T.C.
İSTANBUL ÜNİVERSİTESİ-CERRAHPAŞA
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

# DOKTORA TEZİ

# MAKİNE ÖĞRENMESİ YÖNTEMLERİ KULLANILARAK BORÇ ÖDEME HAREKETLERİNİN ANALİZİ İLE KREDİ SKORU TAHMİNİ

**Cem BULUT**

**DANIŞMAN**
**Doç. Dr. Emel ARSLAN**

**Bilgisayar Mühendisliği Anabilim Dalı**

**Bilgisayar Mühendisliği, Doktora Programı**

**Şubat, 2025**

# TEZ KABUL VE ONAYI

**Cem BULUT** tarafından, **Doç. Dr. Emel ARSLAN** danışmanlığında hazırlanan **"MAKİNE ÖĞRENMESİ YÖNTEMLERİ KULLANILARAK BORÇ ÖDEME HAREKETLERİNİN ANALİZİ İLE KREDİ SKORU TAHMİNİ"** başlıklı bu çalışma, jürimiz tarafından **03/02/2025** tarihinde yapılan sınav sonucunda **oy birliği** ile başarılı bulunarak **Doktora Tezi** olarak kabul edilmiştir.

**Tez Jürisi**

| | | **İmza** | **Sonuç** |
|---|---|---|---|
| **DANIŞMAN** | Doç. Dr. Emel ARSLAN<br>İstanbul Üniversitesi- Cerrahpaşa<br>Bilgisayar Mühendisliği ABD. | | ☒ Kabul<br>☐ Ret |
| **ÜYE** | Prof. Dr. Rüya ŞAMLI<br>İstanbul Üniversitesi- Cerrahpaşa<br>Bilgisayar Mühendisliği ABD. | | ☒ Kabul<br>☐ Ret |
| **ÜYE** | Doç. Dr. Neyir ÖZCAN SEMERCİ<br>Bursa Uludağ Üniversitesi<br>Elektrik-Elektronik Mühendisliği<br>ABD. | | ☒ Kabul<br>☐ Ret |
| **ÜYE** | Prof. Dr. Ahmet SERTBAŞ<br>İstanbul Üniversitesi- Cerrahpaşa<br>Bilgisayar Mühendisliği ABD. | | ☒ Kabul<br>☐ Ret |
| **ÜYE** | Prof. Dr. Erdinç UZUN<br>Tekirdağ Namık Kemal Üniversitesi<br>Bilgisayar Mühendisliği ABD. | | ☒ Kabul<br>☐ Ret |

# BEYAN

Bu tez çalışmasının kendi çalışmam olduğunu, tezin planlanmasından yazımına kadar bütün aşamalarda etik dışı davranışımın olmadığını, bu tezdeki bütün bilgileri akademik ve bilimsel etik kuralları içinde elde ettiğimi, bu tez çalışmasıyla elde edilmeyen bütün bilgi ve yorumlara kaynak gösterdiğimi ve bu kaynakları da kaynaklar listesine aldığımı, yine bu tezin çalışılması ve yazımı sırasında patent ve telif haklarını ihlal edici bir davranışımın olmadığını ve her türlü hukuki sorumluluğu aldığımı kabul ederim.

Cem BULUT

Aileme ithaf ediyorum…

# BÜTÇE DESTEKLERİ

## MAKİNE ÖĞRENMESİ YÖNTEMLERİ KULLANILARAK BORÇ ÖDEME HAREKETLERİNİN ANALİZİ İLE KREDİ SKORU TAHMİNİ

Bu tez çalışması 100/2000 YÖK Doktora Bursu ile desteklenmiştir.

# TEŞEKKÜR

Doktora tez sürecimin her aşamasında bilgi ve tecrübesiyle katkı sağlayan, kişisel ve akademik gelişimimde önemli rol oynayan değerli danışmanım Doç. Dr. Emel ARSLAN' a saygı ve teşekkürlerimi sunarım.

Değerli fikirleri ve yönlendirmeleri ile tezimin gelişimine katkı sağlayarak rehberlik eden Prof. Dr. Rüya ŞAMLI ve Doç. Dr. Neyir ÖZCAN SEMERCİ' ye sonsuz teşekkürlerimi sunarım.

Son olarak bugüne kadar bana güvenen ve destek olan tüm hocalarım, arkadaşlarım ve aileme sonsuz teşekkürler.

Şubat 2025 Cem BULUT

# İÇİNDEKİLER

**Sayfa No**

**1. GİRİŞ ..... 1**
**2. GENEL KISIMLAR ..... 5**
2.1. Kredi Skoru (Credit Score) ..... 5
2.2. Makine Öğrenmesi (Machine Learning) ..... 6
2.3. Veri Ön İşleme (Data Preprocessing) ..... 8
2.3.1. Kusurlu Veriler (Imperfect Data) ..... 8
2.3.2. Boyutsal Problemler (Dimensional Problems) ..... 9
2.3.3. Örnek Azaltma veya Arttırma ..... 11
2.3.4. Ayrıklaştırma ve Normalizasyon (Discretization and Normalization) ..... 11
2.4. Sınıflandırma (Classification) ..... 12
2.4.1. Karar Ağaçları (Decision Trees) ..... 13
2.4.2. Rastgele Orman (Random Forest) ..... 13
2.4.3. Lojistik Regresyon (Logistic Regression) ..... 14
2.4.4. Naive Bayes ..... 14
2.4.5. K-En Yakın Komşu Algoritması (K-Nearest Neighbors) ..... 15
2.4.6. Destek Vektör Makinesi (Support Vector Machine) ..... 16
2.4.7. Çok Katmanlı Algılayıcı (Multi Layer Perception) ..... 16
2.4.8. Ekstra Ağaç Sınıflandırıcı (Extra Tree Classifier) ..... 17
2.4.9. Güçlendirme (Boosting) Yöntemleri ..... 18
2.4.10. Torbalama (Bootstrap Aggregating- Bagging) ..... 22
2.4.11. Boosting ve Bagging Yöntemlerinin Karşılaştırılması ..... 23
2.4.12. İkili Sınıflandırma Algoritmalarının Çok Sınıflı Sınıflandırmada Kullanımı ..... 25
2.5. Dengesiz Veri Problemi (Imbalanced Data Problem) ..... 27
2.5.1. Aşırı Örnekleme (Over Sampling) ..... 28
2.5.2. Yetersiz Örnekleme (Under Sampling) ..... 31
2.5.3. Hibrit Yöntemler (Hybrid Methods) ..... 36
2.6. Açıklanabilir Yapay Zeka (Explainable Artificial Intelligence- XAI) ..... 38
2.6.1. LIME ..... 41

2.6.2. SHAP ..... 42
2.7. Veri Doğrulama (Data Validation) ..... 42
2.8. Değerlendirme Metrikleri (Evaluation Metrics) ..... 43
**3. MALZEME VE YÖNTEM ..... 48**
3.1. Veri ..... 48
3.1.1. Veri Tabanı Yapısı ..... 50
3.1.2. Veri Setinin Hazırlanması ..... 52
3.1.3. Fatura Ödeme Alışkanlıkları Veri Setinin Oluşturulması ..... 54
3.2. YÖNTEM ..... 60
3.2.1. Verinin Sınıflandırılması ..... 60
3.2.2. Verinin İşlenmesi ve Sınıflandırılması ..... 60
3.2.3. XAI Yöntemleri ile En İyi ML Modelinin Açıklanması ..... 63
3.3. ARAÇLAR ..... 65
3.3.1. PyCharm Entegre Geliştirme Ortamı ..... 65
3.3.2. Python Programlama Dili ..... 66
3.3.3. Microsoft SQL Server ..... 69
**4. BULGULAR ..... 71**
**5. TARTIŞMA VE SONUÇ ..... 88**

# ŞEKİL LİSTESİ

**Sayfa No**

Şekil 2.1. Makine Öğrenmesi Modelleri. ..... 8
Şekil 2.2. MLP Akışı ..... 17
Şekil 2.3: Boosting Çalışma Prensibi ..... 24
Şekil 2.4: Bagging Çalışma Prensibi ..... 24
Şekil 2.5: ML Modellerinin XAI Süreci ..... 39
Şekil 2.6: XAI Modelleri ..... 39
Şekil 3.1: Veri Edinme Süreci ..... 49
Şekil 3.2: Ödeme Kuruluşu Veri Tabanı Tablo Yapısı ..... 50
Şekil 3.3: Cinsiyet Dağılımı ..... 55
Şekil 3.4: Yaş Dağılımı ..... 55
Şekil 3.5: Fatura Tipi Dağılımı ..... 56
Şekil 3.6: Ödenen Fatura Sayısı Dağılımı ..... 57
Şekil 3.7: Fatura Tipi Sayısı Dağılımı ..... 58
Şekil 3.8: Kredi Skoru Dağılımı ..... 59
Şekil 3.9: Verinin Sınıflandırılması ..... 60
Şekil 3.10: Verinin İşlenmesi ve Sınıflandırılması ..... 61
Şekil 3.11: Ön İşleme Öncesi ve Sonrasındaki Etiket Dağılımı ..... 62
Şekil 3.12: En Başarılı ML Modelinin Açıklanması ve Yorumlanması Adımları ..... 64
Şekil 4.1: Kredi Skoru=1 Olan Örnek için LIME Sonucu ..... 74
Şekil 4.2: Kredi Skoru=2 Olan Örnek için LIME Sonucu ..... 75
Şekil 4.3: Kredi Skoru=3 Olan Örnek için LIME Sonucu ..... 76
Şekil 4.4: Kredi Skoru=4 Olan Örnek için LIME Sonucu ..... 77

Şekil 4.5: Kredi Skoru=5 Olan Örnek için LIME Sonucu ..... 78
Şekil 4.6: Kredi Skoru=6 Olan Örnek için LIME Sonucu ..... 79
Şekil 4.7: Kredi Skoru=7 Olan Örnek için LIME Sonucu ..... 80
Şekil 4.8: Kredi Skoru=8 Olan Örnek için LIME Sonucu ..... 81
Şekil 4.9: Kredi Skoru=9 Olan Örnek için LIME Sonucu ..... 82
Şekil 4.10: Kredi Skoru=10 Olan Örnek için LIME Sonucu ..... 83
Şekil 4.11: Veri Setine Ait SHAP Özellik Önem Grafiği ..... 84
Şekil 4.12: Sınıf Etiketleri için SHAP Özet Grafikleri..... 86

# TABLO LİSTESİ

**Sayfa No**

Tablo 2.1: Bagging ve Boosting Yaklaşımlarının Karşılaştırılması ..... 25

Tablo 2.2: Bire-karşı-bir yaklaşımı sonucu elde edilen ikili sınıflandırıcı ..... 26

Tablo 2.3: Karışıklık Matrisi ..... 43

Tablo 3.1: Tablo Detayları ..... 50

Tablo 3.2: Fatura Tipleri ..... 51

Tablo 3.3: Nihai Veri Setinin Özellikleri ..... 54

Tablo 3.4: Veri Setinden Örnekler ..... 59

Tablo 4.1: Ham Veri ile Sınıflandırma Sonuçları ..... 71

Tablo 4.2: Ön İşleme ve Örnekleme Adımları Sonrası Elde Edilen En İyi Sonuçlar ..... 72

# SİMGE VE KISALTMA LİSTESİ

| Kısaltmalar | Açıklama |
|---|---|
| **IFRS** | : International Financial Reported Standards (Uluslararası Finansal Raporlama Standartları) |
| **PD** | : Probability of Default (Temerrüt Olasılığı) |
| **LGD** | : Loss Given Default (Temerrüt Halinde Kayıp) |
| **KKB** | : Kredi Kayıt Bürosu |
| **BKN** | : Bireysel Kredi Notu |
| **FICO** | : Fair, Isaac and Company (Fair, Isaac Şirketi) |
| **ML** | : Machine Learning (Makine Öğrenmesi) |
| **FS** | : Feature Selection (Özellik Seçimi) |
| **PCA** | : Principal Component Analysis (Temel Bileşen Analizi) |
| **DT** | : Decision Tree (Karar Ağacı) |
| **RF** | : Random Forest (Rastgele Orman) |
| **AdaBoost** | : Adaptive Boosting (Uyarlanabilir Yükseltme) |
| **LR** | : Logistic Regression (Lojistik Regresyon) |
| **K-NN** | : K-Nearest Neighbors (K-En Yakın Komşu) |
| **SVM** | : Support Vector Machine (Destek Vektör Makinesi) |
| **MLP** | : Multi-Layer Perception (Çok Katmanlı Algılayıcı) |
| **EXT** | : Extra Tree Classifier (Ekstra Ağaç Sınıflandırıcı) |
| **GB** | : Gradient Boosting (Gradyan Arttırma) |
| **XGBoost** | : Extreme Gradient Boosting (Aşırı Gradyan Arttırma) |
| **LGBM** | : Light Gradient Boosting Machine (Hafif Gradyan Arttırma Makinesi) |
| **GOSS** | : Gradient-based One-Side Sampling (Gradyan Tabanlı Tek Taraflı Örnekleme) |
| **EFB** | : Exclusive Feature Bundling (Özel Özellik Paketleme) |
| **Bagging** | : Bootstrap Aggregating (Torbalama) |
| **Boosting** | : Güçlendirme |
| **ROS** | : Random Over Sampling (Rastgele Aşırı Örnekleme) |

| | |
|---|---|
| **SMOTE** | : Synthetic Minority Over-sampling Technique (Sentetik Azınlık Aşırı Örnekleme Tekniği) |
| **ADASYN** | : Adaptive Synthetic Sampling Approach (Uyarlanabilir Sentetik Örnekleme) |
| **CC** | : Cluster Centroids (Küme Merkezleri) |
| **RUS** | : Random Under Sampling (Rastgele Yetersiz Örnekleme) |
| **ENN** | : Edited Nearest Neighbours (Düzenlenmiş En Yakın Komşular) |
| **RENN** | : Repeated ENN (Tekrarlanan ENN) |
| **CNN** | : Condensed Nearest Neighbours (Yoğunlaştırılmış En Yakın Komşular) |
| **OSS** | : One Sided Selection (Tek Taraflı Seçim) |
| **NCR** | : Neighbourhood Cleaning Rule (Komşu Temizliği Kuralı) |
| **XAI** | : Explainable Artificial Intelligence (Açıklanabilir Yapay Zeka) |
| **LIME** | : Local Interpretable Model Agnostic Explanations (Yerel Yorumlanabilir Model Agnostik Açıklamalar) |
| **SHAP** | : Shapley Additive Explanations (Shapley Katkı Açıklamaları) |
| **CV** | : Cross Validation (Çapraz Doğrulama) |
| **TP** | : True Positive (Doğru Pozitif) |
| **FP** | : False Positive (Yanlış Pozitif) |
| **TN** | : True Negative (Doğru Negatif) |
| **FN** | : False Negative (Yanlış Negatif) |
| **ACC** | : Accuracy (Doğruluk) |
| **PRE** | : Precision (Kesinlik) |
| **REC** | : Recall (Duyarlılık) |
| **FBS** | : F-Beta Score (F-Beta Skoru) |
| **MCC** | : Matthews Correlation Coefficient (Matthews Korelasyon Katsayısı) |
| **JSI** | : Jaccard Similarity Index (Jaccard Benzerlik Endeksi) |
| **ROC** | : Receiver Operating Characteristic (Alıcı Çalışma Karakteristiği) |
| **AUC** | : Area Under the Curve (Eğrinin Altındaki Alan) |

# ÖZET


## [DOKTORA TEZİ]

[MAKİNE ÖĞRENMESİ YÖNTEMLERİ KULLANILARAK BORÇ ÖDEME HAREKETLERİNİN ANALİZİ İLE KREDİ SKORU TAHMİNİ ]

**[Cem BULUT]**

**İstanbul Üniversitesi-Cerrahpaşa**

**Lisansüstü Eğitim Enstitüsü**

**Bilgisayar Mühendisliği Anabilim Dalı**

**Bilgisayar Mühendisliği, Doktora Programı**

**[Danışman: Doç. Dr. Emel ARSLAN ]**

[Finans sektörünün hızla büyümesiyle birlikte finansal risklerin minimize edilmesi konusunun önemi de artmaktadır. Bireysel borçlanmanın en yaygın biçimi olan kredi kullanımı da bu riskler arasında en başta yer almaktadır. Kredi sağlayıcılar, kredi başvurusu yapan kişinin geri ödeme riskini öngörecek sistemler kullanarak kişiye kredi verme ya da vermeme kararını vermektedir. Türkiye'de kullanılan kredi skoru Kredi Kayıt Bürosu (KKB)'ndan sağlanmaktadır. Fakat bu skor yalnızca geçmiş kredi ödemelerini baz alarak bir risk değerlendirmesi yaptığından bazı durumlarda yetersiz kalmaktadır.

Bu çalışmada makine öğrenmesi (Machine Learning- ML) ve açıklanabilir yapay zeka teknikleri (Explainable Artificial Intelligence- XAI) kullanılarak yeni ve alternatif bir kredi skoru önerilmektedir. Tez çalışmamızda literatürde yer alan çalışmaların aksine, kredi skoru tahmini için kredi veya kredi kartı geçmişi baz alınmamış; Türkiye' de yer alan bir fatura ödeme kuruluşuna ait gerçek fatura ödeme hareketlerinden, kişi bazlı fatura ödeme alışkanlıkları analizi yapılmış ve elde edilen veri, gerçek KKB skoru ile birleştirilmiştir. Tez çalışmasında veri temizleme, veri ölçeklendirme, yapay örnekleme teknikleri, sınıflandırma algoritmaları,

çapraz doğrulama ve açıklanabilir yapay zeka teknikleri kullanılmıştır. Sonuçlar birçok değerlendirme metriği ile doğrulanarak yeni bir model önerilmiştir.

Önerilen makine öğrenmesi modeli, özellikle bankacılık ve finans sektöründe makine öğrenmesi modellerinin kullanımı ve karmaşık modellerin kullanıcı tarafından anlaşılabilir ve yorumlanabilir olması açısından önemlidir.

Çalışmada sunulan model, geleneksel bir kredi skoru bulunmayan kişiler için alternatif bir skorlama yöntemi olarak kullanılabilir. Kredi skoru bulunan kişiler için ise var olan puana katkı sağlayacak veya doğrulama amaçlı bir çıktı olarak değerlendirilebilir.

Şubat 2025 , 116 sayfa.

**Anahtar kelimeler:** Makine Öğrenmesi, Kredi Skoru, Açıklanabilir Yapay Zeka, Çok Sınıflı Sınıflandırma, Dengesiz Veri, Yapay Örnekleme, LIME, SHAP

# ABSTRACT

## Ph.D. THESIS

PREDICTION A CREDIT SCORE WITH ANALYSIS OF DEBT PAYMENT TRANSACTIONS USING MACHINE LEARNING ALGORITHMS

**Cem BULUT**

**İstanbul University-Cerrahpaşa**

**Institute of Graduate Studies**

**Department of Computer Engineering**

**Computer Engineering, PhD Program**

**Supervisor : Assoc. Prof. Dr. Emel ARSLAN**

With the rapid expansion of the financial sector, minimizing financial risks has become increasingly crucial. Among these risks, credit use, the most common form of individual borrowing, takes the centre stage. Credit providers use systems that predict the repayment risk of the applicant to decide whether to grant credit to the applicant or not. In Turkey, the Credit Registry Bureau (KKB) provides the widely used credit score. However, as this score primarily evaluates risks based on past payment behaviour, it can sometimes fall short in offering a comprehensive assessment.

