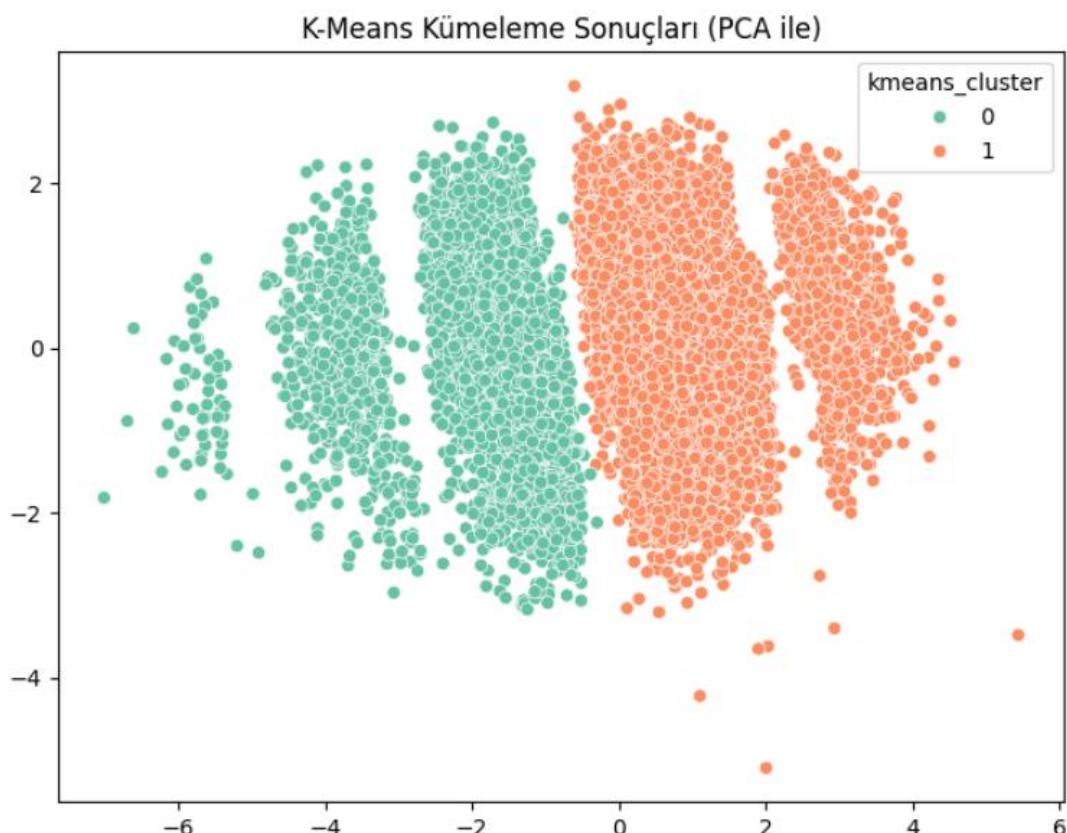
- Bu işlem, makine öğrenmesi algoritmalarının bu verilerle çalışabilmesini sağlar.

7. Ölçeklendirme

- Veriler, StandardScaler kullanılarak standartlaştırılmıştır (ortalama = 0, std = 1).
- Bu adım, özellikle mesafeye dayalı algoritmalarla önemlidir.

8. Kümeleme (K-Means)

- Silhouette skoru kullanılarak farklı k değerleri (küme sayıları) denenmiş ve en uygun küme sayısı belirlenmiştir.
- Kümeleme sonuçları, PCA ile iki boyuta indirgenmiş ve scatter plot ile görselleştirilmiştir.



9. Kümelere Göre Ortalama Ders Başarıları

Bu analizde, KMeans kümeleme sonucunda elde edilen her küme için beş farklı dersin (math_score, reading_score, writing_score, science_score, total_score) ortalamaları hesaplanmıştır.

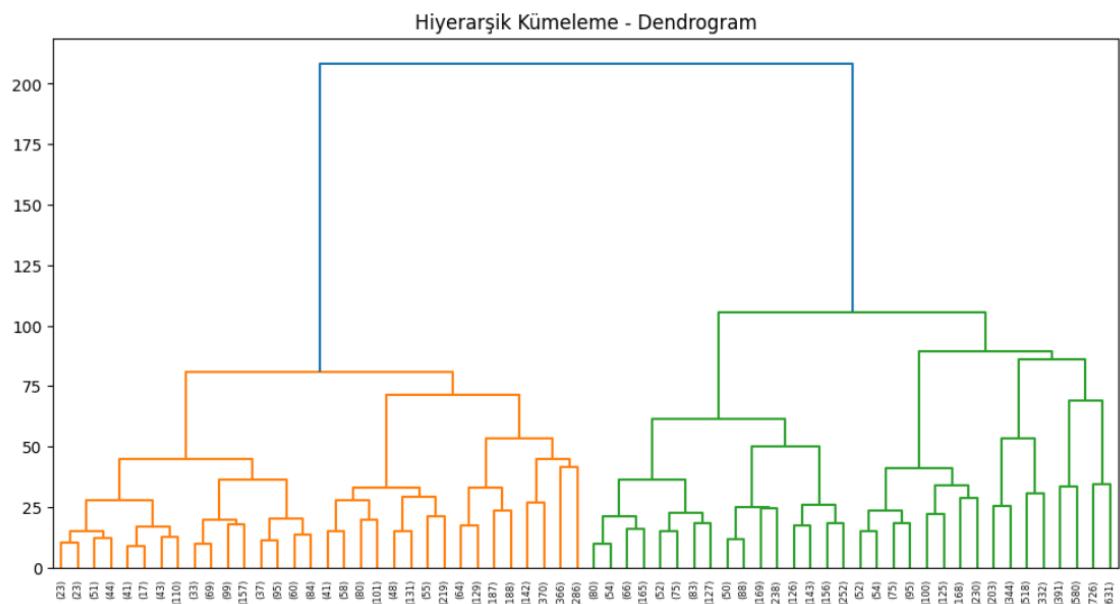
- Amaç: Her kümenin akademik profiline dair genel başarı düzeylerini ortaya koymak.
- Yöntem: groupby() fonksiyonu ile kmeans_cluster değişkenine göre gruptama yapılmış ve her dersin ortalaması alınmıştır.
- Elde edilen ortalamalar uzun formata çevrilerek Plotly kütüphanesi ile grup çubuk grafik (grouped bar chart) şeklinde görselleştirilmiştir.