This study introduces an alternative credit scoring model using machine learning (ML) and explainable artificial intelligence (XAI) techniques. Unlike previous studies, this thesis does not rely on credit or credit card history for credit score estimation. Instead, it analyses individual bill payment habits using real transaction data from a bill payment institution in Turkey, combined with actual KKB scores. The methodology includes data cleaning, scaling, artificial sampling, classification algorithms, cross-validation, and XAI techniques. The model was validated through multiple evaluation metrics, resulting in a novel approach to credit scoring.

The proposed machine learning model holds significant importance, particularly for the application of machine learning models in the banking and finance sector. It is also important for users to understand and interpret the complex models.

The model presented in this study offers an alternative scoring method for individuals without a traditional credit score. For those with an existing credit score, it can serve as a complementary tool to enhance the current score or as a means of verification. ]

February 2025, [116] pages.

**Keywords:**[ Machine Learning, Credit Score, Explainable Artificial Intelligence, Multi-Class Classification, Imbalanced Data, Artificial Sampling, LIME, SHAP]

# 1. GİRİŞ

Bilgisayar sistemlerinin gelişmesiyle birlikte veri depolama ve işleme yeteneği artmaktadır. Mühendislik, istatistik ve yöneylem araştırması gibi alanlarda yeni yöntem ve algoritmalar geliştirilmekte ve uygulanmaktadır. Büyük veri kavramının yaygınlaşması ve süper bilgisayarların kullanılmasıyla birlikte geleneksel yöntemler kullanarak gerçekleştirilen işlemlerin birçoğunun makine öğrenmesi teknikleri ile kontrolü konusunda araştırmalar yapılmaktadır. Finans sektörü, makine öğrenmesi ile ilgili çalışmalar açısından önemli ve gerekli bir alan olarak öne çıkmaktadır.

Kredi skoru modelleri, finans sektörünün en önemli konularından biridir. Son yıllarda yurt içi ve yurt dışında yaşanan ekonomik krizlerle birlikte, kredi riski değerlendirme modellerinin önemi daha çok artmıştır. Güvenilir kredi skoru tahmini ile kuruluşlar, olası riskleri önceden belirleyerek, risk yaratacak unsurları azaltmak veya ortadan kaldırılmak istemektedir.

Kredi skoru konusunda ilk istatistik yaklaşım Fisher tarafından 1936 yılında yapılmıştır [1]. Hemen ardından David Duran, krediye başvuranların özelliklerine göre iyi veya kötü müşteriler olarak ayrılabileceğini öne sürmüştür [2]. 1960'larda kredi kartlarının ortaya çıkması ve bankacılık sektörünün büyümesiyle birlikte kredi risk değerlendirmenin önemi de artmıştır.

Finans kuruluşların risk analizi için kullandığı birçok yöntem bulunmaktadır. İstatistiksel veya farklı parametrelere dayanan yöntemlerin yetersiz kaldığı veya alternatif yöntemlerin doğrulanmak istenmesi gibi durumlarda makine öğrenmesi yaklaşımları iyi bir alternatif olarak öne çıkmaktadır. Literatürde bu konuda çeşitli çalışmalar yapılmasına karşın geliştirmeye açık birçok konu bulunmaktadır.

Literatürdeki çalışmaların çoğu yalnızca ikili kredi sınıflandırmasındaki performanslarına odaklanmıştır. Bu yüzden finansal risk yönetimi açısından verimliliği ve pratikliği tartışma konusudur. Çok sınıflı sınıflandırıcı kullanmak, ikili sınıflandırıcının çıktısından daha güvenilir ve değerli olabilir. Kredi veren kuruluşlara yönelik düzenleyici kuruluşlar, basit ikili kredi sınıflandırmasındaki performanstan ziyade kredi derecelendirmesine sahip yöntemlere ihtiyaç duymaktadır. Örneğin, kredi kuruluşları Uluslararası Finansal Raporlama Standartlarını

(International Financial Reported Standards- IFRS) takip etmesi halinde, her kredi notunda kredi zararı karşılıkları için temerrüt olasılığını (Probability of Default- PD) ve temerrüt halinde kayıp (Loss Given Default- LGD)' yi değerlendirmek üzere çok sınıflı bir kredi derecelendirmesi yapmaları gerekmektedir [3].

Literatürdeki ikili sınıflandırma teknikleri ile kredi skoru tahmin çalışmalarında başarıya ulaşmış tekniklerin, çok sınıflı veri setlerine uygulanabilirliği de araştırılması gereken bir konu olarak öne çıkmaktadır. Bu çalışmada Destek Vektör Makineleri (Support Vector Machine- SVM) ve Lojistik Regresyon (Logistic Regression- LR) gibi yalnızca ikili sınıflandırmada kullanılan algoritmaların, çok sınıflı verilere adapte edilerek kullanımı ve değerlendirmesi de yapılacaktır.

Literatürdeki diğer çalışmalar, esas olarak Almanya, Avustralya, Japonya ve diğer mevcut veri kümeleri üzerinde işlemler gerçekleştirmektedir. İkili kredi puanlama teorisinin ve uygulanmasıyla ilgili incelenen tüm makalelerin yaklaşık %45'inin Avustralya veya Almanya kredi veri setini kullandığı görülmüştür [4]. Her ne kadar bu veri kümeleri yapay zekada kıyaslama olarak görülebilse de sınırlı sayıda değişkene sahip oldukları ve kıyaslama için oluşturulmuş gerçekçi veriler olmadığı gerçek hayatta uygulanabilirliği tartışma konusudur. Makine öğrenimi ile uzman tabanlı modeller arasında doğrudan bir karşılaştırma yapmak oldukça önemlidir. Ayrıca kredi puanlama probleminde halka açık veri setlerini bulmanın hâlâ zor olduğu bilinmektedir [5].

Bununla birlikte, az sayıda çalışma gerçek hayattaki kredi puanlama veri setini kullanmıştır. Ancak bu veri setleri kullanıma açık olmadığından, yeni çalışmalara dahil edilme şansı yoktur [6] [7]. Tez çalışmamızda kullanılan veri seti, Türkiye'de yer alan ve ülkenin en büyük ödeme kuruluşlarından birine ait gerçek fatura ödeme verisidir. Bu ödeme kuruluşu aracılığıyla fatura ödemesi yapmış kişiler bilinmektedir. Bu kişilerin gerçek kredi skorları Kredi Kayıt Bürosu (KKB) tarafından hesaplanmıştır. KKB bir kişinin aldığı veya alacağı kredinin geri ödemesini başka bir kişiye göre ne ölçüde yerine getireceğini tahmin etmek amacıyla hesaplanan sayısal bir göstergedir. Bireysel Kredi Notu (BKN), istatistiksel modeller kullanılarak üretilmiş olan bir karar destek ürünü olarak tanımlanabilir. Fatura ödeme hareketleri ve KBB skorunun birleştirilmesiyle oluşturulan bu kombinasyon çalışmamızın benzersizliği ve önemli açısında oldukça önemlidir.

Kredi talebinde bulunan kişiler, bankalar veya finansal kurumlar tarafından risk değerlendirmesine tabii tutulurlar. Bu risk değerlendirmesi KKB tarafından yapılır ve kurumlarla paylaşılır. Fakat KKB gibi geleneksel kredi risk hesaplama yöntemleri, kişilerin geçmiş kredi veya kredi kartı ödemelerini referans alırlar. Eğer bu konuda herhangi bir geçmişi yoksa risk değerlendirmesi yapamaz. Çalışmamızda herhangi bir kredi geçmişi olmayan kişiler için makine öğrenmesi tabanlı bir değerlendirme yapılabilmesine olanak sağlayan bir öneri sunulmuştur. Bu açıdan bakıldığında finansal kuruluşların yeni müşterilerin risklerini öngörebilmesi ve değerlendirebilmesi açısından sunulan çalışma oldukça önemlidir.

Makine öğreniminde kullanılan modellerin karmaşıklığının artmasıyla birlikte hem tahminleri hem de tahmin sürecini anlama, güvenme, gerekçelendirme, iyileştirme ve makine öğrenimi sonuçlarını açıklamak daha da önem kazanmaktadır. Bu da açıklanabilirlik yöntemlerini değerlendirecek ve belirlenen görevleri yerine getirme yeteneklerini ölçecek mekanizmaların kullanılmasıyla sağlanmaktadır. Açıklanabilirlik ihtiyacının ilk hedefi, özellikle askeriye, bankacılık ve sağlık uygulamaları gibi hassas sektörlerdeki uygulamalar için insan tarafından yorumlanabilir modeller elde etmektir [8]. Bu çalışmada Açıklanabilir Yapay Zeka (Explainable Artificial Intelligence- XAI) modelleri kullanılarak, oluşturulan yüksek doğruluk elde edilen modellerin açıklanabilir ve yorumlanabilir hale getirilmiştir. Bu modellerin bankacılık ve finans sektörü açısından önemi vurgulanmıştır.

Özetlemek gerekirse, pek çok çalışmada kredi puanlama için çeşitli makine öğrenimi algoritmaları uygulanmış olsa da çoğunlukla performanslarını geleneksel yöntemlerle hesaplanmış skorlarla karşılaştırmamıştır. Çünkü bu karşılaştırma için mevcut kıyaslama veri seti nadirdir. Ek olarak, kullanılan gerçek veri setinin klasik kredi geçmişi veri seti olmaması da oldukça önemlidir. Ayrıca kullanılan modellerin Açıklanabilir Yapay Zeka teknikleri ile de açıklanabilir ve yorumlanabilir olarak sunulması akademik ve finans sektörü açısından umut vericidir.

Bu tez çalışmasının amaçları ve öncelikleri aşağıdaki gibi özetlenebilir:

- Alışılagelmiş kamuya açık kredi veri setleri yerine Türkiye'de yer alan bir ödeme kuruluşuna ait büyük veriden (big data) oluşan gerçek kişi ve fatura ödeme hareketlerinin kullanılması,

- Kredi skoru için fatura ödeme hareketleri baz alınarak ödeme alışkanlıkları analizi yapılıp, alternatif bir veri seti oluşturularak kullanılması,

- Oluşturulan fatura alışkanlıkları veri seti için, hedef sınıf olarak bu kişilere ait gerçek KKB skoru baz alınması dolayısıyla alternatif bir veri olarak kullanılan fatura ödeme alışkanlarının, geleneksel ve doğruluğu kanıtlanmış bir skor ile doğrulanması ve tamamının gerçek bilgileri yansıtması,

- Çok sınıflı ve dengesiz veri setleri ile çalışma yöntemlerinin belirlenmesi, bu yöntemlerin kullanılıp, etkilerinin değerlendirilmesi,

- İkili sınıflandırma için özel olarak tasarlanmış makine öğrenmesi algoritmalarının, çok sınıflı sınıflandırmada da kullanabilmesine olanak sağlayan alternatif yaklaşımların kullanılması ve değerlendirilmesi,

- Güçlendirme (Boosting) algoritmalarında, farklı güçlendirme tiplerinin kullanımı ve etkisinin araştırılması,

- Torbalama (Bagging) sınıflandırma yaklaşımında, zayıf öğretici olarak farklı sınıflandırma algoritmalarının kullanımı ve etkisinin araştırılması,

- XAI yaklaşımlarının, fatura ödeme alışkanlıkları ile kredi skoru tahminlemede kullanılması ve değerlendirilmesi,

- BKN olmayan kişiler için kredi skoru ihtiyacının giderilmesi veya varolan kredi skorunun zenginleştirilmesi,

- Kredi skoru için makine öğrenmesi tabanlı yeni bir yöntem geliştirilmesi ve sunulması.

# 2. GENEL KISIMLAR

## 2.1. Kredi Skoru (Credit Score)

Kredi, kişi veya kurumların çeşitli finansal ihtiyaçlarını karşılamak için bankalar veya finansal kuruluşlardan aldıkları borçlar olarak tanımlanır. Tatil, eğitim, araç, konut veya hizmet ihtiyaçları gibi çeşitli kişisel amaçlar için kullanılabilir [9].

Kredi skoru (CS), finansal kurumların kredi talep edenlerin risk seviyesini belirlemek için kullandığı bir ölçüttür. Kredi skoru ilk olarak, 1950lerde bu amaçla kullanılmaya başlanmıştır. Bu yıllarda kredi sağlayanlar, başvuru sahiplerinin mali durumlarını değerlendirmek için finansal geçmişlerini incelemeye başlamıştır. Zamanla, bu inceleme süreci makineleşmiş ve kredi puanı olarak bilinen bir sistem ortaya çıkmıştır. Kredi skoru, bir kişinin kredi risk durumunun değerlendirilmesini veya kişinin borçlarını geri ödeme olasılığını temsil eden bir değerdir ve bir kişinin kredi raporunun istatistiksel analizine dayanarak oluşturulur. Experian, Equifax ve TransUnion gibi kredi büroları, bir kişinin borçlanma ve geri ödeme faaliyetlerinin kaydını tutmaktadır. Bu kredi büroları, kişilerin finansal geçmişleri hakkında bilgi toplamakta ve bu bilgileri temel alarak kredi skorlarını belirlemektedir [10] [11].

Kredi skoru oluşturmak için birçok yöntem geliştirilmiştir. Her birinin kendine özgü metodolojisi ve kriterleri bulunmaktadır. Farklı kredi puanlama modellerini anlamak, bireylerin ve işletmelerin kredi ve krediler hakkında bilinçli kararlar almasına yardımcı olabilmektedir.

- **FICO Skoru:** Fair Isaac Corporation (FICO) tarafından geliştirilen ve en yaygın olarak kullanılan kredi skoru modelidir. FICO skoru, ödeme geçmişi, kredi kullanımı, kredi geçmişi uzunluğu, kredi türleri ve yeni kredi başvuruları gibi faktörlere dayanarak hesaplanmaktadır. Bu faktörler farklı yüzdelikler ile değerlendirilerek bir FICO puanı oluşturulur. FICO puanları 300 ila 850 arasında değişmektedir; temerrüt riski ile puan arasında ters orantı mevcuttur; temerrüt riski düştükçe puan yükselmektedir. Amerika Birleşik Devletleri (ABD)'nde en çok kullanılan kredi skorlama yöntemidir [12].

- **Vantage Skoru:** Vantage Score, FICO skorundan farklı olarak daha geniş bir puan aralığı kullanmakta ve kredi geçmişi olmayan kişileri de dahil ederek daha adil bir

değerlendirme yapmayı amaçlamaktadır. Vantage Skoru, kredi sağlayıcılarına ve diğer finansal kurumlara kişilerin kredi riskini değerlendirmede yardımcı olur. Bu sayede, kredi başvurularının değerlendirilmesi daha adil ve kapsamlı bir şekilde gerçekleşebilir. Kredi skoru noktasında farklılıklar olsa da genel olarak kredi sağlığınızı iyileştirmek için atılacak adımlar benzerdir. Düzenli ödemeler yapmak, borçları azaltmak ve kredi kartı kullanımını bilinçli bir şekilde yönetmek, her iki kredi skoru modeli için de olumlu sonuçlar doğurabilmektedir [13].

Türkiye'de kredi skoru hesaplanması, geleneksel yöntemler ile birlikte kişisel veriler gibi çeşitli kombinasyonlar kullanılarak hesaplanır. Türkiye'de kredi skoruna, Kredi Kayıt Bürosu (KKB) ve Findeks gibi kuruluşlar aracılığıyla erişilmektedir. Bu kuruluşlar, kişilerin kredi geçmişini, mevcut borçlarını, ödeme alışkanlıklarını ve diğer finansal bilgilerini toplayarak bir kredi skoru üretirler. KKB, Türkiye'deki tüm bankaların ve finansal kurumların kredi ve risk verilerini toplayan ve saklayan bir kurumdur. KKB, bireylerin kredi geçmişini izler ve bir kredi notu üretir. Bu not, bireyin kredi riskini belirlemeye yardımcı olur. Findeks ise KKB'nin sunduğu verilere dayanarak bireylerin kredi notunu ve kredi geçmişini sunan bir hizmettir. Findeks, bireylere kredi skorlarını ve kredi raporlarını çevrimiçi olarak sağlar. Bu kuruluşlar, bireylerin finansal durumlarını izleyerek kredi skorlarını hesaplarlar ve finansal kuruluşlara bu bilgileri sunarlar. Bu bilgiler, bankaların ve diğer finansal kurumların kredi başvurularını değerlendirirken kullanılır.

Geleneksel kredi skoru modelleri olan FICO ve VantageScore dışında, alternatif kredi skoru modelleri de geliştirilmektedir. Bazı kredi büroları geleneksel yöntemler ve farklı alternatifleri birleştirip özel skorlar oluşturabilmektedir. Ayrıca bazı şirketler alternatif veri kaynaklarından elde edilen bilgileri kullanarak kredi riskini değerlendiren modeller geliştirmektedir. Bu alternatif veri kaynakları, fatura ödeme geçmişi, kira ödemeleri, hatta sosyal medya aktiviteleri gibi çeşitli bilgileri içerebilir.

## 2.2. Makine Öğrenmesi (Machine Learning)

Makine öğrenmesi, bilgisayar sistemlerinin belirli bir görevi doğrudan programlanmış olmaksızın, veri üzerinden deneyim yoluyla öğrenmelerini sağlayan bir yapay zeka alanıdır. Makine öğrenimi algoritmaları, veriye dayalı modeller oluşturarak desenleri tanıma, tahmin

yapma, kategorize etme ve karar verme gibi görevleri yerine getirebilir. Sağlık, nesnelerin interneti, finans, sosyal medya gibi birçok kullanım alanı vardır [14].

Makine öğrenimi yöntemleri; Denetimli öğrenme (Supervised Learning), denetimsiz öğrenme (Unsupervised Learning), yarı denetimli (Semi-Supervised Learning) ve takviyeli öğrenme (Reiforcement Learning) olmak üzere dört kategoriye ayrılır:

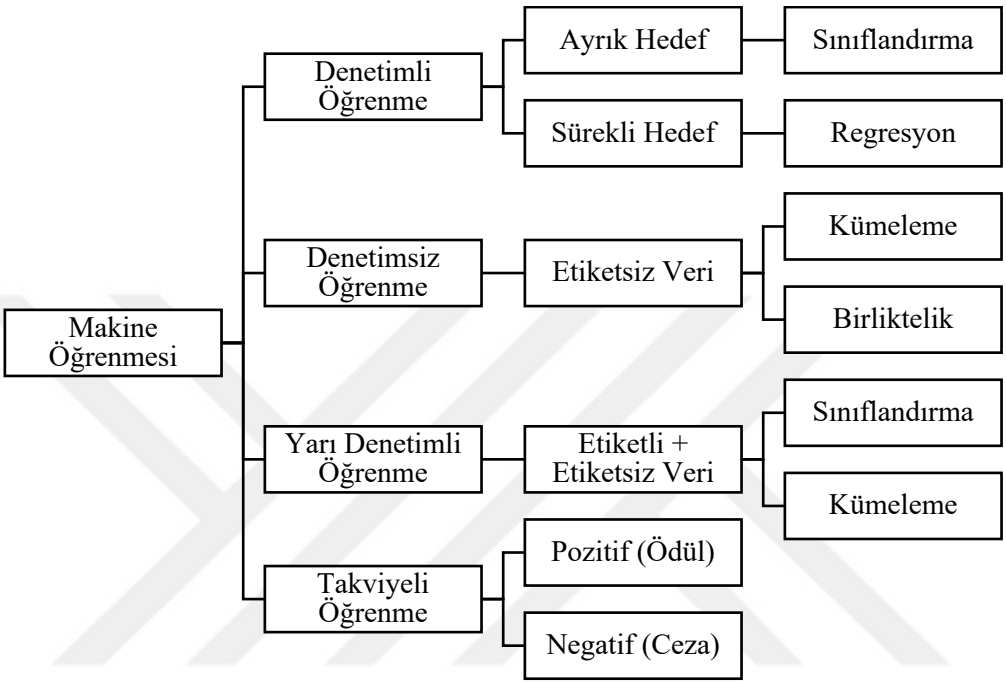
- **Denetimli Öğrenme:** Giriş verileri ve hedef çıktılar arasındaki ilişkiyi modellemek için kullanılır. Bu modeli oluşturmak için etiketlenmiş eğitim ve test örneği kullanılır. Denetimli öğrenme, belirli bir dizi girdi dizisinden, görev odaklı bir yaklaşımla gerçekleştirilecek belirli hedefler belirlendiğinde kullanılır [15].

- **Denetimsiz Öğrenme:** Etiketsiz veri setlerinin yapısını ve desenlerini analiz etmek için kullanılır. Bu sayede, veri setindeki gruplamaları, benzerlikleri veya anlamlı alt yapıları belirlenir. Bu, üretken özelliklerin çıkarılması, anlamlı eğilimlerin ve yapıların belirlenmesi, sonuçlardaki gruplamalar ve keşif amaçları için yaygın olarak kullanılmaktadır. En yaygın denetimsiz öğrenme görevleri kümeleme, yoğunluk tahmini, özellik öğrenme, boyutluluk azaltma, birliktelik kurallarını bulma, anormallik tespiti vb.dir [16].

- **Yarı Denetimli Öğrenme:** Hem etiketli hem de etiketsiz veriler üzerinde çalıştığı için denetimli ve denetimsiz yöntemlerin birleştirilmesi olarak tanımlanabilir. Yarı denetimli öğrenme modelinin nihai hedefi, tahmin için modelden tek başına etiketli veriler kullanılarak üretilenden daha iyi bir sonuç sağlamaktır. Yarı denetimli öğrenmenin kullanıldığı bazı uygulama alanları arasında makine çevirisi, sahtekarlık tespiti, veri etiketleme ve metin sınıflandırma yer alır [17].

- **Takviyeli Öğrenme:** Bulunduğu ortamı algılayan ve kendi başına kararlar alabilen bir sistemin, hedefine ulaşabilmesi doğru kararlar almayı öğrenebilmesi için geliştirilmiş yöntemleri ifade etmektedir. Takviyeli öğrenmede, bir hareketin karşılığı olarak makine veya yazılım, yeni durumun istem durumunu belirtmek için bir ödül veya ceza ile etmeni takviye işlemi gerçekleştirir. Böylece kurgulanan bu sistemde, asıl amaca ulaşmak için uygulanabilecek en iyi eylem seçilmeye çalışılır [18].

Makine öğrenmesi modelleri oluşturmak için öncelikle veri setinin kullanım amacı, türü ve hedef değişkeni incelenmelidir. Daha sonra Şekil 2.1'de gösterilen kriterler gösterilmektedir [17].




**Şekil 2.1.** Makine Öğrenmesi Modelleri.

## 2.3. Veri Ön İşleme (Data Preprocessing)

Özellikle büyük veri ile yapılacak çalışmalarda bilginin performansı yalnızca tasarlanacak modele bağlı değildir. Bunun yanında verilerin kalitesi ve uygunluğu da önemli parametrelerdir. Gürültülü, eksik, tutarsız ve gereksiz veriler, örneklerdeki ve özelliklerdeki büyük boyutlar gibi olumsuz faktörler, öğrenme ve bilgi çıkarma için kullanılan verileri olumsuz etkilemektedir. Düşük kaliteli verinin düşük kaliteli bilgiye yol açacağı iyi bilinmektedir. Bu nedenle veri ön işleme, asıl amacı daha sonraki veri madenciliği algoritmaları için doğru ve yararlı kabul edilebilecek son veri setlerini elde etmek olan önemli bir aşamadır [19].

### 2.3.1. Kusurlu Veriler (Imperfect Data)

Veri setinde çeşitli nedenlerle ortaya çıkan ve kusurlu olarak tanımlanabilen veriler yer alabilir. Bu veriler çeşitli alt kategorilere ayrılır ve farklı çözüm yolları ile işlenebilir.

- **Eksik Veri (Missing Data):** Veri ön işlemede eksik değerlerin dayattığı problemin üstesinden gelmek için birçok yaklaşım mevcuttur. İlk seçenek genellikle eksik değer içerebilecek örneklerin atılmasıdır. Fakat örneklerin ortadan kaldırılması öğrenme sürecinde bir yanlılığa veya önemli bilgilerin atılmasına sebep olabilir. İkinci seçenek, verilerin olasılık fonksiyonlarını modellerler ve eksikliğe neden olan mekanizmaları dikkate alırlar. Maksimum olabilirlik prosedürlerini kullanarak, eksik değerleri doldurmak için yaklaşık olasılıksal modelleri örneklerler. Belirli bir veri seti için gerçek olasılık modeli genellikle bilinmediğinden, herhangi bir ön bilgi vermeden kaçınılarak uygulanabildiği için makine öğrenmesi tekniklerinin kullanımı günümüzde oldukça popüler hale gelmiştir [20].

- **Gürültülü Veri (Noisy Data):** Gürültülü veriyle ilgili veri ön işleme literatüründe yaygın olarak iki ana yaklaşım bulunmaktadır. Bunlardan ilki, özellikle bir örneğin etiketlenmesini etkiliyorsa gürültüyü düzeltmektir. İkincisi , eğitim verilerindeki gürültülü örnekleri tanımlamak ve ortadan kaldırmaktır [21].

### 2.3.2. Boyutsal Problemler (Dimensional Problems)

Veri kümeleri tahmin değişkenlerinin sayısı veya örnek sayısı bakımından büyüdüğünde, veri madenciliği algoritmaları boyutluluk sorunu ile karşı karşıya kalır. Hesaplama maliyeti arttıkça çoğu veri madenciliği algoritmasının çalışmasını engelleyeceğinden ciddi bir sorundur. Bu sorun, Özellik Seçimi (Feature Selection- FS) ve uzay dönüşümüne dayalı yöntemler (Space Transformation Based Methods) ile çözülebilmektedir.

#### *2.3.2.1. Özellik Seçimi (Feature Selection)*

Özellik seçimi, veri setinin ilgili özelliklerini seçen, belirli özellik seçim kriterlerine göre orijinal bir özellik setinden bir alt küme elde etme işlemi olarak tanımlanır. Gereksiz ve ilgisiz özelliklerin ortadan kaldırıldığı veri işleme ölçeğinin sıkıştırılmasını sağlar. Özellik seçme tekniği, öğrenme algoritmalarını ön işleme tabi tutabilir ve iyi özellik seçimi sonuçları, öğrenme doğruluğunu artırabilir, öğrenme süresini azaltabilir ve öğrenme sonuçlarını basitleştirebilir. Özellikle, özellik seçimi ve özellik çıkarımı boyutluluğun azaltılmasının iki yoludur. Özellik seçiminin aksine, özellik çıkarmanın genellikle orijinal verileri güçlü örüntü tanıma yeteneğine sahip özelliklere dönüştürmesi gerekir; burada orijinal veriler, zayıf tanıma yeteneğine sahip özellikler olarak kabul edilebilir.

Özellik seçme algoritmalarının üç genel sınıfı vardır: Filtre (Filter) tabanlı, Sarmalayıcı (Wrapper) ve Gömülü (Embedded) yöntemler [22].

#### *2.3.2.1.1. Filtre Tabanlı Yöntemler*

Filtre tabanlı yaklaşımlar, belirli bir makine öğrenimi algoritmasının performansını dikkate almadan her bir özelliğin değerine göre değerlendirir. Verilerin birbirleri arasındaki ilişiklerine göre veri kümesinden özellik seçimi yapar. Filtre tabanlı yaklaşımlarda dört özellik seçme tekniği vardır. En çok kullanılan yöntemler ise aşağıda sıralanmıştır:

- **Bilgi Kazanımı (Information Gain):** Hedef değeri belirlemek ve entropi değerlerindeki azalmayı ölçmek için ilgili özellik tarafından sağlanan bilgi miktarı olarak tanımlanır. Her özelliğin bilgi kazancı, özellik seçimi için hedef değerler dikkate alınarak hesaplanır.

- **ANOVA F-Test:** ANOVA (Analysis of Variance), iki veya daha fazla veri örneğinden (genellikle üç veya daha fazla) elde edilen ortalamaların aynı dağılımdan gelip gelmediğini belirleyen parametrik istatistiksel hipotez testidir.

- **Karşılıklı Bilgi (Mutual Information):** Bu yöntem, iki değişkenin karşılıklı olarak bağımlı olup olmadığını ölçer ve böylece bir değişken için diğer değişken gözlemlendiğinde elde edilen bilgi miktarını sağlar. Bir özelliğin varlığına/yokluğuna bağlı olarak, o özelliğin hedef tahmininin yapılmasına katkı sağladığı bilgi miktarını ölçer.

- **Ki-Kare Testi (Chi-square Test):** İstatistiklerde iki olayın bağımsızlığını test etmek için ki-kare testi kullanılır. İki değişkenin verileri göz önüne alındığında, gözlemlenen sayım $O$ ve beklenen sayım $E$ elde edilebilir. Ki-Kare, beklenen sayım $E$ ile gözlemlenen sayım $O$ 'nun birbirinden nasıl saptığını ölçmektedir.

$$X_c^2 = \sum \frac{(O_i - E_i)^2}{E_i} \tag{2.1}$$

Burada, c, özgürlük derecesi; O, gözlemlenene değer(ler); E, beklenen değer(ler)’ dir.

#### *2.3.2.1.2. Sarmalayıcı Yöntemler*

Sarmalayıcı yöntemler, algoritmayı bir özellik alt kümesini yinelemeli bir şekilde kullanarak eğitir. Model öncesindeki eğitimden elde edilen sonuçlara dayanarak özelliklerin eklenmesi ve çıkarılması gerçekleşir. En iyi alt kümeyi seçmeye yönelik durdurma kriterleri, örneğin modelin performansı düştüğünde veya belirli sayıda özelliğe ulaşıldığında, genellikle modeli eğiten kişi tarafından önceden tanımlanmaktadır. Sarma yöntemlerinin filtre yöntemlerine göre temel avantajı, modelin eğitimi için en uygun özellikler kümesini sağlaması, dolayısıyla filtre yöntemlerinden daha iyi doğruluk sağlaması ancak hesaplama açısından daha pahalı olmasıdır.

#### *2.3.2.1.3. Gömülü Yöntemler*

Gömülü yöntemlerde özellik seçme algoritması, öğrenme algoritmasının bir parçası olarak harmanlanır, dolayısıyla kendi yerleşik özellik seçme yöntemlerine sahip olur. Gömülü yöntemler, filtre ve sarma yöntemlerinin dezavantajlarıyla karşılaşır ve avantajlarını birleştirir. Bu yöntemler, filtre yöntemleri gibi daha hızlıdır ve filtre yöntemlerinden daha doğrudur ve aynı zamanda özelliklerin bir kombinasyonunu da dikkate almaktadır.

#### *2.3.2.2. Uzay Dönüşümleri (Space Transformations)*

Uzay dönüştürme teknikleri, en umut verici özellikleri seçmek yerine, orijinal özellikleri birleştirerek tamamen yeni bir dizi özellik oluşturmaktadır. Farklı kriterlere uyularak böyle bir kombinasyon yapılabilir. Temel Bileşenler Analizi (Principal Component Analysis- PCA) bu amaçla en çok kullanılan yöntemlerden biridir [23].

### 2.3.3. Örnek Azaltma veya Arttırma

Örnek azaltma, orijinal veri setinin yerini alabilecek bir veri alt kümesinin seçilmesi ve aynı zamanda bir veri madenciliği uygulamasının amacını yerine getirebilmesi gereken bir dizi teknikten oluşur. Örnek arttırma ise, orijinal verilerden yeni yapay örnek oluşturur. Bu süreç, problemin etki alanındaki, orijinal verilerde temsili örnekler bulunmayan bölgelerin doldurulmasına veya büyük miktardaki örneklerin daha az örnekte yoğunlaştırılmasına olanak tanır [24].

### 2.3.4. Ayrıklaştırma ve Normalizasyon (Discretization and Normalization)

Ayrıklaştırma, sürekli değişkenleri ayrı aralıklar kullanarak dönüştürürken, normalleştirme yalnızca dağılımların ayarlanmasını gerçekleştirir [25].

- **İkilileştirici (Binarizer):** Sayısal özellikleri ikili özelliklere dönüştürür.
- **Kovalayıcı (Bucketizer):** Kovalar kullanarak bir dizi sürekli özelliği ayrıklaştırır.
- **Ayrık Kosinüs Dönüşümü (Discrete Cosine Transform):** Zaman alanındaki gerçek değerli bir diziyi, frekans alanındaki başka bir gerçek değerli diziye (aynı boyutta) dönüştürür.
- **Normalleştirici (Normalizer):** Her satırı birim norma sahip olacak şekilde normalleştirir. Kullanılan $p-$ normunu belirten $p$ parametresini kullanır.
- **Standart Ölçekleyici (Standard Scaler):** Her özelliği normal bir dağılım izleyecek şekilde normalleştirmektedir.
- **Minimum-Maksimum Ölçekleyici (Min-Max Scaler):** Alt ve üst sınır parametrelerini kullanarak her özelliği belirli bir aralığa normalleştirir.
- **Element-wise-Product:** Her özelliği bir skaler çarpanla ölçeklendirilmesidir.

### 2.4. Sınıflandırma (Classification)

Sınıflandırma, makine öğreniminde denetimli bir öğrenme yöntemi olarak kabul edilir ve aynı zamanda belirli bir örnek için bir sınıf etiketinin tahmin edildiği tahmine dayalı modelleme problemini referans alır. Matematiksel olarak, bir fonksiyonu $(f)$ giriş değişkenlerinden $(X)$ çıkış değişkenlerine $(Y)$ hedef, etiket veya kategori olarak eşler. Verilen veri noktalarının sınıfını tahmin etmek için yapılandırılmış veya yapılandırılmamış veriler üzerinde gerçekleştirilebilir [26].

- **İkili sınıflandırma (Binary Classification):** İki sınıf etiketine sahip sınıflandırma görevlerini ifade etmektedir (“Doğru ve yanlış” veya “evet ve hayır” gibi). Bu tür ikili sınıflandırma görevlerinde bir sınıf normal durum olabilirken anormal durum başka bir sınıf olabilir. Örneğin, bir e-postanın riskli veya risksiz olarak gruplandırılması, ikili sınıflandırma olarak kabul edilir.

- **Çok sınıflı sınıflandırma (Multi-Class Classification):** İkiden fazla sınıf etiketine sahip sınıflandırma görevlerini ifade eder. Her örnek yalnızca bir sınıf olarak etiketlenebilir [26].

- **Çok etiketli sınıflandırma (Multi-Label Classification):** Veri setinde yer alan bir örneğin, birkaç sınıf veya etiketle ilişkilendirilme durumunu ifade eder. Her örnek birden fazla sınıfa etiketlenebilir [17].

### 2.4.1. Karar Ağaçları (Decision Trees)

Karar Ağaçları (Decision Trees- DT), her bir iç düğümün özelliği, dalların kuralları ve yaprak düğümlerin algoritmanın sonucunu gösteren, akış şemasına benzer bir ağaç yapısıdır. Hem sınıflandırma hem de regresyon problemlerinde kullanılan, çok yönlü, denetimli bir makine öğrenme algoritmasıdır. Ayrıca Rastgele Orman'da eğitim verilerinin farklı alt kümeleri üzerinde eğitim vermek için de kullanılır. Bu da rastgele ormanı makine öğrenimindeki en güçlü algoritmalardan biri haline getirmektedir [27].

Karar ağaçları aşağıdaki adımları kullanır:

1. Ağaca, S veri kümesini içeren kök düğümle başlanır.

2. Özellik Seçim Ölçüsü kullanılarak veri kümesindeki en iyi özelliği bulunur.

3. S en iyi nitelikler için olası değerleri içeren alt kümelere bölünür.

4. En iyi özelliği içeren karar ağacı düğümünü oluşturulur.

5. Adım 3'te oluşturulan veri kümesinin alt kümelerini kullanarak yinelemeli olarak yeni karar ağaçları oluşturulur. Son düğüme gelene kadar bu aşamalar tekrarlanır.

### 2.4.2. Rastgele Orman (Random Forest)

Rastgele Orman (Random Forest- RF), karar ağaçlarına ve torbalamaya dayalı bir makine öğrenmesi algoritmasıdır. Karar Ağaçları, Sınıflandırma ve Regresyon Ağaçları, bir dizi bölme kuralıyla tahminin yapıldığı algoritmalardır. Bölme kuralları düğümlerle, bölme kuralı kararları dallarla ve son tahminler yapraklarla temsil edilmektedir. RF, karar ağaçlarını birleştirmek için torbalama ve rastgele alt uzay öznitelik seçimini yapar. RF modeli rastgeleliği iki aşamada kullanmaktadır: ilk olarak, orijinal veri kümesinden alt kümeleri rastgele seçer; ikinci olarak, orijinal tam özellik boyutlarından türetilen özellik alt kümelerini rastgele seçilir. Bu şekilde

ormandaki karar ağaçları arasındaki korelasyon azaltılır. Nihai karar, girdi örneğinin oyların çoğunluğunu alan sınıf olarak etiketleneceği oylama prosedürüne göre verilir [28] [29].

### 2.4.3. Lojistik Regresyon (Logistic Regression)

Lojistik regresyon (Logistic Regression- LR), basit bir parametrik istatistiksel yaklaşımdır ve kredi puanlama alanında endüstri standardı olarak kabul edilmektedir. İkili sınıflandırma problemlerini ve regresyon problemlerini çözmek için kullanılmaktadır. LR modelinin hedefi ilgilenilen iki olasılık sonucunun oranının logaritmasını elde etmektir [30].

$p$ bağımsız değişken için LR modeli şu şekilde yazılabilir:

$$P(Y = 1) = \frac{1}{1+e^{-(\beta_0+\beta_1 X_1+ \beta_2 X_2+\cdots+\beta_p X_p)}} \tag{2.2}$$

Burada $P(Y = 1)$ olasılık ve $\beta 0, \beta 1, \dots, \beta p$ regresyon katsayılarıdır. Lojistik regresyon modelinin içinde gizli bir doğrusal model vardır. $P(Y = 1)$'in $(1 - P(Y = 1))$'e oranının doğal logaritması $X_i$'de doğrusal bir model vermektedir:

$$g(x) = \ln\left(\frac{P(Y = 1)}{1 - P(Y = 1)}\right) - \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p \tag{2.3}$$

g(x), doğrusal bir regresyon modelinin arzu edilen özelliklerinin çoğuna sahiptir. Bağımsız değişkenler sürekli ve kategorik değişkenlerin birleşimi olabilir [31].

### 2.4.4. Naive Bayes

Özellikler veya değişkenler arasında bağımsızlık varsayımlarına sahip, Bayes teoremine dayanan basit bir olasılıksal sınıflandırıcıdır. Özellik değerleri ile sınıf arasındaki ilişkiler için koşullu bir olasılık türetmek amacıyla Navie Bayes (NB), her bir örnek için her özellik ile sınıf arasındaki ilişkiyi analiz etmektedir [32].