Kümelere Göre Ortalama Ders Başarıları



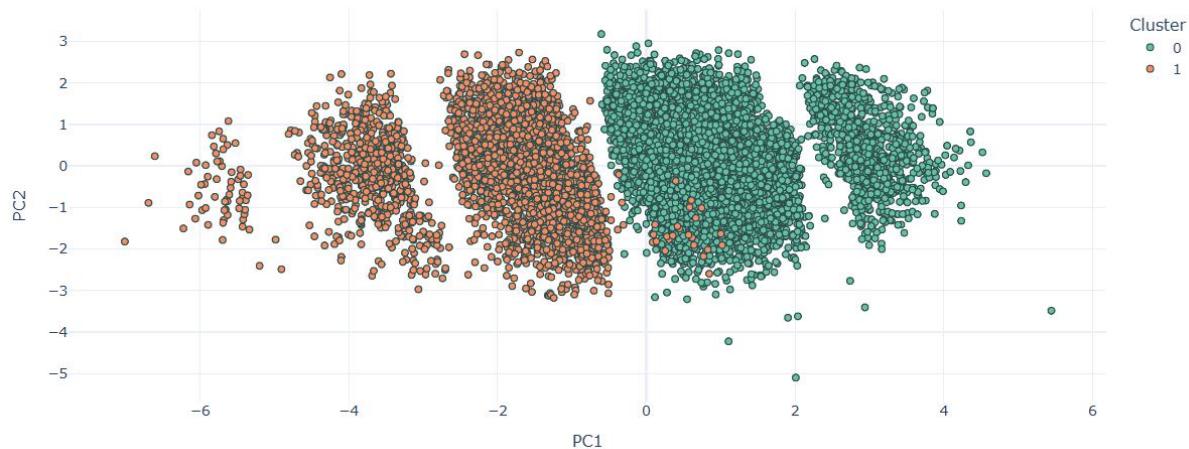
10. Hiyerarşik Kümeleme

- Ward bağlantı yöntemi ile dendrogram çizilmiş ve veri içindeki yapılar incelenmiştir.



- Ardından, AgglomerativeClustering ile 2 kümeli model oluşturulmuş ve PCA ile görselleştirilmiştir.

Hiyerarşik Kümeleme Sonuçları (PCA ile 2B)

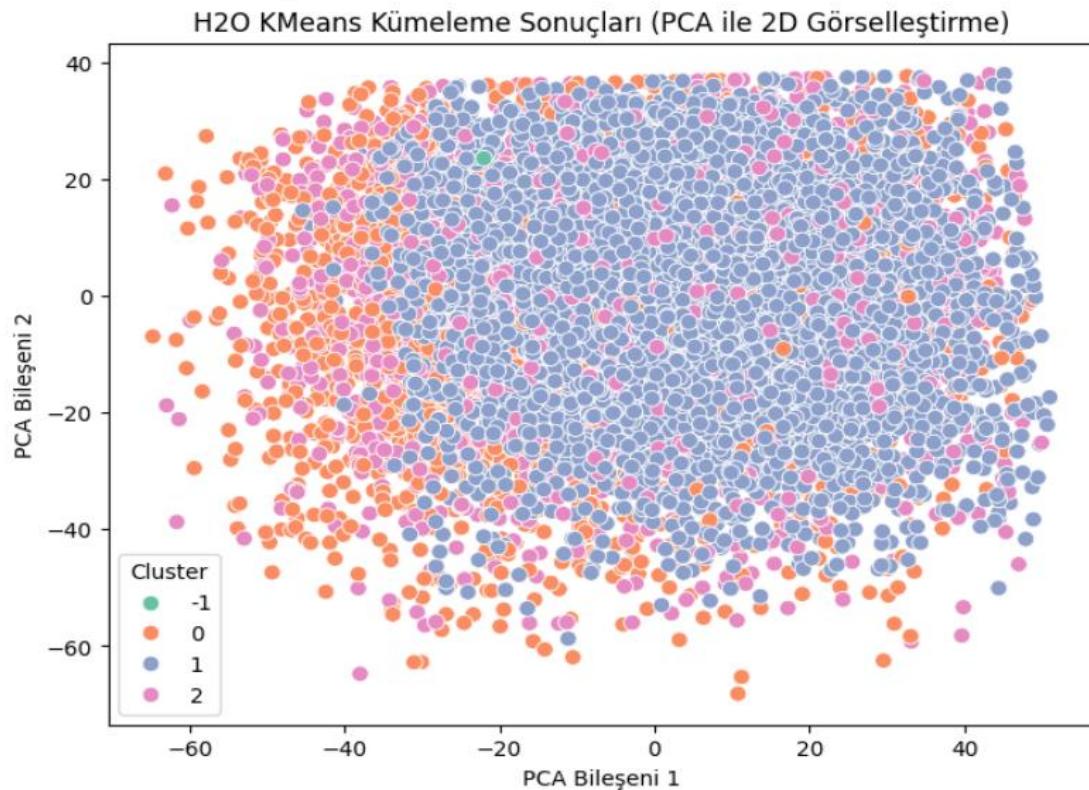


11. DBSCAN Kümeleme

- Gürültü verileri de tespit edebilen yoğunluk temelli bir yöntem olan DBSCAN uygulanmıştır.
- Gürültü veriler (-1 etiketli) ayrı olarak değerlendirilmiş ve silhouette skoru hesaplanmıştır.

12. H2O AutoML KMeans Sonuçlarının İncelenmesi

- H2O ile yapılan KMeans kümeleme sonuçları, PCA ile 2 boyuta indirgenmiş ve seaborn ile görselleştirilmiştir.
- Eksik tahmin değerleri -1 ile doldurularak küme ataması tamamlanmıştır.



4. Öğrenci Performansı Faktörü Verisi Üzerinde Birlikteklilik Kural Çıkarımı

1. Veri Seti

- Gözlem Sayısı: 6607
- Değişken Sayısı: 20
- Veri Tipleri: Sayısal (int64) ve kategorik (object) değişkenler içerir.
- Hedef Değişken: Exam_Score (int)

2. Eksik Veri İşlemleri

Aşağıdaki sütunlarda eksik değerler gözlemlenmiştir:

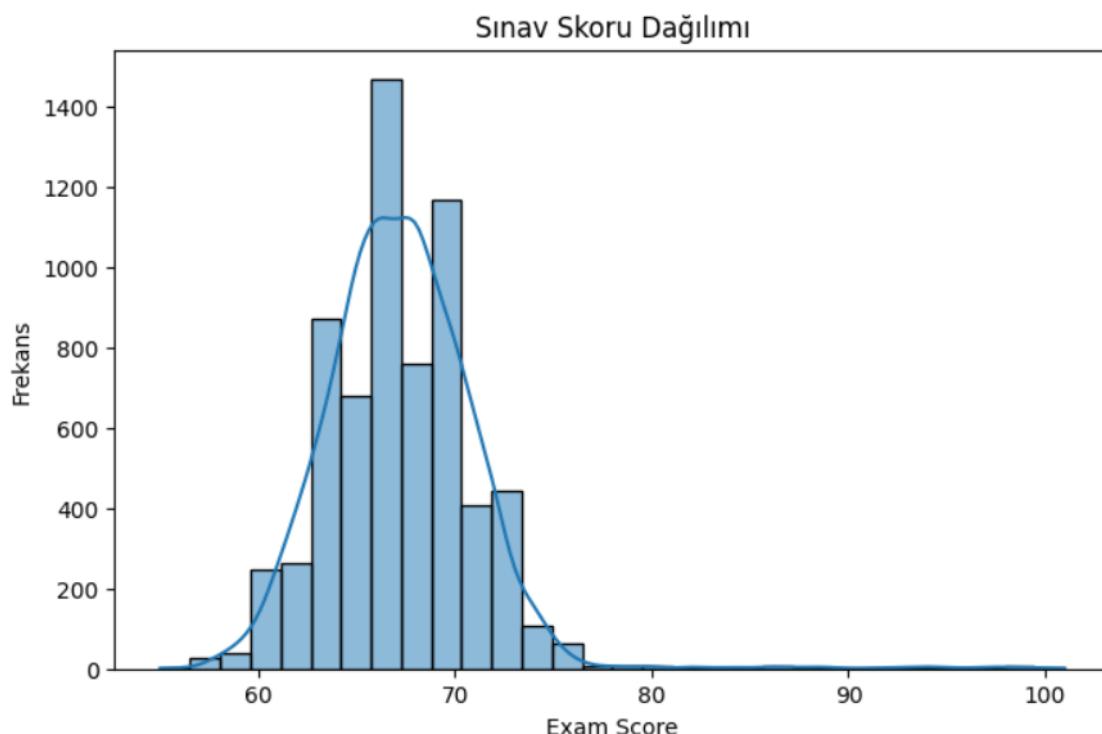
- Teacher_Quality: 78 eksik → 'Unknown' ile dolduruldu.
- Parental_Education_Level: 90 eksik → 'Unknown' ile dolduruldu.
- Distance_from_Home: 67 eksik → 'Unknown' ile dolduruldu.

Böylece veri kaybı yaşanmadan temiz bir analiz ortamı sağlanmıştır.

3. Temel Görselleştirmeler

■ Sınav Skoru Dağılımı

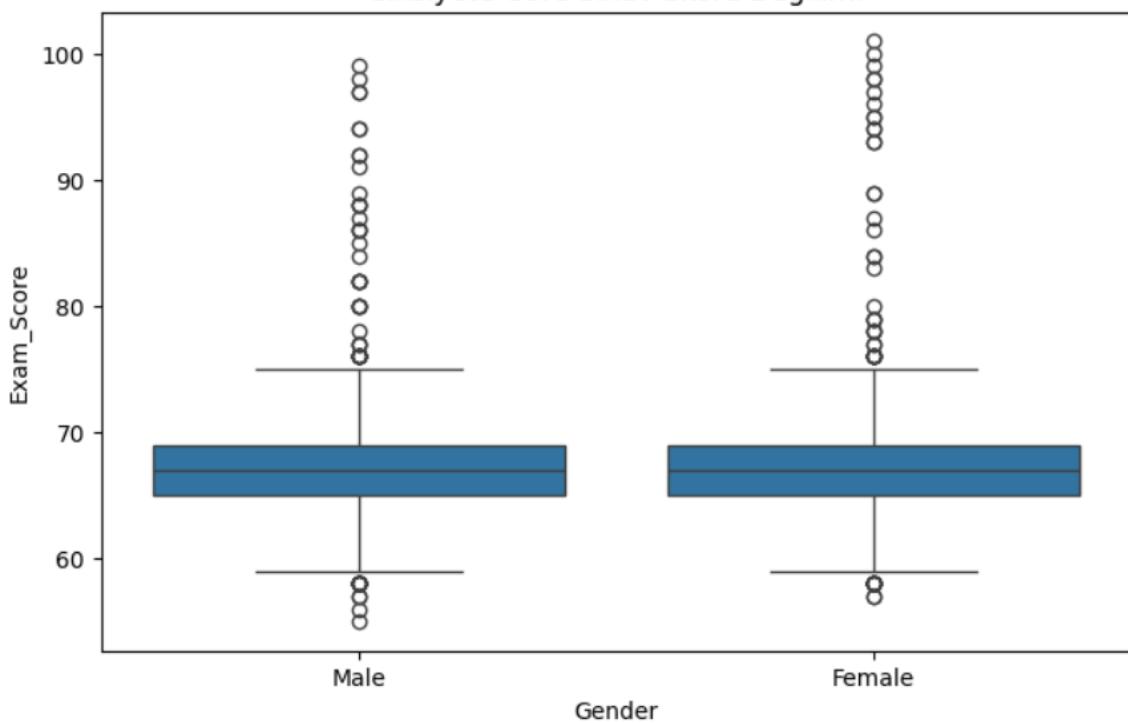
Sınav skorlarının histogram ve KDE grafiği ile dağılımı analiz edilmiştir. Skorlar genelde 60-80 arasında yoğunlaşmaktadır.



■ Cinsiyete Göre Sınav Başarısı

Boxplot kullanılarak kadın ve erkek öğrencilerin sınav skor dağılımları karşılaştırılmıştır.