$x = (x^1, x^2, \dots, x^d)$ sınıf etiketi olmayan $d$ boyutlu bir örnek iken Bayes teoremine dayalı olarak bilinmeyen sınıf etiketini tahmin edecek bir sınıflandırıcı oluşturulmak istendiği varsayılsın. $C = \{C_1, C_2, \dots, C_k\}$ sınıf etiketlerinin kümesi olsun. $P(C_k)$ , yeni kanıttan önce çıkarılan $C_k$ (k $= 1, 2, \dots,$ k) 'nin önsel olasılığı olup $P(x| C_k)$, eğer $C_k$ hipotezi doğruysa, $x$ kanıtını görmenin koşullu olasılığı olsun. Aşağıdakileri elde etmek için Bayes teoremini kullanarak bu tür sınıflandırıcıları oluşturmaya yönelik bir tekniktir (2.4).

$$P(C^k|x) = \frac{P(x|C_k)P(C_k))}{\sum_{k'} P(x|C_{k'})P(C_{K'})} \tag{2.4}$$

NB, bir sınıfın belirli bir özelliğinin değerinin başka herhangi bir özelliğin değeriyle ilişkili olmadığını varsayar (2.5).

$$P(x|C_k) = \prod_{j=1}^{d} P(x^j|C_k) \tag{2.5}$$

### 2.4.5. K-En Yakın Komşu Algoritması (K-Nearest Neighbors)

Geleneksel $k$-En Yakın Komşu (K-Nearest Neighbors- K-NN) algoritması örüntü sınıflandırması için en eski ve en basit yöntemlerden biridir. $k$-NN, etiketlenmemiş her örneği, eğitim setindeki $k$ en yakın komşuları arasındaki çoğunluk etiketine göre sınıflandırır. Bu nedenle performansı, en yakın komşuları tanımlamak için kullanılan mesafe ölçüsüne önemli ölçüde bağlıdır. Ön bilginin yokluğunda çoğu $k$-NN sınıflandırıcısı, vektör girdileri olarak temsil edilen örnekler arasındaki farklılıkları ölçmek için basit Öklid metriğini kullanır [33]. Öklid mesafesi aşağıdaki formülle tanımlanır:

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^{n} w_r(a_r(x_i) - a_r x_j))^2} \tag{2.6}$$

Burada bir örneği $x = (a_1, a_2, \dots, a_n)$ vektörü olarak tanımlandığında; $n$, vektör girişinin boyutluluğu, yani bir örneğin niteliklerinin sayısıdır. $a_r$ örneğin $r$'ninci niteliğidir, $w_r$ ise $r$ 'ninci niteliğinin ağırlığıdır, $r$ 1'den $n$ 'ye kadardır, $d(x_i, x_j)$ ne kadar küçükse iki örnek o derece benzerdir [34].

Bir test örneğine atanan sınıf etiketi, en yakın $k$ komşusunun çoğunluk oyu ile belirlenir.

$$y(d_i) = arg \max_{\mathrm{k}} \sum_{x_j \in kNN} y(x_j, c_k) \tag{2.7}$$

Burada $d_i$ bir test örneğidir, $x_j$ eğitim setindeki $k$ en yakın komşularından biridir, $y(x_j, c_k)$, $x_j$'nin $c_k$ sınıfına ait olup olmadığını göstermektedir. Tahminin en yakın $k$ komşuda en çok üyeye sahip sınıf olacağı anlamına gelir. Örneğin, 7-en yakın komşu algoritmasını sınıflandırıcı yaparsak, bir örneğin en yakın 7 komşusundan beş tanesi "Birinci" sınıfa, diğer ikisi ise "İkinci"

sınıfa aitse, test örneğinin Birinci sınıfa ait olduğu sonucuna varılabilmektedir. Bir örneğin sınıf etiketi yalnızca en yakın komşusunun tanımlanmasıyla elde edildiğinde, algoritmaya en yakın komşu algoritması adı verilir.

### 2.4.6. Destek Vektör Makinesi (Support Vector Machine)

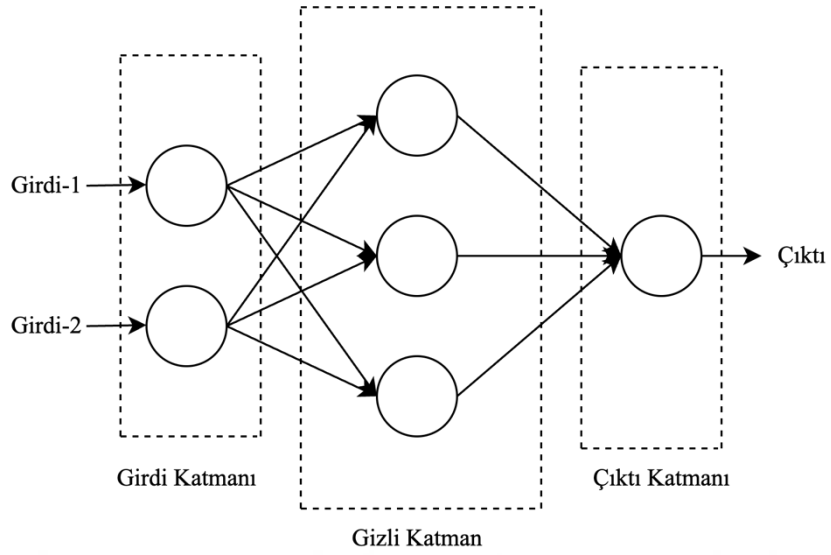
Destek Vektör Makinesi (Support Vector Machine- SVM) algoritması, veriyle ilgili birleşik dağılım fonksiyonu bilgisine ihtiyaç duymadan çalışır. Dolayısıyla dağılımdan bağımsız çalışabilme yeteneği olan bir öğrenme algoritmasıdır. SVM'nin temel prensibi, sınıfların birbirinden ayrılması için optimum ayırma hiper düzleminin bulunmasıdır. Yani farklı sınıfların destek vektörleri arasındaki uzaklığı büyütmektedir. Bir destek vektör makinesi, verileri optimum bir şekilde iki gruba ayıran $n$ boyutlu bir hiper düzlem oluşturur.

SVM, doğrusal verilerin iki sınıfa ayrılması için tasarlanmış olsa da doğrusal olmayan ve çok sınıflı veriler üzerinde de kullanılmaktadır. Doğrusal ayrılabilen veriler için genellikle iki sınıfa ayrılabilen örnek veriler kullanılarak eğitilir. Eğitim sonucunda elde edilen karar fonksiyonu ile verilerin birbirinden ayrılması amaçlanır [35] [36].

### 2.4.7. Çok Katmanlı Algılayıcı (Multi Layer Perception)

Çok Katmanlı Algılayıcı (Multi Layer Perception- MLP), herhangi bir giriş boyutunu istenilen boyuta dönüştüren, tamamen birbirine bağlı yoğun katmanlardan oluşur. Çok katmanlı bir algılayıcı, birden fazla katmana sahip bir sinir ağıdır. Bir sinir ağı oluşturmak için nöronlar bir araya getirilir, böylece bazı nöronların çıktıları diğer nöronların girdileri olur.

Çok katmanlı bir algılayıcıda bir giriş katmanı ve her giriş için bir nöron (veya düğüm) vardır, her çıkış için tek bir düğüm içeren bir çıkış katmanı vardır ve herhangi bir sayıda gizli katmana sahip olabilir. Çok Katmanlı Algılayıcı, bir veri kümesi üzerinde eğitim alarak bir $f(.): R^m \rightarrow R^o$ fonksiyonunu öğrenen denetimli bir öğrenme algoritmasıdır; burada $m$, girdi için boyut sayısıdır ve $o$, çıktı için boyut sayısıdır [37]. Bir dizi özellik $X = x_1, x_2, \ldots, x_n$ ve bir $y$ hedefi verildiğinde, sınıflandırma veya regresyon için doğrusal olmayan bir fonksiyon yaklaşıklayıcısı öğrenebilir. Giriş ve çıkış katmanı arasında gizli katmanlar adı verilen bir veya daha fazla doğrusal olmayan katman bulunabilmesi açısından lojistik regresyondan farklıdır. Şekil 2.2, tek gizli katmanlı bir MLP'yi göstermektedir.




**Şekil 2.2.** MLP Akışı

Girdi katmanı olarak bilinen en soldaki katman, giriş özelliklerini temsil eden bir dizi nörondan $\{x_i|x_1, x_2, \dots, x_m\}$ oluşur. Gizli katmandaki her nöron, önceki katmandaki değerleri ağırlıklı doğrusal toplam $w_1x_1 + w_2x_2 + \cdots + w_mx_m$ ile dönüştürür ve doğrusal olmayan bir aktivasyon fonksiyonu $g(.): R \rightarrow R$ gelir. Çıkış katmanı, son gizli katmandan değerleri alır ve bunları çıkış değerlerine dönüştürür [38].

### 2.4.8. Ekstra Ağaç Sınıflandırıcı (Extra Tree Classifier)

Ekstra Ağaç Sınıflandırıcı (Extra Tree Classifier- EXT) Rastgele Ormanlara benzer şekilde, çok sayıda karar ağacını eğiten ve bir tahmin çıktısı almak için karar ağaçları grubundan sonuçları bir araya getiren bir topluluk yaklaşımıdır. Ancak Ekstra Ağaçlar ile Rastgele Orman arasında bazı farklar vardır.

Rastgele Orman, karar ağaçlarının yeterince farklı olmasını sağlamak amacıyla eğitim verilerinin farklı varyasyonlarını seçmek için torbalamayı kullanır. Ancak Ekstra Ağaçlar karar ağaçlarını eğitmek için veri kümesinin tamamını kullanır. Bu nedenle, bireysel karar ağaçları arasında yeterli farklılıkları sağlamak için, bir özelliğin bölüneceği ve alt düğümlerin oluşturulacağı değerleri rastgele seçer. Bunun tersine, Rastgele Orman'da açgözlü arama yapmak ve bir özelliğin bölüneceği değeri seçmek için bir algoritma kullanırız. Bu iki fark dışında Rastgele Orman ve Ekstra Ağaçlar büyük ölçüde aynıdır [39].

---

1 **Dugumu_Bol**(S)

*Girdi:* Bölmek istediğimiz düğüme karşılık gelen yerel öğrenme alt kümesi S
*Çıktı:* $a < a_c$ veya hiçbir şey
    Bölmeyi_Durdur(S) DOĞRU ise hiçbir şey döndürme.
    Aksi takdirde, tüm sabit olmayan (S cinsinden) aday nitelikleri arasından K niteliğini {$a_1$, ..., $a_k$} seç;
    K bölmelerini çiz$\{S_i, \dots, S_k\}$, $S_i =$ Rastgele_Bolme_Sec (S, $a_i$) , $\forall_i = 1, \dots, K$;
    $Skor\ (s_*, S) = max_{i=1,\dots,K} Skor(s_i, S)$ olacak şekilde bir $s_*$ bölmesi döndür.
**Rastgele_Bolme_Sec** (S,a)
*Girdi:* S'nin bir alt kümesi ve a özniteliği
*Çıktı:* Bir düğüm
    $a^S_{max}$ ve $a^S_{min}$, a'nın S'deki maksimum ve minimum değerinin göstersin;
    $\left[a^S_{min}, a^S_{max}\right]$ içinde rastgele bir $a_c$ kesme noktası çiz;
    Bölmeyi döndür $[a < a_c]$.
**Bolmeyi_Durdur**(S)
*Girdi:* S'nin bir alt kümesi
*Çıktı:* Doğru veya Yanlış
    Eğer $[|S| < n_{min}]$ ise, DOĞRU değerini döndür;
    Eğer S'de tüm nitelikler sabitse, DOĞRU değerini döndür;
    Eğer çıkış S'de sabitse, DOĞRU değerini döndür;
    Aksi halde, YANLIŞ değerini döndür.

### 2.4.9. Güçlendirme (Boosting) Yöntemleri

#### *2.4.9.1. Uyarlanabilir Yükseltme (Adaptive Boosting)*

Uyarlanabilir Yükseltme (Adaptive Boosting- AdaBoost)' un temel prensibi, bir dizi zayıf öğreniciyi (yani, küçük karar ağaçları gibi rastgele tahminden sadece biraz daha iyi olan modelleri) verilerin tekrar tekrar değiştirilmiş versiyonlarına sığdırmaktır. Hepsinden gelen tahminler daha sonra nihai tahmini oluşturmak için ağırlıklı çoğunluk oyu (veya toplamı) yoluyla birleştirilir. Her güçlendirme yinelemesindeki veri değişiklikleri, eğitim örneklerinin her birine $w_1, w_2, \dots, w_N$ ağırlıklarının uygulanmasından oluşur. Başlangıçta bu ağırlıkların tümü $w_i = 1/N$ olarak ayarlanmıştır, böylece ilk adım zayıf bir öğrenciyi orijinal veriler üzerinde eğitir. Her ardışık yineleme için örnek ağırlıklar ayrı ayrı değiştirilir ve öğrenme algoritması yeniden ağırlıklandırılmış verilere yeniden uygulanır. Belirli bir adımda, önceki

adımda başlatılan güçlendirilmiş model tarafından yanlış tahmin edilen eğitim örneklerinin ağırlıkları artırılırken, doğru tahmin edilenlerin ağırlıkları azaltılır. Yinelemeler ilerledikçe, tahmin edilmesi zor örneklerin etkisi giderek artıyor. Böylece her bir sonraki zayıf öğrenci, dizideki öncekilerin kaçırdığı örneklere konsantre olmaya zorlanır [40] [41] [42]. AdaBoost için seçilebilecek zayıf öğreniciler; RF, DT, EXT, LR ve NB olabilir.

#### *2.4.9.2. Gradyan Arttırma (Gradient Boosting)*

Güçlendirme algoritmaları, zayıf öğrenenleri, yani rastgele öğrenenlerden biraz daha iyi olan öğrenenleri yinelemeli bir şekilde güçlü bir öğrenen olarak birleştirir. Gradyan Artırma (Gradient Boosting- GB), regresyon için artırmaya benzer bir algoritmadır. Bir eğitim veri seti $D = \{x_i, y_i\}_1^N$ verildiğinde, gradyan artırmanın amacı, belirli bir kayıp fonksiyonunun $L(y, F(x))$ beklenen değerini en aza indirerek x örneklerini çıkış değerleri y ile eşleştiren $F^*(x)$ fonksiyonunun bir $\hat{F}(x)$ yaklaşımını bulmaktır. Gradyan Artırma, fonksiyonların ağırlıklı toplamı olarak $F^*(x)$ 'ın toplamsal bir yaklaşımını oluşturur [43].

$$F_m(x) = F_{m-1}(x) + p_m h_m(x) \tag{2.8}$$

Burada $p_m$ m'ninci fonksiyonun ağırlığıdır, $h_m(x)$. Bu fonksiyonlar, karar ağaçları gibi topluluk modellerdir. Yaklaşım yinelemeli oluşturulur. Öncelikle $F^*(x)$ şu şekilde elde edilir:

$$F_0(x) = arg \min_{\alpha} \sum_{i=1}^{N} L(y_i, \alpha) \tag{2.9}$$

Sonraki modellerin minimize edilmesi beklenir:

$$(p_m, h_m(x)) = arg \min_{\mathrm{p,h}} \textstyle\sum_{i=1}^{N} L(y_i, F_{m-1}(x_i) + ph(x_i)) \tag{2.10}$$

Bununla birlikte, optimizasyon problemini doğrudan çözmek yerine, her $h_m$ için $F^*$ için gradyan düşürme optimizasyonunda açgözlü bir adım olarak görülebilir. Bunun için her $h_m$ modeli, yeni bir veri kümesi $D = \{x_i, r_{mi}\}_{i=1}^N$ üzerinde eğitilir; burada $r_{mi}$'ler, şu şekilde hesaplanır:

$$r_{mi} = \left[\frac{\partial L(y_i, F(x))}{\partial F(x)}\right]_{F(x)=F_{m-1}(x)} \tag{2.11}$$

$p_m$ 'in değeri daha sonra bir satır arama optimizasyon probleminin çözülmesiyle hesaplanır.

### *2.4.9.3. Aşırı Gradyan Arttırma (Extreme Gradient Boosting- XGBoost)*

Aşırı Gradyan Artırma (Extreme Gradient Boosting- XGBoost), yüksek düzeyde ölçeklenebilir olacak şekilde tasarlanmış, gradyan artırmaya dayalı bir karar ağacı topluluğudur. Gradyan artırmaya benzer şekilde XGBoost, kayıp fonksiyonunu en aza indirerek amaç fonksiyonunun ilave bir genişlemesini oluşturur. XGBoost'un temel sınıflandırıcı olarak yalnızca karar ağaçlarına odaklandığı göz önüne alındığında, ağaçların karmaşıklığını kontrol etmek için kayıp fonksiyonunun bir varyasyonu kullanılır.

$$L_{xgb} = \sum_{i=1}^{N} L(y_i, F(x_i)) + \sum_{m=1}^{M} \Omega(h_m) \tag{2.12}$$

$$\Omega(h) = \gamma T + \frac{1}{2}\lambda||w||^2 \tag{2.13}$$

Burada $T$ ağacın yaprak sayısıdır ve $w$ yaprakların çıktı puanlarıdır. Bu kayıp fonksiyonu, bir ön budama stratejisine yol açan karar ağaçlarının bölünmüş kriterine entegre edilebilir. $\gamma$'nin daha yüksek değerleri daha basit ağaçlarla sonuçlanır. $\gamma$'nin değeri, dahili bir düğümü bölmek için gereken minimum kayıp azaltma kazancını kontrol eder. XGBoost'taki ek bir düzenleme hiper parametresi, eklemeli genişletmedeki adım boyutunu azaltan küçültmedir. Son olarak, ağaçların karmaşıklığı, ağaçların derinliği vb. gibi diğer stratejiler kullanılarak da sınırlanabilir. Ağaç karmaşıklığının azaltılmasının ikincil bir faydası, modellerin daha hızlı eğitilmesi ve daha az depolama alanı gerektirmesidir. XGBoost'ta hem aşırı uyumu azaltmak hem de eğitim hızını artırmak için rastgeleleştirme teknikleri de uygulanmaktadır. XGBoost'ta yer alan rastgeleleştirme teknikleri, ağaç ve ağaç düğümü düzeylerinde ayrı ayrı ağaçları ve sütun alt örneklemesini eğitmek için rastgele alt örneklerdir. Ayrıca, XGBoost, gradyanı ve hessian'ı (ikinci dereceden gradyan) çıkaran bir fonksiyon tanımlayarak ve bunu "objektif" hiper parametreden geçirerek kullanıcı tanımlı herhangi bir kayıp fonksiyonuna genişletilebilir [44].

### *2.4.9.4. RF Destekli XGBoost*

XGBoost normalde gradyan güçlendirilmiş karar ağaçlarını ve diğer gradyan ile güçlendirilmiş modelleri eğitmek için kullanılır. Rastgele Ormanlar, gradyan destekli karar ağaçlarıyla aynı model temsilini ve çıkarımını kullanır, ancak farklı bir eğitim algoritmasıdır. Bağımsız bir

rastgele ormanı eğitmek için XGBoost kullanılabilir veya rastgele orman, gradyan artırma için temel model olarak kullanılabilir. Bu çalışmada bağımsız rastgele ormanları eğitmek için XGBoost'tan faydalanılacaktır.

#### *2.4.9.5. Hafif Gradyan Arttırma Makinesi (Light Gradient Boosting Machine)*

Hafif Gradyan Arttırma Makinesi (Light Gradient Boosting Machine- LGBM), büyük ölçekli verileri doğru bir şekilde işlemek, verileri etkili bir şekilde hızlandırmak ve bellek kullanımını azaltmak için yeni bir Gradyan Tabanlı Tek Taraflı Örnekleme (Gradient-based One-Side Sampling- GOSS) ve Özel Özellik Paketleme (Exclusive Feature Bundling- EFB) tekniğini kullanır.