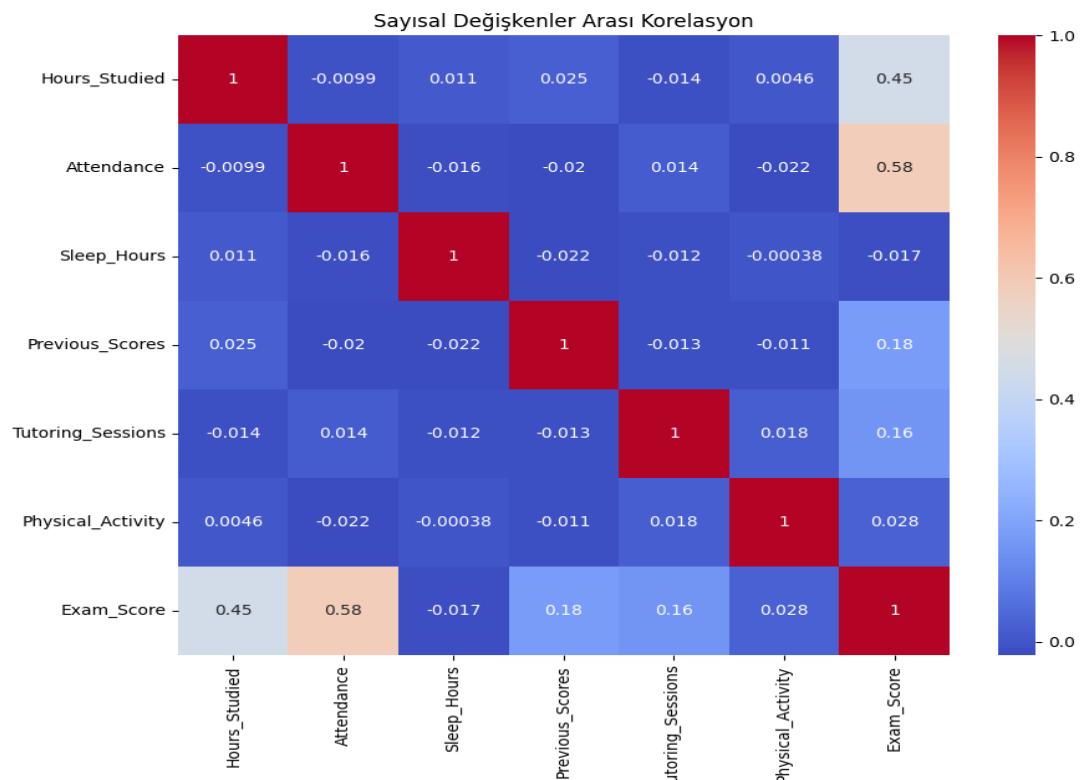
Cinsiyete Göre Sınav Skoru Dağılımı



■ Korelasyon Matrisi

Sayısal değişkenler arasındaki ilişkiler Pearson korelasyonu ile görselleştirilmiştir.

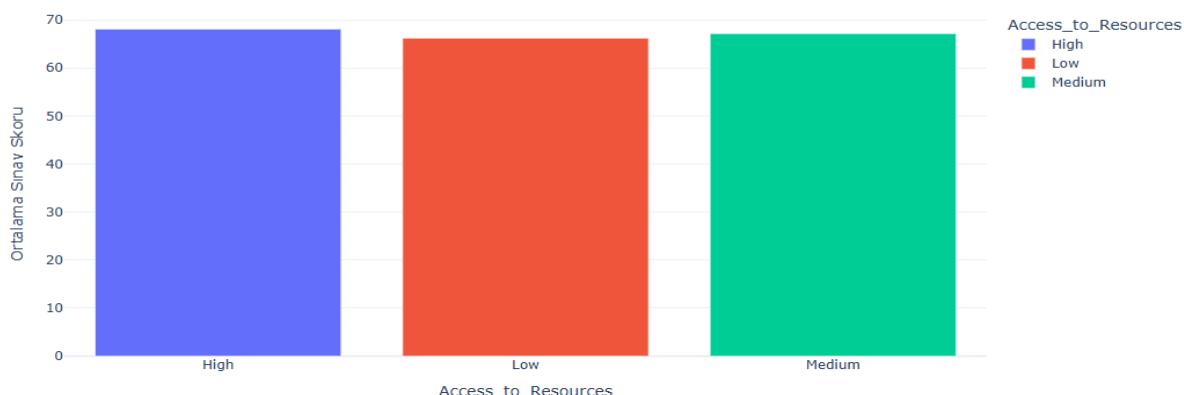
Previous_Scores ile Exam_Score arasında pozitif ilişki gözlemlenmiştir.



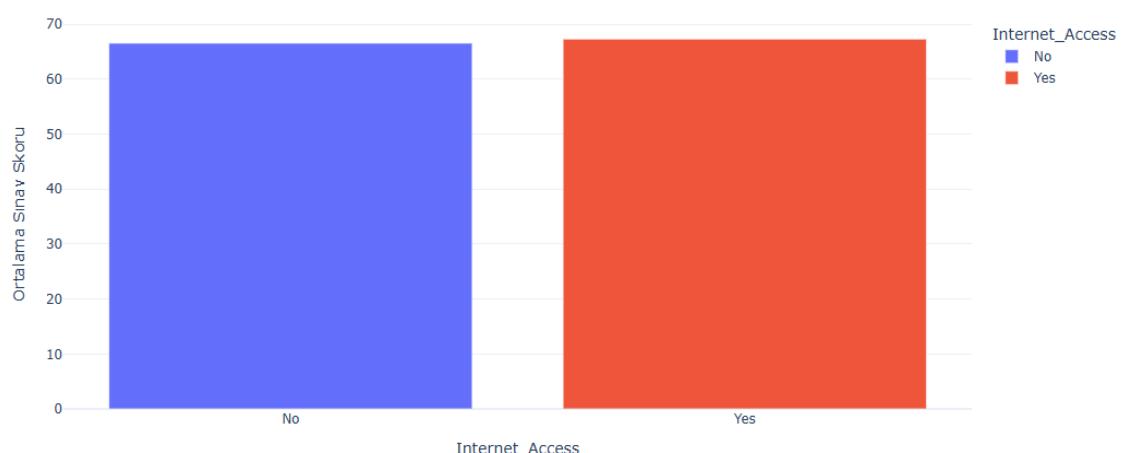
Kategorik Değişken Analizi

- Access_to_Resources, Parental_Involvement, Motivation_Level, Internet_Access, School_Type, Family_Income gibi değişkenlerin her bir seviyesi için ortalama Exam_Score hesaplanmış ve bar grafiklerle karşılaştırılmıştır.
- Bu sayede hangi kategorilerin başarıya daha çok katkı sağladığı görselleştirilmiştir.