- **Gradyan Tabanlı Tek Taraflı Örnekleme:** Farklı veri örneklerinin bilgi kazanımının hesaplanmasında farklı rolleri vardır. Daha büyük gradyanlara sahip örnekler (yani az eğitilmiş örnekler) bilgi kazanımına daha fazla katkıda bulunacaktır. GOSS, büyük gradyanlara sahip örnekleri (örneğin, önceden tanımlanmış bir eşikten daha büyük veya en üst yüzdelikler arasında) tutar ve bilgi kazanımı tahmininin doğruluğunu korumak için küçük gradyanlara sahip örnekleri yalnızca rastgele bırakır. Bu işlem, özellikle bilgi kazancının değeri geniş bir aralığa sahip olduğunda, aynı hedef örnekleme oranıyla, tekdüze rastgele örneklemeden daha doğru bir kazanç tahminine yol açabilir.

  GOSS yöntemi, $n$ örnekli $\{x_1,\ x_2,\cdots, x_n\}$ içeren bir eğitim kümesinde gradyan artırmada kullanılır; burada her örnek $x_i,\ X^s$ uzayında $s$ boyutuna sahip bir vektördür. Gradyan artırmanın her yinelemesinde, modelin çıktısına göre kayıp fonksiyonunun negatif gradyanları $\{g_1, g_2, \dots, g_n\}$ olarak temsil edilir. Eğitim setindeki örnekler, mutlak gradyan değerlerine göre azalan sırada sıralanır ve en büyük gradyanlara sahip üst $a \times$ %100 örnekler, bir $A$ alt kümesi oluşturmak üzere seçilir. Daha küçük gradyanlara sahip $(1-a)/b \times$ %100 örneklerden oluşan geri kalan $A_c$ kümesi için, $b \times |Ac|$ boyutunda rastgele bir $B$ alt kümesi vardır. Örnekler daha sonra $V_j(d)$ vektöründe tahmin edilen varyans kazancına dayalı olarak bölünür [45].

$$\tilde{V}_j(d) = \frac{1}{n}\left(\frac{\left(\sum_{x_i \in A_l} g_i + \frac{1-a}{b}\sum_{x_i \in B_l} g_i\right)^2}{n_l^j(d)} + \frac{\left(\sum_{x_i \in A_r} g_i + \frac{1-a}{b}\sum_{x_i \in B_r} g_i\right)^2}{n_r^j(d)}\right) \quad (2.14)$$

Burada $A_l = \{x_i \in A : x_{ij} \leq d\}$, $A_r = \{x_i \in A : x_{ij} > d\}$, $B_{l1} = \{x_i \in B : x_{ij} \leq d\}$, $B_r = \{x_i \in B : x_{ij} > d\}$ ve $(1-a)/b$ katsayısı, B üzerindeki gradyanların toplamını $A_c$ boyutuna geri döndürmek içim kullanılmaktadır.

LBGM algoritmasının avantajları aşağıdaki gibi sıralanabilir:

- LBGM algoritması daha hızlı ve daha yüksek doğruluğa sahiptir
- Daha düşük Bellek kullanımına sahiptir
- Daha iyi doğruluğa sahiptir
- Paralel ve dağıtılmış GPU öğrenimini destekler
- Büyük ölçekli verileri işleyebilmektedir

- **Özel Özellik Paketleme Tekniği:** Yüksek boyutlu veriler genellikle çok seyrektir ve bu da bize özellik sayısını azaltmak için neredeyse kayıpsız bir yaklaşım tasarlama olanağı sağlar. Spesifik olarak, seyrek bir özellik uzayında birçok özellik birbirini dışlar, yani hiçbir zaman aynı anda sıfırdan farklı değerler almazlar. Ayrıcalıklı özellikler tek bir özellikte güvenli bir şekilde birleştirilebilir (Özel Özellik Paketi olarak adlandırılır). Dolayısıyla histogram oluşturmanın karmaşıklığı O(veri × özellik)'ten O(veri × paket)'e değişirken paket<<özellik. Dolayısıyla eğitim çerçevesinin hızı, doğruluktan ödün vermeden artırılır.

Bu çalışmada LGBM sınıflandırması için arttırma yöntemi olarak; Gradyan Arttırıcı Karar Ağacı (Gradient Boosting Decision Tree- GBDT), Bırakılanların Çoklu Eklemeli Regresyon Ağaçlarıyla Buluşması (Dropouts meet Multiple Additive Regression Trees- DART) ve Rastgele Orman kullanılacaktır.

### 2.4.10. Torbalama (Bootstrap Aggregating- Bagging)

Torbalama (Bagging), birden fazla temel modelin, eğitim verilerinin farklı alt kümeleri üzerinde bağımsız ve paralel olarak eğitildiği bir topluluk öğrenme türüdür. Her alt küme, veri noktalarının değiştirilerek rastgele seçildiği önyükleme örneklemesi kullanılarak oluşturulur. Torbalama sınıflandırıcısı durumunda, nihai tahmin, çoğunluk oyu kullanılarak tüm temel modelin tahminlerinin toplanmasıyla yapılır. Regresyon modellerinde nihai tahmin, tüm temel modelin tahminlerinin ortalaması alınarak yapılır ve buna torbalama regresyonu denir.

Torbalama, özellikle yüksek varyansa sahip modellerde doğruluğu artırmaya ve aşırı uyumu azaltmaya yardımcı olur [46] [28] [47].

Torbalama yöntemi aşağıdaki adımlardan oluşur:

- **Bootstrap Örnekleme:** Orijinal eğitim verileri, rastgele $n$ alt küme ile değiştirilerek örneklenir. Bu adım, bazı örnekler yeni alt kümede birden çok kez görünebileceğinden, diğerleri atlanabileceğinden, temel modellerin farklı veri alt kümeleri üzerinde eğitilmesini sağlar. Aşırı uyum riskini azaltır ve modelin doğruluğunu artırır.

- **Temel Model Eğitimi:** Torbalamada birden fazla temel model kullanılır. Önyükleme örneklemesinden sonra, her temel model, karar ağaçları, destek vektör makineleri veya farklı bir önyüklemeli veri alt kümesindeki sinir ağları gibi belirli bir öğrenme algoritması kullanılarak eğitilir . Bu modellerin kendi başlarına her zaman yüksek doğruluk oranlarına sahip olmadığından, "zayıf öğrenenler" olarak adlandırılır. Model geliştirilirken zaman ve maliyet açısından verimli hale getirmek için temel modeller paralel olarak eğitilebilir .

- **Toplama:** Tüm temel modeller eğitildikten sonra, temel model tarafından eğitilmeyen veri alt kümesi, tahminlerde bulunmak için kullanılır. Torbalama sınıflandırıcısında, verilen örnek için tahmin edilen sınıf etiketi çoğunluk oylamasına göre seçilir. Çoğunluk oyu alan sınıf modelin öngörüsüdür.

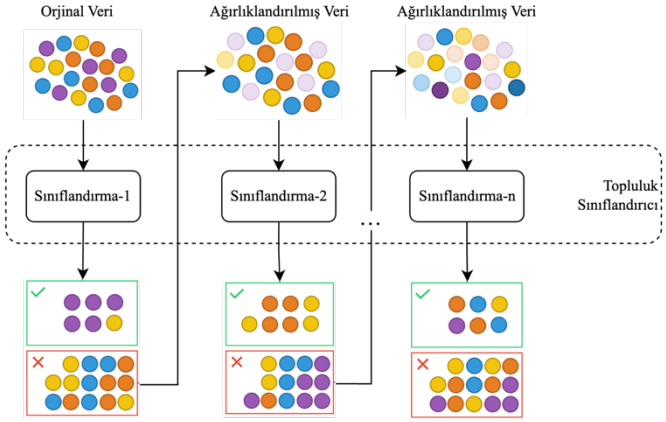
- **Çanta Dışı (Out of Bag) Değerlendirme:** Bazı örnekler, önyükleme yöntemi sırasında belirli temel modellerin eğitim alt kümesinin dışında bırakılır. Bu "çanta dışı" örnekler, çapraz doğrulamaya gerek kalmadan modelin performansını tahmin etmek için kullanılabilir.

- **Nihai Tahmin:** Tüm temel modellerden gelen tahminleri topladıktan sonra Bagging, her örnek için nihai bir tahmin üretir.

### 2.4.11. Boosting ve Bagging Yöntemlerinin Karşılaştırılması

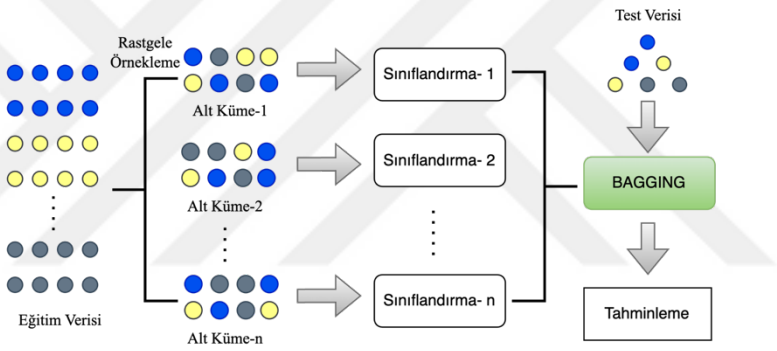
Bagging ve Boosting, yaygın olarak kullanılan ve topluluk yöntemleri olarak sınıflandırma konusunda evrensel bir benzerliğe sahiptir.

Boosting yöntemlerinin çalışma prensibi aşağıdaki şekilde gösterilmiştir:




**Şekil 2.3:** Boosting Çalışma Prensibi

Bagging yönteminin çalışma mantığı ise aşağıdaki gibidir:




**Şekil 2.4:** Bagging Çalışma Prensibi

Bu yöntemler arasındaki benzerlik aşağıdaki gibi özetlenebilir:

- Her ikisi de 1 öğrenciden N öğrenci elde etmeye yönelik topluluk yöntemleridir.
- Her ikisi de rastgele örnekleme yoluyla çeşitli eğitim veri setleri üretir.
- Her ikisi de N öğrencinin ortalamasını alarak (veya çoğunluk oyunu alarak) nihai kararı verir.
- Her ikisi de varyansı azaltmada iyidir ve daha yüksek stabilite sağlar.

Bu yöntemler arasındaki farklar aşağıdaki tabloda açıklanmıştır:

**Tablo 2.1:** Bagging ve Boosting Yaklaşımlarının Karşılaştırılması

| Bagging | Boosting |
|---|---|
| Aynı türe ait tahminleri birleştirmenin bir yoludur. | Farklı türlere ait tahminleri birleştirmenin bir yoludur. |
| Önyargıyı değil, varyansı azaltmayı hedefler. | Varyansı değil, önyargıyı azaltmayı hedefler. |
| Her model eşit ağırlık alır. | Modeller performansa göre ağırlıklandırılır. |
| Her model bağımsız olarak inşa edilmiştir. | Yeni modeller daha önce oluşturulmuş modellerin performansından etkilenir. |
| Farklı eğitim verisi alt kümeleri, tüm eğitim veri kümesinden değiştirmeli satır örnekleme ve rastgele örnekleme yöntemleri kullanılarak seçilir. | Her yeni alt küme, önceki modellerde yanlış sınıflandırılan unsurları içerir. |
| Bagging aşırı uyum sorununu çözmeye çalışır. | Boosting önyargıyı azaltmaya çalışır. |
| Sınıflandırıcı kararsızsa (yüksek varyans) torbalama uygulanabilir. | Sınıflandırıcı kararlı ve basitse (yüksek önyargı) artırmayı uygulanabilir. |
| Sınıflandırıcılar paralel olarak eğitilir. | Sınıflandırıcılar sırayla eğitilir. |
| Rastgele Orman, Baging'i kullanır. | AdaBoost, Boosting tekniklerini kullanır. |

### 2.4.12. İkili Sınıflandırma Algoritmalarının Çok Sınıflı Sınıflandırmada Kullanımı

Sınıf sayısının ikiden fazla olmasına çok sınıflı sınıflandırma denir. Örneğin; bir müzik platformunda yer alan şarkı türlerinin klasik, elektronik, pop ve caz olarak kategorize edildiğini düşünelim. Bu örnekte bir adet sınıf etiketi (müzik türü) ve dört adet sınıf bulunmaktadır.

Lojistik regresyon ve destek vektör makineleri gibi yöntemler yalnızca ikili sınıflandırma problemi için tasarlanmıştır. Bu gibi yöntemlerin çok sınıflı sınıflandırma işlemlerinde de kullanımı için iki yaklaşım bulunmaktadır: bire karşı hepsi (one-vs-all) ve bire karşı bir (one-vs-one).

#### *2.4.12.1. Bire-karşı-hepsi yaklaşımı (one-vs-all)*

Bu yöntemdeki temel mantık, $n$ sınıftan oluşan çok sınıflı bir sınıflandırma probleminin, $n$ adet ikili sınıflandırma problemine bölünmesidir. Müzik türü örneğinde; bu veriyi modellemek için her sınıf için dört ayrı ikili sınıflandırıcı oluşturulabilir. İlk sınıflandırıcı, müzik türünün klasik olup olmadığını bir miktar güven değeriyle tahmin edecektir. Benzer şekilde, ikinci

sınıflandırıcı da müzik türünün elektronik olup olmadığını vb. tahmin edecektir. Son aşamada ise tahminin güven değerinin en yüksek olduğu sınıfı seçecektir.

#### *2.4.12.2. Bire-karşı-bir yaklaşımı (one-vs-one)*

Bire-karşı-bir önceki yaklaşıma benzer bir strateji sunmaktadır. Fakat burada, her sınıf için farklı ikili sınıflandırıcıları birbirine karşı, tüm benzersiz kombinasyonlar dikkate alınarak eğitilir. Bu yaklaşım için, yine müzik türü örneğinden yola çıkarsak, aşağıdaki gibi altı adet ikili sınıflandırıcı elde ederiz:

**Tablo 2.2:** Bire-karşı-bir yaklaşımı sonucu elde edilen ikili sınıflandırıcı

| | |
|---|---|
| Sınıflandırıcı 1: | Klasik-Elektronik |
| Sınıflandırıcı 2: | Klasik-Pop |
| Sınıflandırıcı 3: | Klasik-Caz |
| Sınıflandırıcı 4: | Elektronik-Pop |
| Sınıflandırıcı 5: | Elektronik-Caz |
| Sınıflandırıcı 6: | Pop-Caz |

Örnekten de anlaşılacağı üzere n sınıftan oluşan çok sınıflı bir sınıflandırma için ikili sınıflandırıcı sayısı $n * (n-1)/2$ ve karmaşıklığı $O(n^2)$ olacaktır. Dolayısıyla bire-karşı-hepsi yaklaşımına göre daha yavaştır. Ancak bu yöntem, çekirdek algoritmaları gibi iyi ölçeklenemeyen algoritmalar için avantajlı olabilir. Bu yaklaşım yalnızca ikili sınıflandırma için tasarlanmış modellerde kullanılabileceği gibi farklı modellerde de kullanılabilir [48].

#### *2.4.12.3. Hata Düzeltme Çıktı Kodu tabanlı stratejiler*

Hata düzeltme çıktı koduna dayalı stratejiler, bire-karşı-bir ve bire-karşı-hepsi yaklaşımlarından oldukça farklıdır. Bu stratejilerle her sınıf, her boyutun yalnızca 0 veya 1 olabileceği bir öklid uzayında temsil edilir. Yani her sınıf bir ikili kodla (0 ve 1 dizisi) temsil edilir. Her sınıfın konumunu/kodunu takip eden matrise kod kitabı denir. Kod boyutu yukarıda bahsedilen alanın boyutluluğudur. Sezgisel olarak, her sınıf mümkün olduğunca benzersiz bir kodla temsil edilmeli ve sınıflandırma doğruluğunu optimize etmek için iyi bir kod kitabı tasarlanmalıdır.

Uydurma zamanında, kod kitabındaki bit başına bir ikili sınıflandırıcı takılır. Tahmin zamanında, sınıflandırıcılar sınıf uzayındaki yeni noktaları yansıtmak için kullanılır ve bu noktalara en yakın sınıf seçilir. 0 ile 1 arasındaki bir sayı, bire karşı geri kalana göre daha az sınıflandırıcı gerektirecektir. Teorik olarak $\log_2(n)/n$ her sınıfı açık bir şekilde temsil etmek için yeterlidir. Ancak pratikte $\log_2(n)$ $n$ 'den çok daha küçük olduğundan iyi bir doğruluğa yol açmayabilir. 1'den büyük bir sayı, bire karşı geri kalana göre daha fazla sınıflandırıcı gerektirecektir. Bu durumda, bazı sınıflandırıcılar teorik olarak diğer sınıflandırıcıların yaptığı hataları düzeltecektir, dolayısıyla "hata düzeltme" adı da buradan gelmektedir. Ancak pratikte sınıflandırıcı hataları genellikle birbiriyle ilişkili olacağından bu gerçekleşmeyebilir. Hata düzelten çıktı kodları torbalamaya benzer bir etkiye sahiptir [49] [50].

## 2.5. Dengesiz Veri Problemi (Imbalanced Data Problem)

Sınıf dağılımı eşit olmayan herhangi bir veri seti teknik olarak "dengesiz" (imbalanced data) olarak tanımlanır. Bununla birlikte, problemin her sınıfına ait örneklerin sayısı arasında aşırı orantısızlık olduğunda, bir veri kümesinin dengesiz olduğu sonucuna ulaşılır. Başka bir deyişle sınıf dengesizliği, bir sınıfı temsil eden örnek sayısının diğer sınıflara göre çok daha az olması durumunda ortaya çıkar. Bu nedenle, bir veya daha fazla sınıf veri kümesinde yetersiz temsil edilebilir. Sınıflandırıcılar çoğunluk sınıfına daha yatkın karar vererek performansın gerçekçilikten uzaklaşmasına neden olmaktadır [51]. İkili sınıf dengesizliği probleminde bir azınlık sınıfı ve bir çoğunluk sınıfı vardır. Öte yandan, çok sınıflı dengesizlik probleminde hem azınlık hem de çoğunluk sınıflarının birden fazla örneği olabilir. Sınıf dengesizliği problemi literatürde çok tartışılan ve önem verilen bir konudur ve bu konuda çeşitli yaklaşımlar mevcuttur.

Literatürde sınıf dengesizliği problemine veri ve algoritma düzeyinde olmak üzere iki yaklaşım öne çıkmaktadır. Veri düzeyi yaklaşımları, yeniden örnekleme yöntemleriyle veri dağılımını dengelemeyi amaçlar. Bu yöntemler, azınlık sınıfına ait örnekleri artırarak veya çoğunluk sınıfı örneklerini azaltarak sorunu çözmeye çalışır. Algoritma düzeyindeki yaklaşımlar, makine öğrenimi algoritmasını dengesizlik verilerini barındıracak şekilde değiştirir. Değişiklik, maliyete duyarlı yöntemler (cost-sensitive) olarak da bilinen, her sınıf için farklı yanlış sınıflandırma maliyetlerini dikkate alarak doğruluk oranını maksimuma çıkarmak yerine maliyet hatasını en aza indirebilir [52]. Diğer değişiklik ise uygun bir endüktif önyargının (inductive bias) seçilmesidir. Örneğin, öğrenici olarak bir karar ağacını kullanırken yaprakta

olasılık tahminini ayarlamak veya birliktelik kurallarında farklı sınıflar için çoklu minimum desteği dikkate almak şeklinde ifade edilebilir. Öğrenme algoritmalarında topluluk yaklaşımları ve derin tabanlı algoritmalar gibi azınlık sınıfı örneklerine yönelik ve bunları güçlendirmeyi amaçlayan birçok başka yaklaşım önerilmiştir.