Access_to_Resources Değişkenine Göre Ortalama Sınav Skoru



Internet_Access Değişkenine Göre Ortalama Sınav Skoru



3. Başarı Seviyesi Oluşturma

Sınav skoru (Exam_Score) değişkeni üç eşit gruba bölünerek başarı düzeyi (Success_Level) kategorisi oluşturulmuştur:

- Low, Medium, High

Bu değişken birlikte analizi için hedef değişken olarak kullanılmıştır.

4. Apriori Birliktelik Analizi

1. Veri Hazırlığı:

- Kategorik değişkenler ve Success_Level birleştirilerek her öğrenci için birer "alışveriş sepeti" formatı oluşturulmuştur.

- TransactionEncoder ile one-hot encode edilmiş yapı elde edilmiştir.

2. Sık Örüntülerin Çıkarılması:

	support	itemsets	
17	0.972908	(Yes)	
7	0.947934	(Medium)	• min_support = 0.05 ile sık öğe kümeleri çıkarılmıştır.
12	0.943242	(No)	• En sık geçen kombinasyonlar belirlenmiştir.
111	0.922658	(Medium, Yes)	
141	0.916150	(No, Yes)	

3. Birliktelik Kuralları:

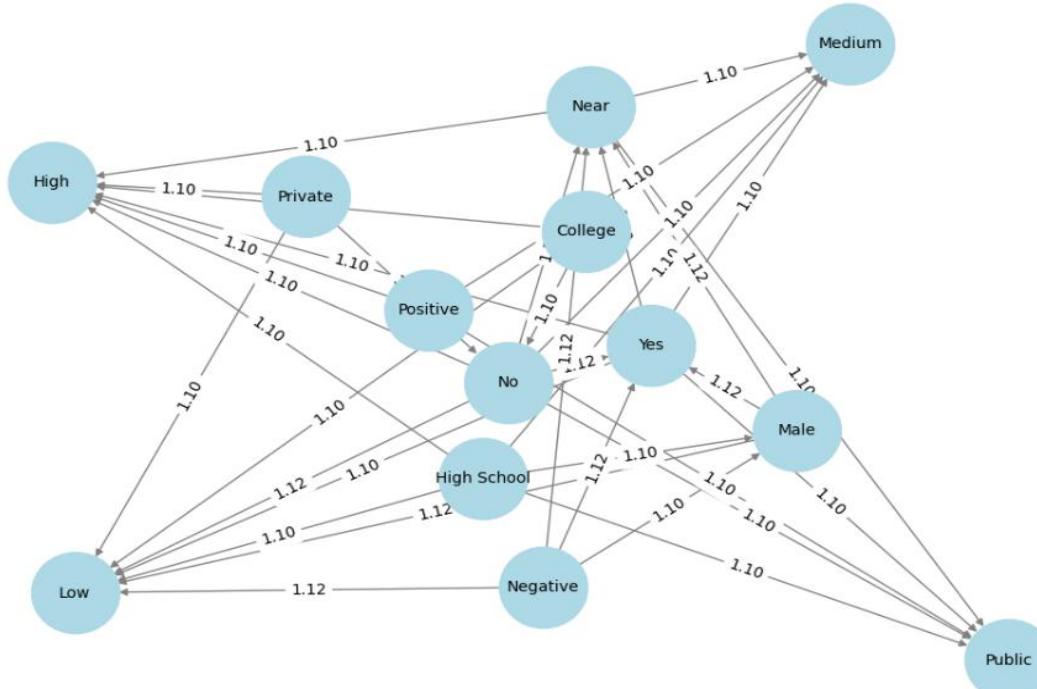
- confidence ≥ 0.6 , lift ≥ 1.1 şartları ile kurallar türetilmiştir.
- antecedents (öncüller) ve consequents (sonuçlar) şeklinde en anlamlı kurallar analiz edilmiştir.

5. Network Grafiği ile Kural Görselleştirmesi

- İlk 10 kuralın görselleştirilmesi NetworkX kütüphanesi ile yapılmıştır.
- Düğümler (nodes): Değişken değerleri
- Kenarlar (edges): Kurallar
- Kenar kalınlıkları: Lift değerini temsil eder

Bu grafik, hangi özelliklerin bir arada başarıyı etkilediğini görsel olarak anlamayı kolaylaştırır.

Birliktelik Kuralları - Network Grafiği



6.Tüm Değişken Dönüşümleri

Sürekli değişkenler kategorik hale getirilerek birliktelik analizine uygun formata getirilmiştir:

- Hours_Studied: Low / Medium / High
- Attendance: Low / Medium / High
- Sleep_Hours: Very Low / Low / Optimal / High
- Previous_Scores: Low / Medium / High
- Tutoring_Sessions: None / Few / Frequent
- Physical_Activity: None / Low / Medium / High

Bu dönüşümler, öğrencilerin özelliklerinin sınıflandırılmasını kolaylaştırarak desen keşfini kolaylaştırmıştır.

Tüm Değişkenlerin kategorik değerlere dönüştürülmesinin ardından, öğrencilerin sahip olduğu tüm nitelikler birer “öge” gibi değerlendirilmiş ve her öğrenci bir “işlem” (transaction) gibi ele alınarak analiz için uygun hale getirilmiştir. Bu yapı sayesinde birliktelik analizi (Association Rule Mining) uygulanabilir hâle gelmiştir.

► Kullanılan Yöntem: Apriori Algoritması

Bu aşamada mlxtend kütüphanesi kullanılarak aşağıdaki adımlar uygulanmıştır:

1. Sık Örüntülerin Belirlenmesi (Frequent Itemsets):

Apriori algoritması ile en az %5 destek oranına (min_support=0.05) sahip öge kombinasyonları belirlenmiştir. Bu kombinasyonlar, öğrenciler arasında sıkça birlikte görülen nitelikleri yansımaktadır.

2. Birliktelik Kurallarının Türetilmesi:

confidence ≥ 0.6 koşulu ile güçlü kurallar çıkarılmıştır. Her kural şu üç temel ölçütle değerlendirilmiştir:

- **Support (Destek):** Kurala konu olan öğelerin tüm veri setinde görülmeye oranı.
- **Confidence (Güven):** Kuraldaki öncül gerçekleştiğinde sonucun gerçekleşme olasılığı.
- **Lift:** Kurala konu olan öğelerin bir arada bulunmasının rastgelelige göre gücü (lift > 1 ise pozitif ilişki vardır).