Çoklu sınıflandırıcı sistemi olarak da bilinen topluluk sınıflandırıcılar (ensemble classifiers), sınıflandırma sistemi için temel sınıflandırıcıların bir koleksiyonunu birleştirerek öğrenme yönteminin performansını artırmaktadır. Her bir temel sınıflandırıcının çıktısı toplanır ve yeni örnekler için sınıflandırma kararının verilmesinde kullanılır. Spesifik olarak, güçlendirmede (boosting), sınıflandırıcının performansına göre değişen örnek ağırlıklarına dayalı olarak sınıflandırıcı üretmenin her aşamasında yanlış sınıflandırılmış örneklere odaklanarak, eğitim verilerinin ağırlıklı versiyonları üzerinde sıralı bir temel sınıflandırıcı kümesi oluşturulur [53].

Dengesiz veri problemleri için birçok strateji mevcuttur. Örneklemeye dayalı yaklaşım en etkili yöntemlerdendir. Yapay örneklemeye dayalı yaklaşım üç kategoriye ayrılmaktadır. Aşırı Örnekleme (Over Sampling), Yetersiz Örnekleme (Under Sampling) ve Hibrit Örnekleme (Hybrid Sampling) [54] [55].

### 2.5.1. Aşırı Örnekleme (Over Sampling)

Aşırı örnekleme algoritmaları çoğunluk sınıfındaki örneklerle ilgilenmez, ancak sınıflandırma performansını iyileştirmek için azınlık sınıfındaki örnekleri artırır.

#### *2.5.1.1. Rastgele Aşırı Örnekleme (Random Over Sampling)*

Orijinal örneklerin tekrarlanmasıyla veri kümesinin boyutunu arttırılmasını amaçlayan yöntem Rastgele Aşırı Örnekleme (Random Over Sampling- ROS) olarak adlandırılır. Rastgele örnekleyici yeni örnekler oluşturmadığından, örneklerin çeşitliliği değişmez.

$R^d$ 'de yer alan $X$ alanlarına odaklanıldığında ve P(x) = f(x), $X$ üzerinde bir olasılık yoğunluk fonksiyonudur. $y_j$ 'nin boyutu $n_j < n$ iken ( $j = 0,1$ ) ROS, aşağıdaki adımlardan oluşur [56]:

1. $\frac{1}{2}$ olasılıkla $y = y_j \in y$ seçilir.
2. $p_i = \frac{1}{n_j}$ olasılıkla, $y_i = y$ olacak şekilde $T_n$ 'de $(x_i, y_i)$ seçilir.

3. $K_{H_j}(., x_i)$'den örnek $x$, $K_{H_j}$ ile $x_i$ merkezli ve ölçek parametrelerinin $H_j$ matrisine bağlı bir olasılık dağılımıdır.

#### *2.5.1.2. Sentetik Azınlık Aşırı Örnekleme Tekniği (SMOTE)*

SMOTE (Synthetic Minority Over Sampling Technique) olarak kısaltılan bu teknik, yeni örnekler üreterek veri kümesindeki azınlık örneklerin sayısını artıran istatistiksel bir yöntemdir. Azınlık sınıfından yalnızca bazı rastgele örnekleri kopyalayan rastgele aşırı örneklemeden farklı olarak SMOTE, her verinin mesafesine (genellikle Öklid mesafesini kullanarak) ve azınlık sınıfının en yakın komşularına dayalı olarak örnekler üretir, dolayısıyla oluşturulan örnekler orijinal azınlık sınıfından farklıdır [57].

SMOTE yöntemi aşağıdaki adımları takip etmektedir:

1. Azınlık sınıfından rastgele verileri seçilir.
2. Rastgele veriler ile $k$ en yakın komşuları arasındaki Öklid mesafesini hesaplanır.
3. Farkı 0 ile 1 arasında rastgele bir sayıyla çarpılır ve ardından sonucu sentetik örnek olarak azınlık sınıfına eklenir.
4. Azınlık sınıfının istenen oranı karşılanıncaya kadar ilk üç adım tekrarlanır.

#### *2.5.1.3. Sınırda SMOTE (Borderline SMOTE)*

SMOTE yönteminin bir türevi olan Borderline SMOTE göre, örneklerin ve komşularının, sınır çizgisine uzak olanlara göre yanlış sınıflandırılma olasılığı daha yüksektir. Bu yöntem, azınlık örneklerini güvenli, tehlikeli ve gürültülü olmak üzere üç gruba bölmek için her bir azınlık örneğinin çoğunluk komşularının sayısını kullanılmaktadır. Tehlike grubunun yeni örnekler oluşturmak için kullanıldığına dikkat edilmelidir [58].

#### *2.5.1.4. SVM-SMOTE*

Mevcut azınlık sınıfı örneklerinden en yakın komşularına doğru yönlendirmelerle birlikte yeni azınlık sınıfı örneklerini üretmeyi amaçlayan bir yöntemdir. SVM-SMOTE, sınıflar arasında sınırların belirlenmesine yardımcı olmak için SVM modelini kullanarak sınır çizgilerinin yakınında yeni azınlık sınıfı örnekleri oluşturmaya odaklanır [59].

#### *2.5.1.5. K-Ortalama SMOTE (K-Means SMOTE)*

K-Means SMOTE, gürültü oluşumunu önleyen ve sınıflar arasındaki ve içindeki dengesizliklerin etkili bir şekilde üstesinden gelen, $k$-ortalama kümeleme ve SMOTE aşırı örneklemeye dayanan basit ve etkili bir aşırı örnekleme yöntemidir. SMOTE 'u uygulamadan önce $k$-ortalama kümeleme yöntemini kullanır. Kümeleme, örnekleri bir arada gruplayacak ve küme yoğunluğuna bağlı olarak yeni örnekler üretecektir [60].

#### *2.5.1.6. Uyarlanabilir Sentetik Örnekleme (Adaptive Synthetic Sampling Approach-ADASYN)*

Uyarlanabilir Sentetik Örnekleme Yaklaşımı (Adaptive Synthetic Sampling Approach-ADASYN), azınlık veri örneklerinin dağılımlarına göre uyarlanabilir şekilde üretilmesi temeline dayanmaktadır. Öğrenilmesi daha zor olan azınlık sınıfı örnekleri, öğrenilmesi daha kolay olan örneklerden daha fazla sentetik veri oluşturmak için kullanılır; bu da başlangıçta dengesiz veri dağıtımı nedeniyle ortaya çıkan öğrenme yanlılığının azaltılmasına yardımcı olur. SMOTE algoritmasında her bir azınlık sınıfı için oluşturulan sentetik örneklerin sayısı aynıdır, oysa ADASYN algoritmasında yoğunluk dağılımı her bir azınlık sınıfı örneği için oluşturulması gereken sentetik örneklerin sayısına otomatik olarak karar vermek için kullanılır. Algoritmanın temel işleyişi, her örnek için farklı miktarlarda sentetik veri üretmek amacıyla farklı azınlık sınıfı örneklerine ağırlıklar atamasıdır. ADASYN yöntemini adımları aşağıdaki gibidir [61]:

1. Azınlık sınıfının çoğunluğa oranı hesaplanır:

$$d = \frac{m_s}{m_l} \tag{2.15}$$

Burada $m_s$ ve $m_l$ azınlık ve çoğunluk sınıfı örneklerinin sayısını temsil etmektedir. Eğer $d$ belirli bir eşikten düşükse algoritma başlatılır.

2. Oluşturulacak sentetik azınlık verilerinin toplam sayısını hesaplanır:

$$G = (m_l - m_s)\beta \tag{2.16}$$

Burada $G$, üretilecek azınlık verilerinin toplam sayısıdır. ß, ADASYN' den sonra istenen azınlık:çoğunluk verilerinin oranıdır. ß =1, ADASYN' den sonra mükemmel dengelenmiş bir veri seti anlamına gelir.

3. Her bir azınlık örneğinin $k$-en yakın komşuları bulunur ve $r_i$ değerini hesaplanır. Bu adımdan sonra her azınlık örneğinin farklı bir komşuyla ilişkilendirilir.

$$r_i = \frac{\#majority}{k} \tag{2.17}$$

$r_i$ değeri, her bir komşu çoğunluk sınıfının hakimiyetini gösterir. Daha yüksek $r_i$ komşular daha fazla çoğunluk sınıfı örneği içerir ve öğrenilmesi daha zordur.

4. Tüm $r_i$ değerlerinin toplamı 1'e eşit olacak şekilde $r_\text{i}$ değerlerini normalleştirilir.

$$\widehat{r_i} = \frac{r_i}{\sum \widehat{r_i}} \tag{2.18}$$

$$\sum \widehat{r_i} = 1 \tag{2.19}$$

5. Komşu başına üretilecek sentetik örneklerin miktarı hesaplanır.

$$G_i = G\,\widehat{r_i} \tag{2.20}$$

Çoğunluk sınıfı örneklerinin hakim olduğu komşularda $r_\text{i}$ daha yüksek olduğundan, bu komşular için daha sentetik azınlık sınıfı örnekleri üretilecektir. Dolayısıyla bu, ADASYN algoritmasına uyarlanabilir doğasını verir; "öğrenilmesi daha zor" komşular için daha fazla veri üretilir.

6. Her komşu için $G_\text{i}$ verileri oluşturulur. İlk olarak komşu için azınlık örneğini ele alınsın, $x_\text{i}$. Ardından, o mahalledeki başka bir azınlık örneğini rastgele seçilir $xz_\text{i}$. Yeni sentetik örnek aşağıdakiler kullanılarak hesaplanabilir:

$$s_i = x_i + (x_{zi} - x_i) \lambda \tag{2.21}$$

Yukarıdaki denklemde λ, 0-1 arasında rastgele bir sayıdır, $s_\text{i}$ yeni sentetik örnektir, $x_\text{i}$ ve $xz_\text{i}$ aynı mahalledeki iki azınlık örneğidir.

### 2.5.2. Yetersiz Örnekleme (Under Sampling)

Yetersiz örnekleme, baskın sınıfların boyutunu azaltarak çoğunluk ve azınlık sınıflarını dengelemeyi amaçlar. İşlem sırasında kaybedilen veri miktarı büyük ölçüde, bilgiyi silmek için

kullanılan yönteme göre belirlenir. Yetersiz örnekleme stratejilerine göre şu şekilde gruplandırılabilir: Prototip oluşturma yöntemleri ve Prototip seçme yöntemleri.

- **Prototip oluşturma yöntemleri:** *S* orijinal bir veri seti olarak düşünüldüğünde, prototip oluşturma algoritmaları, $|S'| < |S|$ ve $S' \not\subset S$ 'nin bulunduğu yeni bir *S'* seti üretecektir. Yani prototip oluşturma teknikleri hedeflenen sınıflardaki örneklerin sayısını azaltacaktır. Fakat geri kalan örnekler orijinal kümeden üretilecektir. Küme Merkezleri (Cluster Centroids) bu yöntemlerdendir.

- **Prototip seçme yöntemleri:** Prototip seçim algoritmaları, orijinal S kümesinden örnekleri seçecek ve $|S'| < |S|$ ve $S' \subset S$ 'nin bulunduğu bir $S'$ veri kümesi üretecektir. Başka bir deyişle S', S 'nin bir alt kümesidir. Prototip seçim algoritmaları, kontrollü eksik örnekleme teknikleri ve yetersiz örneklemeyi temizleme teknikleri olmak üzere iki gruba ayrılır. Kontrollü yetersiz örnekleme yöntemleri, çoğunluk sınıfındaki veya sınıflarındaki gözlem sayısını kullanıcı tarafından belirlenen isteğe bağlı sayıda örnek sayısına düşürür. Tipik olarak gözlem sayısını azınlık sınıfında gözlemlenen örneklerin sayısına düşürürler. Random Under Sampling ve Near Miss bu kategoride yer almaktadır. Buna karşılık, yetersiz örnekleme tekniklerinin temizlenmesi, yönteme bağlı olarak "gürültülü" veya "sınıflandırılması çok kolay" gözlemleri kaldırarak özellik alanını "temizler". Her sınıftaki nihai gözlem sayısı temizleme yöntemine göre değişir ve kullanıcı tarafından belirlenemez. Tomek's Links ve en yakın komşuyu kullanan algoritmalar (Edited Nearest Neighbours, Repeated edited nearest neighbours, All-KNN, Condensed Nearest Neighbours, One Sided Selection, Neighbourhood Cleaning Rule) bu başlık altındadır.

#### *2.5.2.1. Küme Merkezleri (Cluster Centroids)*

Azınlık sınıfını yeniden örneklemek yerine, çoğunluk sınıfındaki örneklerin sayısını azalmaya yönelik bir yaklaşımdır. Bu örneklemeyi, çoğunluk sınıfı örneklerinden oluşan bir kümeyi kümenin ağırlık merkeziyle değiştirerek yapar [62].

Bu yöntemin uygulanma süreci aşağıdaki gibidir:

1. Çoğunluk sınıfı verileri üzerinde, $k$-ortalama kümeleme yöntemi uygulanır.

2. Her kümenin ağırlık merkezi hesaplanır.

3. Çoğunluk sınıfına ait örneklerin azaltılması için bu ağırlık merkezlerini kullanılır.

#### *2.5.2.2. Rastgele Yetersiz Örnekleme (Random Under Sampling)*

Rastgele Yetersiz Örnekleme (Random Under Sampling- RUS), yetersiz örneklemenin en basit yöntemidir. Çoğunluk sınıfındaki bazı örnekleri rastgele atarak dengeli veri örneklerine ulaşmayı amaçlar. Ancak örneklerin rastgele atılması, çoğunluk örneklerindeki potansiyel olarak yararlı bilgileri silebilir ve bu da sınıflandırıcı performansını azaltabilir [63].

#### *2.5.2.3. Near Miss*

Near Miss, örnekleri seçmek için bazı sezgisel kurallara ve en yakın komşu algoritmasına dayalı bir yöntem benimser. Bu yöntem 3 farklı sezgisel parametre ile kullanılabilir. NearMiss-1, negatif sınıfın $N$ en yakın örneğine olan ortalama mesafenin en küçük olduğu pozitif örnekleri seçer. NearMiss-2, negatif sınıfın en uzak $N$ örneğine olan ortalama mesafenin en küçük olduğu pozitif örnekleri seçer. NearMiss-3 ise iki adımlı bir algoritmadır. İlk olarak, her negatif örnek için $M$ en yakın komşusu tutulacaktır. Daha sonra seçilen pozitif örnekler, N en yakın komşuya olan ortalama mesafenin en büyük olduğu örnektir [64].

#### *2.5.2.4. Tomek Bağlantıları (Tomek's Links)*

Farklı sınıflardan iki örnek birbirine en yakın komşu olduğunda Tomek bağlantısı oluşur. Farklı x ve y sınıflarından iki örnek arasındaki Tomek bağlantısı, herhangi bir $z$ örneği için şöyle tanımlanır. Burada $d$ iki örnek arasındaki mesafeyi ifade eder.

$$d(x,y) < d(x,z) \; ve \; d(x,y) < d(y,z) \tag{2.22}$$

Tomek Bağlantıları yaklaşımı, Tomek'in bağlantılarını algılar ve kaldırır. Temel fikir, Tomek'in bağlantılarının gürültülü veya gözlemleri sınıflandırmanın zor olduğu ve algoritmanın uygun bir ayrım sınırı bulmasına yardımcı olmayacağıdır [65]. Öncelikle, örnekler arasındaki komşuluk ilişkileri hesaplanır ve Tomek Bağlantıları oluşturulur. Gürültülü ve aykırı değer olarak belirlenen bu örnek veya örnekler veri setinden çıkarılır. Bu değerlerin temizlenmesi, daha iyi genelleme yapmaya yardımcı olur. Tomek Bağlantıları şu adımlardan oluşur:

1. A örneğinin en yakın komşusu B'dir.

2. B örneğinin en yakın komşusu A'dır.

3. A ve B örnekleri farklı bir sınıfa aittir. Yani, A ve B sırasıyla azınlık ve çoğunluk sınıfına aittir (veya tam tersi).

Tomek Bağlantıları çok sınıflı problemlerde de uygulanabilir. A, B ve C etiketlerinde oluşan üç sınıf olduğunu varsayarsak, aşağıdaki adımlar takip edilir.

1. A sınıfı ve diğer sınıflar (B, C) arasında Tomek Bağlantısı uygulanır.
2. B sınıfı ve diğer sınıflar (A, C) arasında Tomek Links uygulanır.
3. C sınıfı ve diğer sınıflar (A, B) arasında Tomek Links uygulanır.

#### *2.5.2.5. Düzenlenmiş En Yakın Komşular (Edited Nearest Neighbours- ENN)*

Düzenlenmiş en yakın komşular metodolojisi, hedeflenen sınıf örneklerinin komşularını tanımlamak için $k$-en yakın komşuları kullanır. Komşularından herhangi birinin veya çoğunun farklı bir sınıftan olması durumunda gözlemleri kaldırır [66].

Bu yöntem aşağıdaki adımlardan oluşur:

1. Veri kümesinin tamamını kullanılarak $k$-en yakın komşular eğitilir.
2. Yalnızca hedeflenen sınıflar için, her gözlemin $k$-en yakın komşusu tespit edilir.
3. Komşularından herhangi biri veya çoğu farklı bir sınıfa aitse gözlemleri kaldırılır.

#### *2.5.2.6. Tekrarlanan ENN (Repeated ENN- RENN)*

RENN, ENN'nin birden çok kez tekrarlayarak genişletilmiş versiyonudur. Genellikle algoritmanın tekrarlanması daha fazla veriyi siler. Kullanıcı, düzenlenen en yakın komşular yönteminin kaç kez tekrarlanması gerektiğini ayarlayabilir. Tekrarlar aşağıdaki durumlarda sonlandırılır [67]:

1. Maksimum yineleme sayısına ulaşıldığında,
2. Kaldırılacak örnek sayısı kalmadığında,
3. Çoğunluk sınıflarından birinin azınlık sınıfı haline geldiğinde,

4. Yetersiz örnekleme sırasında çoğunluk sınıflarından birini kaybolduğunda tekrarlanması durdurulur.

#### *2.5.2.7. ALL-KNN*

All-KNN, RENN' in her bir ENN turunda değerlendirilen komşu sayısının arttığı bir çeşididir. All-KNN, kullanıcı tarafından parametre aracılığıyla belirlenen, incelenecek maksimum komşu sayısına ulaşıldığında veya çoğunluk sınıfı azınlık sınıfı haline geldiğinde temizliği durdurur [67].

#### *2.5.2.8. Yoğunlaştırılmış En Yakın Komşular (Condensed Nearest Neighbours- CNN)*

CNN, bir örneğin kaldırılması gerekip gerekmediğine yinelemeli olarak karar vermek için $k$-en yakın komşu algoritmasını kullanır [68]. Algoritma şu şekilde çalışır:

1. C kümesinden azınlık örnekler alınır.
2. C'de hedeflenen sınıftan (düşük örneklenecek sınıf) bir örnek ve bu sınıfın diğer tüm örneklerini bir S kümesine eklenir.
3. C kümesi 1- NN ile eğitilir.
4. S kümesindeki örnekler, tek tek örneklenerek incelenir. Her biri en yakın komşu yöntemi (3. adımda eğitilen) kullanılarak sınıflandırılır.
5. Örnek yanlış sınıflandırılmışsa C'ye eklenir ve 6. adıma geçilir.
6. S'deki tüm gözlemler incelenene kadar 3'ten 5'e kadar olan adımları tekrarlanır.

Nihai veri seti S'dir ve azınlık sınıfından ve ardışık En Yakın Komşu algoritmaları tarafından yanlış sınıflandırılmış çoğunluktan gelen tüm gözlemleri içermektedir.

#### *2.5.2.9. Tek Taraflı Seçim (One Sided Selection- OSS)*

Tek Taraflı Seçim yöntemi, CNN tarafından sunulan gürültülü örnekleri ortadan kaldırmak amacıyla, öncelikle sınıflandırılması zor olan örnekler bulur ve gürültülü örnekleri kaldırmak için TomekLinks'i kullanmaktadır. Bu yöntem aşağıdaki adımlardan oluşur [68].

1. Azınlık sınıfına ait örnekler C kümesinden alınır.

2. C'de hedeflenen sınıftan (düşük örneklenecek sınıf) bir örnek ve bu sınıfın diğer tüm örneklerini bir S kümesine eklenir.

3. C kümesi 1- NN ile eğitilir.

4. Üçüncü adımda eğitilmiş veri, en yakın komşu yöntemi kullanılarak, S kümesindeki tüm örnekleri sınıflandırır.

5. Yanlış sınıflandırılmış tüm örnekler C'ye eklenir.

6. Tomek bağlantıları C'den kaldırılır.

Yukarıdaki adımlardan sonra nihai veri seti S'dir.

#### *2.5.2.10. Komşu Temizliği Kuralı (Neighbourhood Cleaning Rule- NCR)*

Verileri yoğunlaştırmak yerine temizlemeye odaklanır. Sınıflandırma sonuçlarının kalitesinin sınıfın büyüklüğünden ziyade gürültüsüz olmasına bağlı olduğunu savunur. Bu nedenle, sınıf dağılımının yanı sıra verilerin gürültü gibi sınıflandırmayı engelleyebilecek diğer özelliklerini de dikkate almıştır. Bu yöntem aşağıdaki adımları takip eder [69]:

1. T verisi ilgili C sınıfına ve geri kalanını O olarak bölünür.

2. ENN algoritması ile O içindeki gürültülü veri seti, $A_1$ olarak ayrılır.

3. For each class $C_i$ in O

if ($x \in C_i$ in 3-nearest neighbors of misclassified $y \in C$) and ( $|C_i|$ 0.5. $|C|$) then $A_2 = \{x\} \cup A_2$

4. İndirgenmiş veri $S = T - (A_1 \cup A_2)$

### 2.5.3. Hibrit Yöntemler (Hybrid Methods)

SMOTE yöntemi, marjinal aykırı değerler ile iç değerler arasında yeni noktaları enterpolasyona tabi tutarak gürültülü örnekler oluşturabilmektedir. Bu sorun, aşırı örneklemeden kaynaklanan alanın temizlenmesiyle çözülebilir.

Bu bağlamda, Tomek'in bağlantısı ve düzenlenmiş en yakın komşular (ENN), daha gürültüsüz veriler elde etmek için SMOTE aşırı örnekleme uygulandıktan sonra ardışık düzene eklenen iki

temizleme yöntemidir. Aşırı ve yetersiz örnekleme yöntemlerini birleştirmek için kullanıma hazır iki dengesiz öğrenme uygulaması şunlardır: SMOTE-Tomek ve SMOTE-ENN.

#### *2.5.3.1. SMOTE-Tomek*

Bu yöntem, azınlık sınıfı için sentetik veriler üretmeye yönelik SMOTE tekniğini ve Tomek bağlantıları olarak tanımlanan verileri çoğunluk sınıfından (yani çoğunluk sınıfından en yakın olan veri örnekleri) kaldırmaya yönelik Tomek Bağlantıları yeteneğini birleştirmektedir [70].

1. (SMOTE'un Başlangıcı) Azınlık sınıfından rastgele verileri seçilir.
2. Rasgele veri ile en yakın $k$- komşusu arasındaki mesafeyi hesaplanır.
3. Farkı 0 ile 1 arasında rastgele bir sayıyla çarpılır ve sonuç sentetik örnek olarak azınlık sınıfına eklenir.
4. Azınlık sınıfının istenen oranı karşılanıncaya kadar 2-3 numaralı adımları tekrarlanır. (SMOTE'un sonu)
5. (Tomek Bağlantılarının Başlangıcı) Çoğunluk sınıfından rastgele veriler seçilir.
6. Rastgele verinin en yakın komşusu azınlık sınıfından gelen veriler ise (yani Tomek Link'i oluşturulur), Tomek Link'i kaldırılır. (Tomek Bağlantılarının Sonu)

#### *2.5.3.2. SMOTE-ENN*

Bu yöntem, azınlık sınıfı için sentetik örnekler üretmeye yönelik SMOTE tekniğini ve gözlemin sınıfı ile $k$-en yakın komşu çoğunluğu arasında farklı sınıfa sahip olduğu tanımlanan her iki sınıftan bazı gözlemleri silmeye yönelik ENN yeteneğini birleştirir [71]. SMOTE-ENN aşağıdaki adımları takip eder:

1. (SMOTE'un Başlangıcı) Azınlık sınıfından rastgele verileri seçilir.
2. Rasgele veri ile en yakın $k$ komşusu arasındaki mesafeyi hesaplanır.
3. Farkı 0 ile 1 arasında rastgele bir sayıyla çarpılır, ardından sonucu sentetik örnek olarak azınlık sınıfına eklenir.

4. Azınlık sınıfının istenen oranı karşılanıncaya kadar 2-3 numaralı adımları tekrarlanır. (SMOTE'un sonu)

5. (ENN'nin Başlangıcı) En yakın komşuların sayısı olarak $k$ belirlenir. Belirlenmezse $k$=3 olarak atanır.

6. Veri kümesindeki diğer gözlemler arasında gözlemin $k$-en yakın komşusunu bulunur, ardından $k$-en yakın komşudan çoğunluk sınıfını döndürür.

7. Gözlemin sınıfı ile gözlemin $k$-en yakın komşusunun çoğunluk sınıfı farklıysa, bu durumda gözlem ve onun $k$-en yakın komşusu veri setinden silinir.

8. Her sınıfın istenen oranı karşılanıncaya kadar 2. ve 3. adımlar tekrarlanır. (ENN'nin sonu)

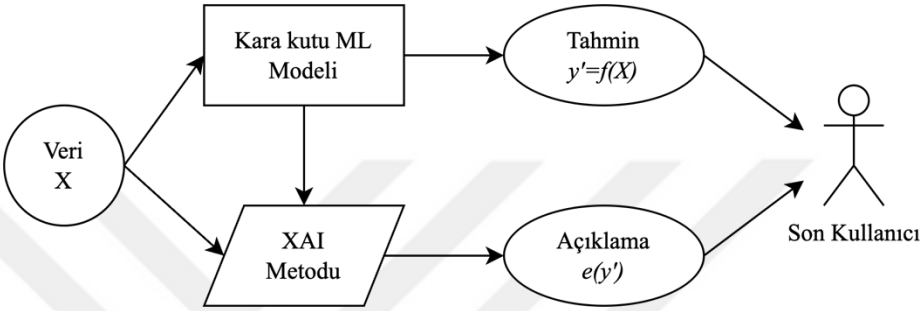
## 2.6. Açıklanabilir Yapay Zeka (Explainable Artificial Intelligence- XAI)

Yapay zeka modellerinde başarının artmasıyla birlikte kullanılan modellerin karmaşıklığı da artmıştır. Bu modelleri inceleyen bir insanın belirli bir sonuca nasıl ulaşıldığını izlemesi, anlaması ve yorumlaması zordur. Bu kara kutu yaklaşımları; açıklanabilirlik, yorumlanabilirlik, anlaşılabilirlik açısından şeffaf değildir. Bu yüzden etik, güven ve hukuki açıdan kaygılar barındırmaktadır [72]. Bu kaygılar için güven oluşturmaya katkı sağlamak için bazı açıklanabilir yapay zeka ilkeleri bulunmaktadır:

- **Şeffaflık:** Paydaşların modellerin karar verme sürecini anlamalarını sağlamak.
- **Adalet:** Modellerin kararlarının, korunan gruplardaki (ırk, din, cinsiyet, engellilik, etnik köken) kişiler de dahil olmak üzere herkes için adil olmasını sağlamak.
- **Güven:** Yapay zeka sistemini kullanan insan kullanıcıların güven düzeyini değerlendirmek.
- **Sağlamlık:** Giriş verilerindeki veya model parametrelerindeki değişikliklere karşı dirençli olmak, belirsizlikle veya beklenmedik durumlarla karşılaşıldığında bile tutarlı ve güvenilir performansı sürdürmek.
- **Mahremiyet:** Hassas kullanıcı bilgilerinin korunmasını garanti etmek.

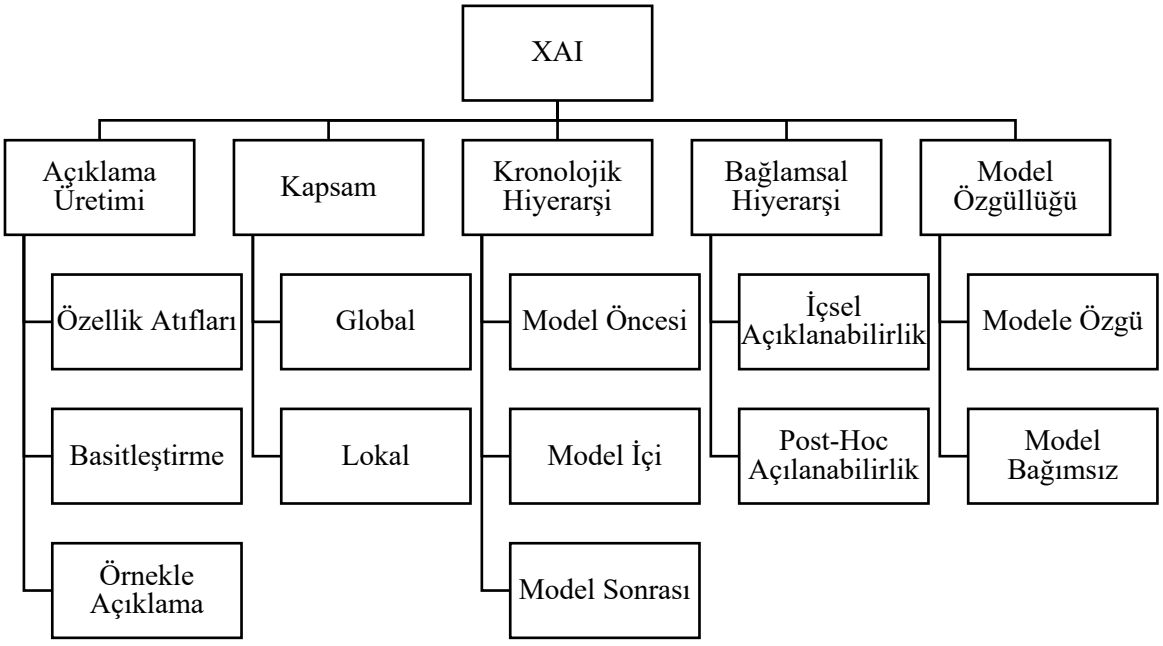
- **Yorumlanabilirlik:** Tahminleri ve sonuçları için insanların anlayabileceği açıklamalar sağlamak.

XAI'daki amaç, kullanıcının ML modellerinin açıklamalar yoluyla elde edilen tahminlerini anlamasını sağlamaktır. Bunun için açıklamalar üreten bir algoritmadan ve ML yöntemlerinden yararlanılır. Açıklama yöntemlerinin ve oluşturulan açıklamaların ML kanalındaki rolü Şekil 2.5' de gösterilmektedir [73]:




**Şekil 2.5:** ML Modellerinin XAI Süreci

XAI çeşitli yaklaşımlara göre aşağıdaki gibi gruplanabilir [74]:




**Şekil 2.6:** XAI Modelleri

- **Açıklama üretimi (Explanation Generation):** Bu yaklaşım, XAI yöntemlerinin önemli bir sınıflandırma boyutunu temsil eder. Belirli bir açıklama formunun seçimi, iletilecek açıklamanın karmaşıklık düzeyine ve son kullanıcının uzmanlık düzeyine bağlıdır. Belirli açıklama formlarına atıfta bulunmaktadır. Özellik Atıfları (Feature Attributions), özelliklerin alaka düzeyinin veya açıklayıcı gücünün orijinal model tarafından oluşturulan tahminlere göre hesaplandığı bir açıklama formunu temsil eder. Basitleştirme (Simplification), orijinal modelin davranışını taklit etmek ve açıklamak için yorumlanabilir daha basit bir model kullanır. Örnekle Açıklama (Explain by Example), bir örneğe karşılık gelen bir tahminin, benzer bir tahminle benzer bir örnek veya benzer bir tahmine sahip farklı bir örnek bularak açıklandığı başka bir açıklama biçimini oluşturur [75].

- **Kapsam (Coverage):** Bu yaklaşım global ve yerel olarak ikiye ayrılır. Global açıklanabilirlik yöntemleri, çok sayıda örnek üzerinden bir ML modeli tarafından öğrenilen kalıpları özetleyen açıklamalar üretir. Bu yöntemler, tahmin çıktı alanının girdi özellikleri açısından dağılımını anlama eğilimindedir. Lokal açıklanabilirlik yöntemleri, belirli bir girdinin belirli bir örnekte belirli bir çıktıya nasıl yol açtığını daha iyi anlamak için modeller arasındaki etkileşimleri inceler. Sonuç olarak, lokal açıklanabilirlik yöntemleri, tek bir örnek veya benzer örneklerden oluşan bir grup için açıklamalar üretir [76].

- **Kronolojik Hiyerarşi (Chronological Hierarchy):** XAI yöntemlerini, modelleme adımına göre bir XAI yönteminin uygulandığı zaman noktasına göre gruplandırır. Birincisi, Model Öncesi Açıklanabilirlik, modelleme adımına bakılmaksızın XAI yönteminin orijinal veri kümesine uygulanmasıdır. Bu kategoriye giren yöntemler, girdi verilerinin keşif ve sunum perspektiflerini benimseme eğilimindedir. Model İçi Açıklanabilirlik, model eğitim sürecinin bir parçası olarak açıklamalar üretmek anlamına gelir. Model Sonrası Açıklanabilirlik, modeli geçmiş veriler üzerinde eğittikten sonra momentum sonrası adım olarak bir ML modelinin tahminleri için açıklamalar üretmek anlamına gelir.

- **Bağlamsal Hiyerarşi (Contextual Hierarchy):** Bir ML modelinin tahminlerin açıklamalarını tek başına sağlama yeteneği veya ayrı bir XAI yönteminin uygulanmasıyla ilgilidir. Bu bağlamda, içsel açıklanabilirlik, doğası gereği

yorumlanabilir modellere sahip olmak anlamına gelir; yani simüle edilebilir, algoritmik olarak şeffaf ve ayrıştırılabilir olma açısından yüksek derecede şeffaflık gösteren modellerdir. Doğrusal modeller ve basit karar ağaçları, doğası gereği açıklanabilirliğe sahip makine öğrenimi modellerinin ortak örneklerini sağlar. Post-hoc açıklanabilirlik, tasarım açısından yorumlanamayan karmaşık modelleri hedefler. Bu kategorideki XAI yöntemleri, analiz edilen makine öğrenimi modelinin akıl yürütme sürecine tersine mühendislik yapmak için eğitimli bir modele uygulanır.

- **Model Özgüllüğü (Model Specificity):** Kullanılan ML modelinin XAI yöntemi seçimi üzerindeki etkisidir. Modele özgü açıklama yöntemleri belirli modellerle sınırlıdır. Çünkü XAI yöntemleri belirli modelin iç unsurlarına göre uyarlanmıştır [73]. Modelden bağımsız yöntemler, belirli bir kara kutu tipine, açıklamaya veya veri tipine bağlanan tahmin edicileri içerir. LIME ve SHAP ikinci alt kategorinin örnekleridir [76].

### 2.6.1. LIME

Makine öğrenimi modellerini kara kutu işlevleri olarak ele alan modelden bağımsız açıklama yöntemleri genellikle yalnızca modelin çıktısına erişebilmektedir. LIME (Local Interpretable Model Agnostic Explanations- Yerel Yorumlanabilir Model Agnostik Açıklamalar) yaklaşımının ana fikri, modelden elde edilen bu karmaşık çıktıyı, tahminlerinin açıklaması kolay yerel bir vekil modele uydurarak açıklamaktır [77].

LIME kullanarak açıklamak istenen bir modelin denklemi aşağıdaki gibidir:

$$\xi(x) = arg\min_{g \in G} \mathcal{L}(f, g, \pi_x) + \Omega(g) \tag{2.23}$$

$\boldsymbol{f}$: orijinal tahminci, $\boldsymbol{x}$: orijinal özellikler, $\boldsymbol{g}$: Model, $\boldsymbol{\pi_x}$: $\boldsymbol{z}$ ve $\boldsymbol{x}$ arasındaki yakınlık ölçüsü.

LIME yöntemi aşağıdaki adımları takip eder:

1. Yeni örneklerin üretilmesi ve orijinal modeli kullanarak tahminlerinin alınması.
2. Yeni örneklerin, açıklanan örneğe yakınlığa göre ağırlıklarının belirlenmesi.

Açıklanması istenen girdinin bir kısmını kapsayan belirli bir örnek koleksiyonundan elde edilen çıktı olasılıklarını kullanarak doğrusal bir model oluşturulur. Daha sonra vekil model ağırlıkları,

girdi özelliklerinin değerini ölçmek için kullanılır. Ayrıca LIME modelden bağımsızdır, dolayısıyla makine öğreniminin herhangi bir modeline uygulanabilir [77].

### 2.6.2. SHAP

SHAP (SHapley Additive exPlanations), her bir özelliğin bu tahmine katkısını hesaplayarak, verilen girdi için model tahminini açıklamayı önerir. SHAP, bu hedefe ulaşmak için başlangıçta oyun teorisinden gelen Shapley Değerlerini kullanır. Shapley değeri (SHAP) kavramı başlangıçta işbirlikçi bir takımdaki bireysel bir oyuncunun önemini tahmin etmek için geliştirilmiştir. Bu konsept, oyunun nihai sonucuna katkılarının göreceli önemine bağlı olarak toplam kazancı veya getiriyi oyuncular arasında dağıtmayı amaçlamaktadır. SHAP değerleri, her oyuncuya adil veya makul bir ödül verilmesine bir çözüm sağlar ve aşağıdaki doğal özellikler veya aksiyomlarla karakterize edilen benzersiz bir sonucu temsil etmektedir: yerel doğruluk (toplanabilirlik), tutarlılık (simetri) ve var olmama (sıfır etkisi) [78] [79].

Bir i özelliğinin önemini SHAP değeri olarak açıklayan denklem aşağıda verilmiştir [80]:

$$\phi_i = \frac{1}{|N|!} \sum_{S \subseteq N \setminus \{i\}} |S|! \, (|N| - |S| - 1)! \, [f(S \cup \{i\}) - f(S)] \tag{2.24}$$

Burada $f(S)$ modelin çıktısını, $N$ ise tüm özelliklerin setini göstermektedir. $\phi_i$ ($i$. özelliğe ait SHAP değeri), bir özellik setinin tüm olası permütasyonlarındaki katkılarını gösterir. Buna göre, özellikler sete ayrı ayrı eklenmekte ve model çıktısındaki değişiklik $o$ değişkenin ilgi düzeyini göstermektedir.