3. En Güçlü Kuralların İncelenmesi:

rules.head(5) komutu ile en yüksek güven düzeyine sahip ilk 5 kural listelenmiştir. Bu kurallar, öğrencilerin belirli özellik kombinasyonlarına sahip olduklarıanda başarı düzeylerinin de tahmin edilebilir olduğunu göstermektedir.

rules.head(5)											
	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	representativity	leverage	conviction	zhangs_metr
0	(Access_to_Resources=High)	(Internet_Access=Yes)	0.298925	0.924474	0.275314	0.921013	0.996256		1.0	-0.001035	0.956178
1	(Access_to_Resources=High)	(Learning_Disabilities=No)	0.298925	0.894809	0.268352	0.897722	1.003255		1.0	0.000871	1.028481
2	(Access_to_Resources=High)	(Physical_Activity=Low)	0.298925	0.631451	0.191010	0.638987	1.011934		1.0	0.002253	1.020874
3	(Access_to_Resources=High)	(School_Type=Public)	0.298925	0.695929	0.201756	0.674937	0.969836		1.0	-0.006275	0.935422
4	(Access_to_Resources=Low)	(Extracurricular_Activities=Yes)	0.198729	0.596035	0.121689	0.612338	1.027354		1.0	0.003240	1.042056

5. Öğrenci Performansı Verisi Üzerinde Anomali Tespiti

1. Veri Seti

Bu veri seti, öğrencilerin demografik bilgileri, ders çalışma alışkanlıkları, teknolojik araç kullanımı, stres seviyeleri ve sosyal medya aktiviteleri gibi çeşitli akademik ve davranışsal değişkenleri içermektedir. Toplamda 10.000 öğrenciden oluşmakta ve 15 değişken barındırmaktadır.

- Nitel (kategorik) değişkenler: Gender, Learning Style, Stress Level gibi öğrencilerin özelliklerini tanımlayan alanlar.
- Sayısal değişkenler: Age, Study Hours per Week, Exam Score (%) gibi ölçülebilir bilgiler.

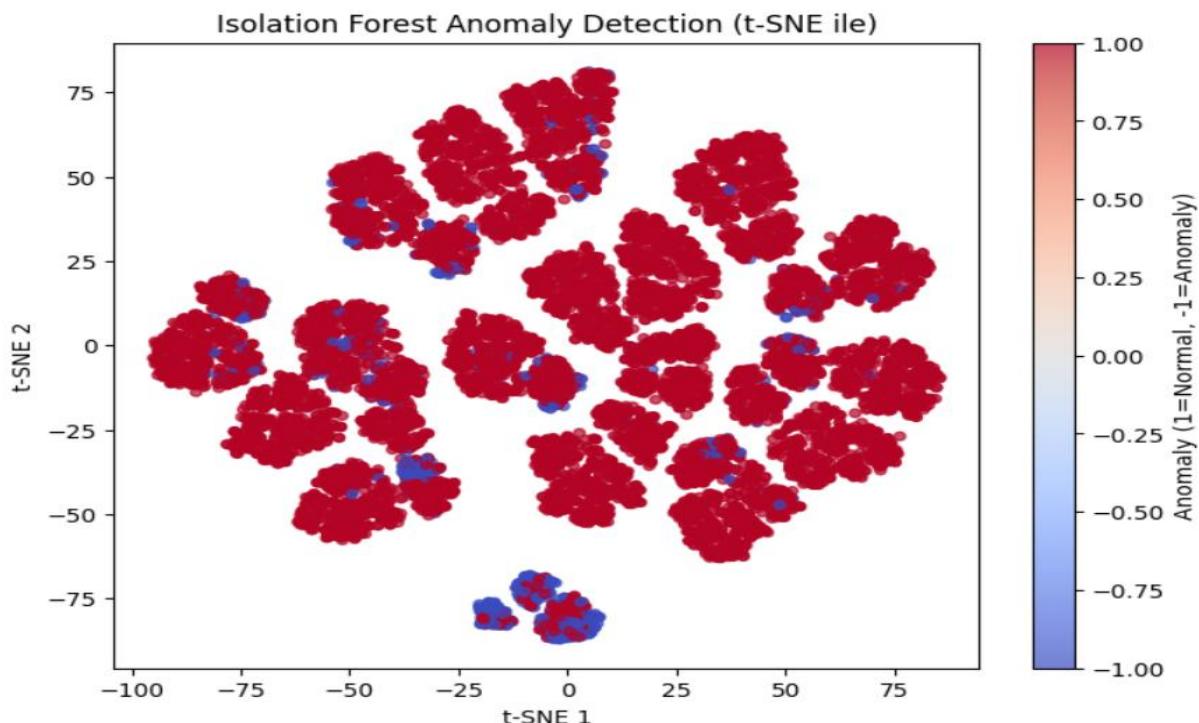
Veride eksik değer bulunmamaktadır ve veri ön işleme aşamasında `get_dummies()` ile kategorik değişkenler sayısal forma çevrilmiş, ardından `StandardScaler` ile ölçeklendirme yapılmıştır.

2. Anomali Tespiti (Outlier Detection)

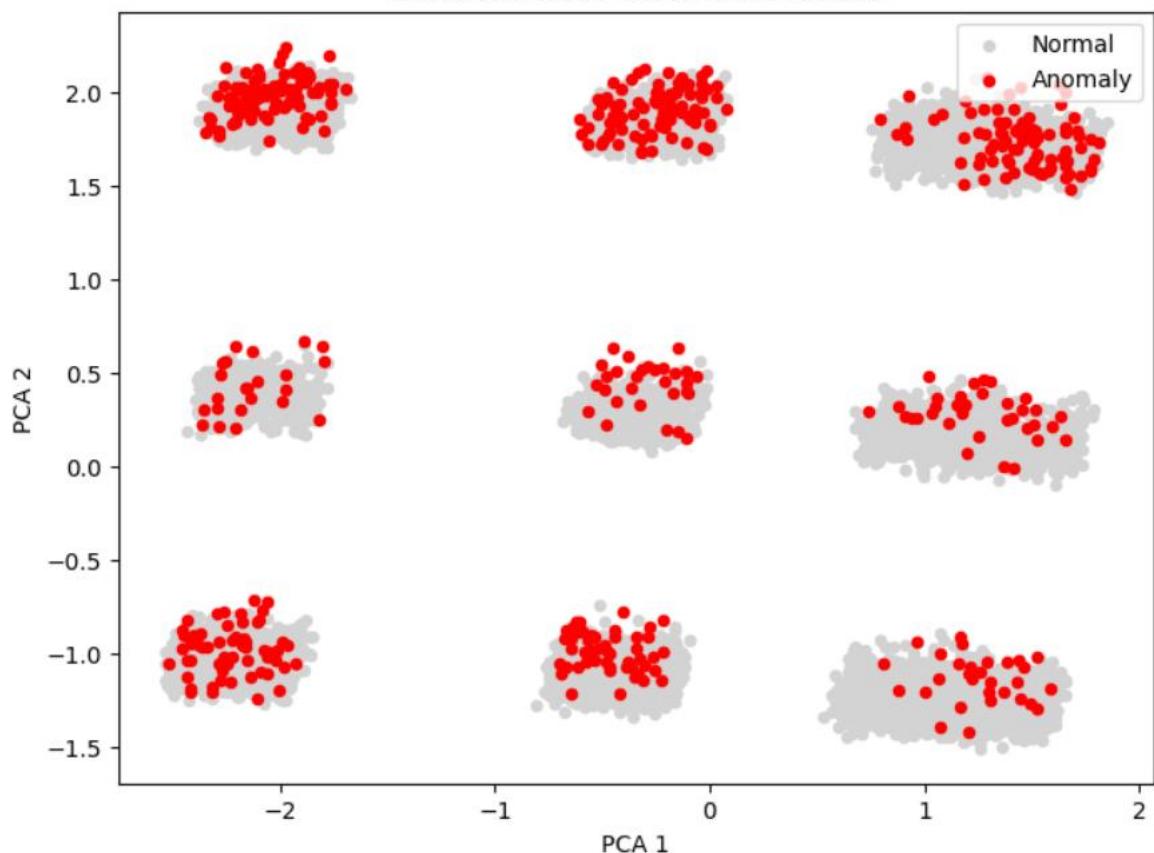
Öğrencilerin davranış ve performans özellikleri analiz edilerek olağan dışı (anormal) durumlar tespit edilmiştir. Bu kapsamında aşağıdaki yöntemler uygulanmıştır:

3. Isolation Forest ile Anomali Tespiti

- Yöntem: IsolationForest algoritması, veri noktalarını izole ederek sıradışı olanları (-1) işaretler.
- Görselleştirme: PCA ve t-SNE ile 2 boyuta indirilen veriler üzerinde normal öğrenciler gri, anomaliler kırmızı renkle gösterilmiştir.
- Sonuç: Tespit edilen anormal örnek sayısı gözlemlenmiş ve dağılım grafiklerle görselleştirilmiştir.

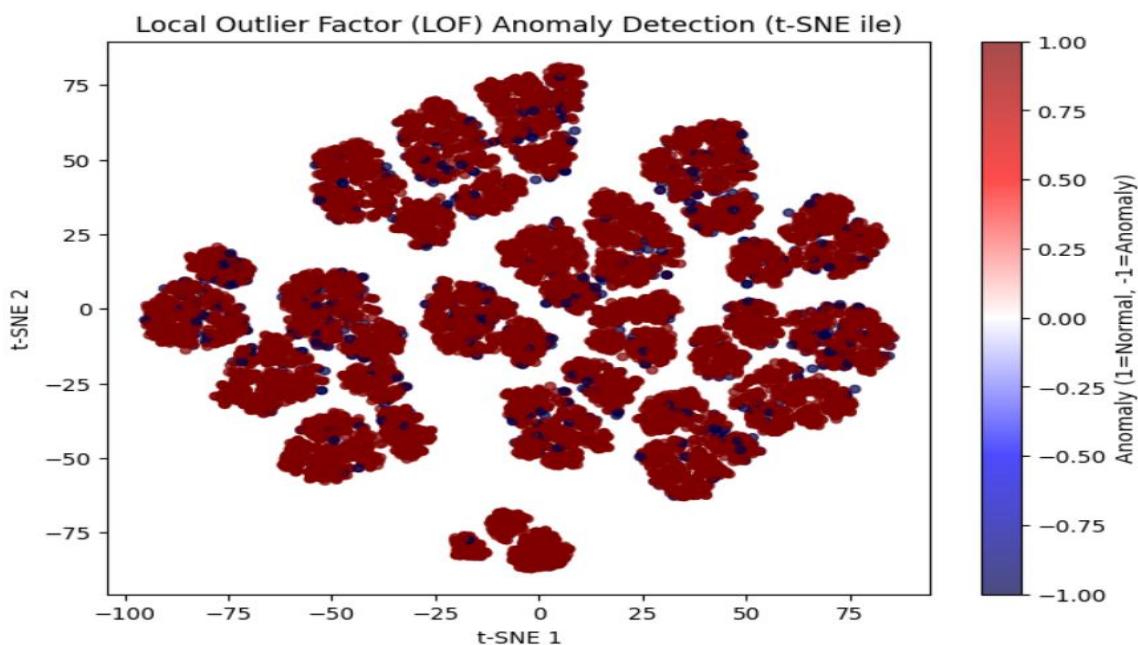


Isolation Forest - Anomaliler Kırmızı



4. Local Outlier Factor (LOF) ile Anomali Tespiti

- Yöntem: LOF algoritması, her veri noktasının komşularına olan uzaklığına göre anomali puanı hesaplar.
- Görselleştirme: t-SNE kullanılarak LOF ile bulunan anomaliler 2 boyutlu düzlemede görselleştirilmiştir.

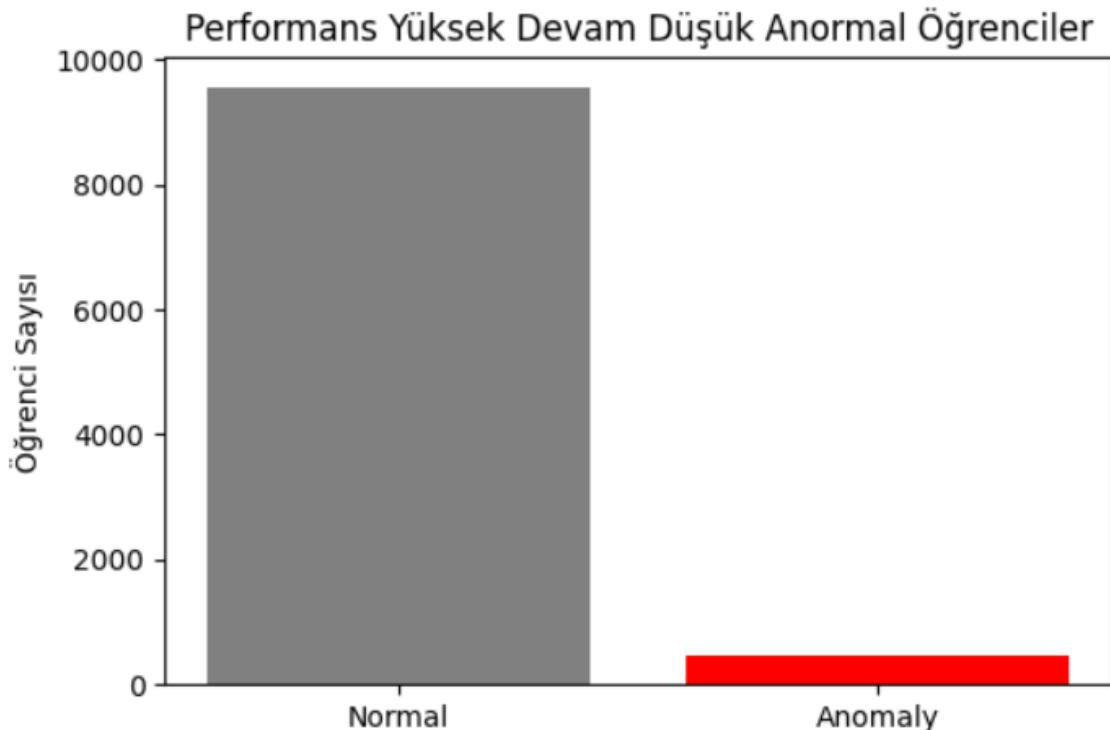


5. Özel Durumlara Dayalı Anomali Analizi

Veri içeriğine göre anlamlı ancak olağan dışı durumlar manuel olarak da tanımlanmıştır:

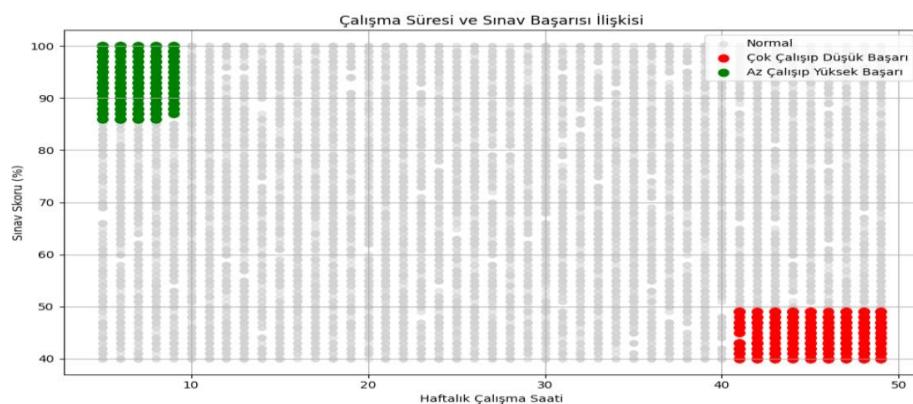
◆ Yüksek Başarı - Düşük Devam:

- Koşul: Exam Score > 85, Attendance Rate < 60
 - Bu durumdaki öğrenciler başarıya rağmen derslere düzenli katılmamaktadır. Grafiksel olarak da ayrı gösterilmiştir.



◆ Çok Çalışıp Düşük Başaranlar / Az Çalışıp Yüksek Başaranlar:

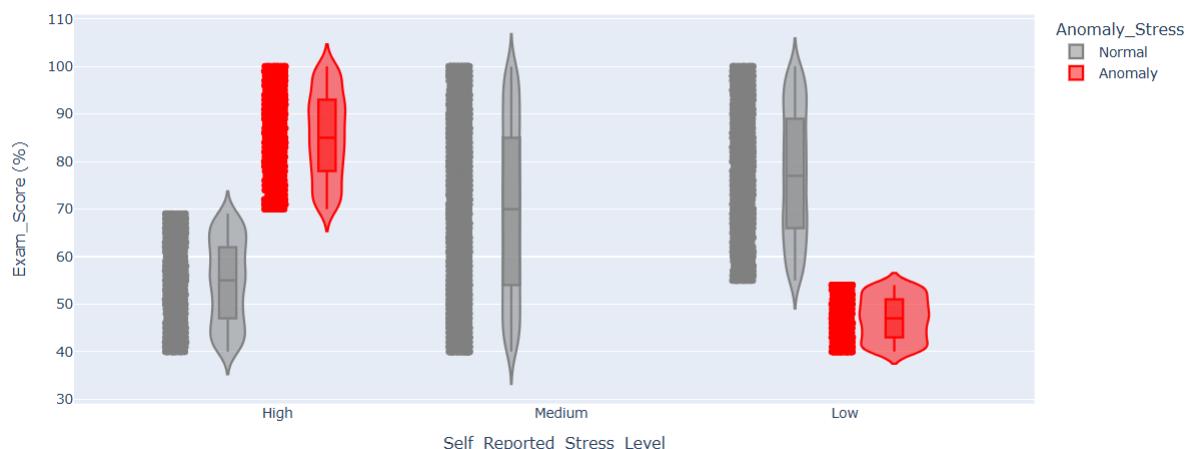
- Koşullar:
 - Study Hours > 40 & Exam Score < 50
 - Study Hours < 10 & Exam Score > 85
 - Bu üç örnekler başarı ile çalışma süresi arasında beklenen ilişkinin dışındaki durumları gösterir.



◆ **Stres ve Not Uyumsuzluğu:**

- Koşul: Stres seviyesi düşük ama notu düşük olan veya stres seviyesi yüksek ama notu yüksek olan öğrenciler.
- Amaç: Öğrencilerin öznel stres beyanları ile objektif başarıları arasındaki uyumsuzlukları belirlemek.

Stres Seviyesi ve Sınav Skoru Dağılımı (Anormal vs Normal)



◆ **Tartışmaya Katılmadan Başarı Gösterenler:**

- Koşul: Katılım = "No" ve Final Grade $\in \{A, B\}$
- Bu öğrenciler, aktif katılım göstermeden yüksek başarıya ulaşmışlardır. Grafikle anomali dağılımı görselleştirilmiştir.