## 2.7. Veri Doğrulama (Data Validation)

Makine öğrenmesi modellerinin performansının değerlendirilmesinin temel ilkesi, eğitim için kullanılan örneklerin aynı zamanda değerlendirme için de kullanılmaması gerektiğidir. Bu gereklilik sınıflandırmanın performans metriklerini doğrudan etkileyebilir. Kullanılan veri setindeki örnek sayısı küçükse, eğitim örneklerinin ayrı eğitim ve ayarlama örneklerine sabit bir şekilde bölünmesi, her örnek bir kez ve yalnızca tek bir amaç için (örneğin, eğitim) kullanıldığından, küçük örnek boyutunun sınırlamalarını artıracaktır. Çapraz Doğrulama (Cross Validation- CV), nihaî değerlendirme için eğitim veri seti hazırlanır, fakat artık doğrulama setine ihtiyaç yoktur. $k$-katmanlı CV ($k$-fold cross validation) olarak adlandırılan temel yaklaşımda, eğitim seti $k$ adet küçük bölüme ayrılır. Her $k$ için aşağıdaki prosedür izlenir:

- Seçilen bir model, eğitim veri kullanılarak katmanların $k-1$'i kullanılarak eğitilir.
- Ortaya çıkan model, verilerin geri kalanıyla doğrulanır.

$k$- katmanlı çapraz doğrulamayla bildirilen performans ölçüsü, döngüde hesaplanan değerlerin ortalamasıdır. Bu yaklaşım hesaplama açısından pahalı olabilir, ancak çok fazla veri israfına neden olmaz (rastgele bir doğrulama kümesini sabitlerken olduğu gibi), bu da örnek sayısının çok küçük olduğu ters çıkarım gibi problemlerde büyük bir avantajdır [81]. Çapraz doğrulama yaklaşımlarından biri de Katmanlı $k$-Kat Çapraz Doğrulayıcı (Stratified $k$-Fold)' dır. $k$-katman CV' nin gelişmiş versiyonudur. Veri setini katmanlara ayırırken, her katmanda her sınıfa ait örneklerin yüzdesel olarak korunmasını amaçlar. Özellik dengesiz sınıflı veri setlerinde kullanılması önerilir. Her katmanı, orijinal veri setinin sınıf dağılımına yakın bir dağılıma sahip olacak şekilde oluşturmayı amaçlar.

## 2.8. Değerlendirme Metrikleri (Evaluation Metrics)

Değerlendirme metrikleri, bir istatistiksel veya makine öğrenimi modelinin performansını ve etkinliğini değerlendirmek için kullanılan niceliksel ölçümlerdir. Bu ölçümler, modelin ne kadar iyi performans gösterdiğine ilişkin bilgiler sağlar ve farklı modelleri veya algoritmaları karşılaştırmaya yardımcı olur. Karışıklık matrisi (Confusion Matrix) bir $N\ x\ N$ matrisidir; burada $N$, tahmin edilen sınıfların sayısıdır. Problem için $N = 2$'yi varsayarsak, $2\ x\ 2$ 'lik bir karışıklık matrisi elde edilir. Çıktının iki veya daha fazla sınıf olabildiği makine öğrenmesi sınıflandırma problemlerine yönelik bir performans ölçümüdür. Karışıklık matrisi, öngörülen ve gerçek değerlerin 4 farklı kombinasyonunu içeren bir tablodur. Duyarlılık, Özgüllük, Doğruluk ve AUC-ROC eğrilerini ölçmek için son derece kullanışlıdır [82].

**Tablo 2.3:** Karışıklık Matrisi

| **Doğru etiket** | **Tahminlenen etiket** | |
|---|---|---|
| | Pozitif | Negatif |
| Pozitif | Doğru pozitif (TP) | Yanlış negatif (FN) |
| Negatif | Yanlış pozitif (FP) | Doğru negatif (TN) |

- **Doğruluk (Accuracy- ACC):** Doğru olan toplam örneğin, doğru tahmin edilen örneklerin sayısına oranıdır. ACC, modellerin genel tahmin etkinliğini ölçer; ancak

gerçek dünyadaki kredi veri kümelerinde sıklıkla görülen, veri kümesinin dengeli olmaması durumunda yanıltıcı sonuçlar vermesi nedeniyle güvenilir bir parametre değildir.

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.25}$$

- **Kesinlik (Precision- PRE):** Pozitif olarak tahmin edilen bir durumdaki başarıyı gösteren parametredir.

$$PRE = \frac{TP}{TP + FP} \tag{2.26}$$

- **Duyarlılık (Recall- REC):** Pozitif durumların ne kadar başarılı tahmin edildiğini gösteren parametredir.

$$REC = \frac{TP}{TP + FN} \tag{2.27}$$

- **F-Beta Skoru (FBS):** F-beta puanı, kesinlik ve duyarlılığın ağırlıklı harmonik ortalamasıdır ve optimal değerine 1'de ve en kötü değerine 0'da ulaşır. Çok sınıflı ve çok etiketli durumda bu, ortalama parametreye bağlı ağırlıklandırmayla birlikte her sınıfın FBS puanının ortalamasıdır.

$$F - Beta = (1 + \beta^2) * \frac{PRE * REC}{(\beta^2 * PRE) + REC} \tag{2.28}$$

- **F1 Skoru (F1 Score):** Bir sınıflandırma problemi için kesinlik ve duyarlılık değerlerinin harmonik ortalamasıdır. F1 skorunun 1 olması hem kesinlik hem de duyarlılığın iyi olduğu gösterir. Modelin tüm pozitif vakaları doğru bir şekilde tanımladığı ve negatif bir durumu pozitif durum olarak işaretlemediği anlamına gelir.

$$F1 = 2 * \frac{PRE * REC}{PRE + REC} \tag{2.29}$$

- **F2 Skoru (F2 Score):** Kesinlik ve duyarlılığın ağırlıklı harmonik ortalamasıdır. Kesinlik ve duyarlılığa eşit ağırlık veren F1 skorunun aksine, F2 skoru duyarlılığa kesinlikten daha fazla ağırlık verir. Yanlış negatiflerin, yanlış pozitiflerden daha kötü olduğu düşünülen durumlarda duyarlılığa daha fazla ağırlık verilmelidir. Hassasiyete 2

kat önem veren bir ölçümdür [83]. Çok sınıflı ve çok etiketli durumda bu, ortalama parametreye bağlı olarak ağırlıklandırılan her sınıfın F2 puanının ortalamasıdır.

$$F2 = 5\left(\frac{(PRE)(REC)}{((4)(PRE)) + REC}\right) \tag{2.30}$$

- **Matthews Korelasyon Katsayısı (Matthews Correlation Coefficient- MCC):** Çok sınıflı ve çok etiketli durumda bu, ortalama parametreye bağlı ağırlıklandırmayla birlikte her sınıfın MCC puanının ortalamasıdır. MCC puanı $[-1, +1]$ aralığındadır. 1 en iyi değerdir [84] [85].

$$MCC = \frac{(TP * TN) - (FP * FN)}{\sqrt{(TP + FP) * (TP + FN) * (TN + FP) * (TN + FN)}} \tag{2.31}$$

Çok sınıflı durumlarda, Matthews korelasyon katsayısı, K sınıfları için bir karışıklık matrisi C cinsinden tanımlanabilir:

$$MCC = \frac{c * s - \sum_k^K p_k t_k}{\sqrt{(s^2 - \sum_k^K p_k^2) * (s^2 - \sum_k^K t_k^2)}} \tag{2.32}$$

Bu formüldeki ara değişkenler aşağıda detaylandırılmıştır:

$t_k = \sum_i^K C_{ik}$ : k sınıfının gerçekten meydana gelme sayısı,

$p_k = \sum_i^K C_{ki}$: k sınıfının tahmin edilme sayısı,

$c = \sum_k^K C_{kk}$: doğru tahmin edilen toplam örnek sayısı,

$s = \sum_i^K \sum_j^K C_{ij}$: toplam örnek sayısıdır.

- **Jaccard Benzerlik Endeksi (Jaccard Similarity Index- JSI):** Jaccard benzerlik katsayısı veya Jaccard indeksi olarak da bilinen Jaccard benzerlik indeksi, ikili ve çok sınıflı sınıflandırma problemlerinde yaygın olarak kullanılan bir değerlendirme ölçütüdür. Gerçek pozitif (TP) tahminlerin sayısının, doğru pozitif (TP) ve yanlış pozitif (FP) tahminlerin sayısına oranı olarak tanımlanır.

  İkili sınıflandırma probleminde hem doğru etiketin hem de öngörülen etiketin pozitif olması durumunda bir tahminin doğru pozitif olduğu kabul edilir; doğru etiket negatif

ve tahmin edilen etiketin pozitif olması durumunda ise yanlış pozitif kabul edilir. Yanlış negatif (FN) tahminler, gerçek etiketin pozitif ve tahmin edilen etiketin negatif olduğu tahminlerdir. Gerçek negatif (TN) tahminler hem gerçek etiketin hem de tahmin edilen etiketin negatif olduğu tahminlerdir.

Çok sınıflı bir sınıflandırma probleminde, Jaccard benzerlik indeksi her sınıf için ayrı ayrı hesaplanabilir ve daha sonra genel Jaccard benzerlik indeksini elde etmek için tüm sınıfların ortalaması alınabilir. Jaccard benzerlik endekslerinin ağırlıklı ortalaması da her sınıftaki örnek sayısına göre verilen ağırlıklarla hesaplanabilir.

Jaccard benzerlik indeksini hesaplamak için formül şu şekildedir:

$$JSI = \frac{TP}{TP + FP} \tag{2.33}$$

Jaccard benzerlik indeksi 0 ila 1 arasında değişir; 1 değeri, tahmin edilen etiketler ile gerçek etiketler arasında mükemmel en iyi uyumu belirtir. Jaccard benzerlik indeksi, yanlış pozitif veya yanlış negatif tahminleri değil, yalnızca doğru pozitif ve doğru negatif tahminleri hesaba kattığı için sınıf dağılımının dengesiz olduğu durumlarda yararlı bir değerlendirme ölçütüdür. Jaccard benzerlik indeksinin her sınıftaki örnek sayısına duyarlı olduğunu unutmamak önemlidir; bu nedenle, tam bir tablo elde etmek için bu ölçümün hassasiyet, hatırlama ve F1 puanı gibi diğer değerlendirme ölçümleriyle birlikte kullanılması önerilir [86].

- **Alıcı Çalışma Karakteristiği- Eğrinin Altındaki Alan (Receiver Operating Characteristic- Area Under the Curve ROC-AUC):** İkili sınıflandırma modelinin performansını değerlendirmek için kullanılan bir ölçümdür. ROC eğrisi, çeşitli eşik ayarlarında gerçek pozitif oranın (TPR) yanlış pozitif orana (FPR) karşı grafiğidir. TPR, gerçek pozitif sayısının toplam pozitif sayısına oranıdır; FPR ise yanlış pozitif sayısının toplam negatif sayısına oranıdır.

  AUC, ROC eğrisinin altındaki alandır. 0 ile 1 arasında değişir; 0,5 değeri, rastgele tahminden daha iyi performans göstermeyen bir modeli temsil eder ve 1 değeri, mükemmel tahminler yapan bir modeli temsil etmektedir. Daha yüksek bir AUC değeri, modelin pozitif ve negatif sınıfları ayırt etmede daha iyi olduğunu göstermektedir.

ROC eğrisinin ve AUC değerinin yorumlanması spesifik soruna ve alana bağlıdır. Genel olarak AUC değeri 0,7 ile 0,8 arasında olan bir model kabul edilebilir olarak kabul edilirken, 0,8'den büyük bir değer iyi olarak kabul edilir. Ancak yorum, spesifik kullanım durumuna ve yanlış pozitif ve yanlış negatiflerin maliyetine bağlı olarak değişiklik gösterebilir.

Çok sınıflı ve çok etiketli durumda, ortalama parametreye bağlı olarak ağırlıklandırılan her sınıfın doğruluk skorunun puanının ortalamasıdır. Çok sınıflı bir sınıflandırma probleminde, ROC-AUC hala bir metrik olarak kullanılabilir ancak çoklu sınıfları hesaba katmak için bazı değişiklikler gerektirir. Diğer bir yaklaşım ise, her sınıf çifti için bir ikili sınıflandırıcıyı eğittiğimiz, bir sınıfı pozitif sınıf ve diğerini negatif sınıf olarak ele aldığımız bire karşı bir (OvO) stratejisini kullanmaktır. Her sınıf çifti için ROC eğrisini ve AUC değerini hesaplıyoruz ve ardından tek bir ölçüm elde etmek için tüm çiftlerdeki AUC değerlerinin ortalamasını alıyoruz.

Her iki durumda da sınıfların dengeli olmasını sağlamak, yani her sınıftaki örnek sayısının kabaca eşit olmasını sağlamak veya sınıf dengesizliğini gidermek için uygun örnekleme tekniklerini kullanmak önemlidir.

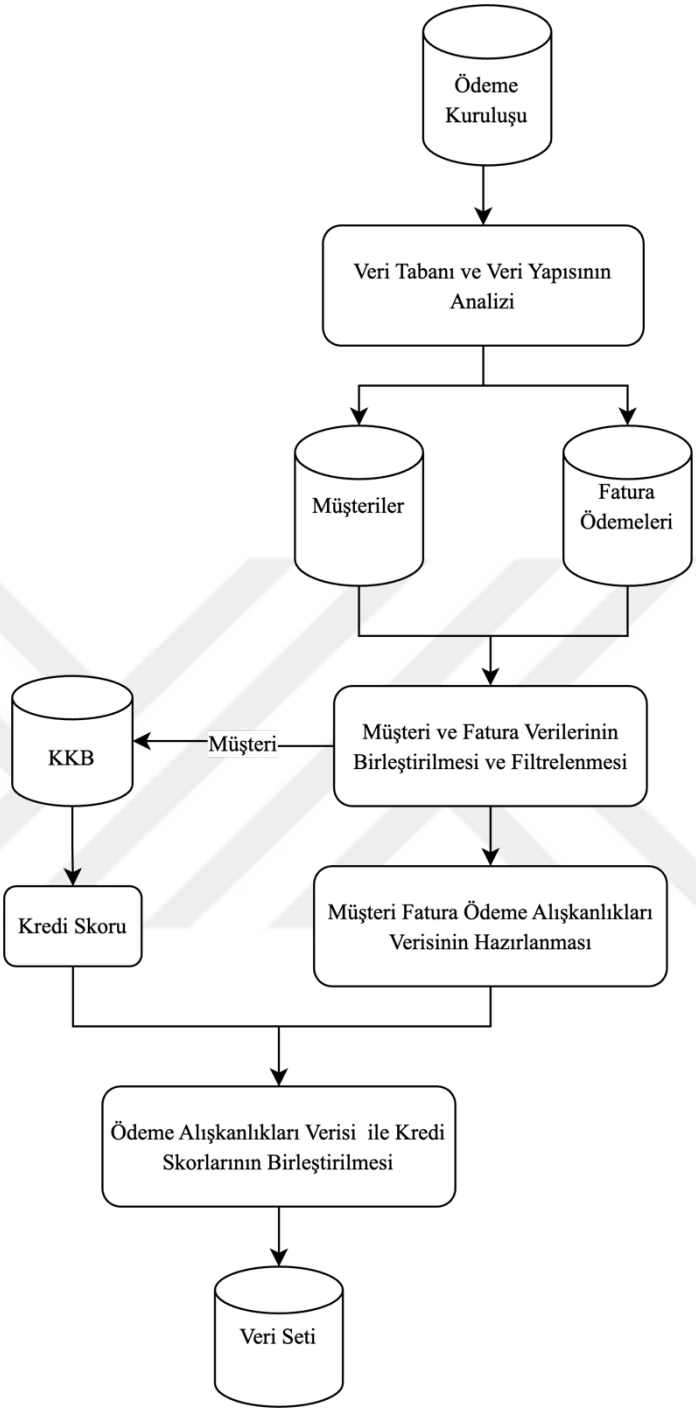
ROC-AUC' nin, özellikle sınıflar oldukça dengesiz olduğunda veya yanlış pozitiflerin ve yanlış negatiflerin maliyeti sınıflar arasında değiştiğinde, çok sınıflı problemler için her zaman en iyi ölçüm olamayabilmektedir. Bu gibi durumlarda kesinlik, duyarlılık, F1 puanı veya bu metriklerin ağırlıklı ortalaması gibi diğer metrikler daha uygun olabilir [86].

# 3. MALZEME VE YÖNTEM

## 3.1. Veri

Çalışmamızda, Türkiye'de bulunan bir fatura ödeme merkezine ait gerçek verilerden oluşan bir veri seti kullanılmıştır. Veri seti Ocak 2021 ve Ocak 2022 fatura dönemini kapsamaktadır. Kişisel ve hassas herhangi bir içerik bulunmamaktadır.

Ödeme merkezi veri tabanındaki büyük veri, çeşitli aşamalardan geçirilerek bu çalışmada kullanılabilir duruma getirilmiştir. Bu süreçteki aşamalar, Şekil 3.1'de özetlenmiştir.




**Şekil 3.1:** Veri Edinme Süreci

Öncelikli olarak ödeme kuruluşuna ait veri tabanına erişim izinleri ve erişim yöntemleri belirlenmiş ve testler yapılmıştır.

Veri tabanına erişim sonrası, veri tabanı ve veri yapıları incelenmiştir. İlişkisel bir mimariye sahip veri tabanında yer alan tüm şema ve tablolar gözden geçirilmiştir.

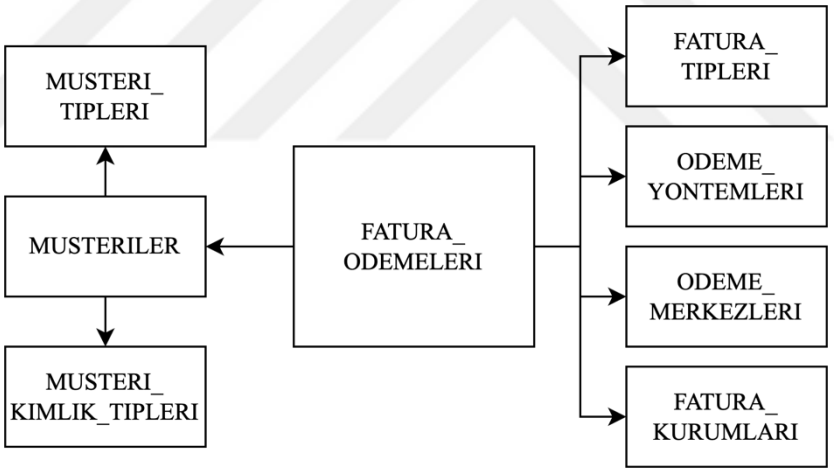
Çalışmaya konu olacak verilere sahip tablo ve şemalar belirlenerek, bu yapılar üzerinde çalışmaya başlanmıştır. Müşteriler ve Fatura Ödemeleri olarak iki farklı şemada yer alan veriler için gerekli ilişkisel bağlantılar kullanılarak, tek bir tabloda yere alacak şekilde düzenlenmiştir.

Fatura ödemesi yapan müşteriler içerisinde, gerçek kredi skoru olan kişilerin bulunması için farklı bir veri tabanı kullanılmıştır. Gerekli entegrasyon yapılarak ilgili kişiler ve bu kişilerin kredi skorlarının bulunması sağlanmıştır.

Elde edilen fatura ödeme hareketleri, müşteri ve kredi skoru verileri kullanılarak, sonraki adımlarda detaylı olarak anlatılacak "Fatura Ödeme Alışkanlıkları" veri kümesi oluşturulmuştur.

### 3.1.1. Veri Tabanı Yapısı

Veri kaynağı SQL Server tabanlı bir sistemden oluşmaktadır. Burada yer alan veri tabanı tabloları Şekil 3.2' de gösterilmiştir.




**Şekil 3.2:** Ödeme Kuruluşu Veri Tabanı Tablo Yapısı

Veri kaynağında yer alan tablolara ait detaylar ise Tablo 3.1' de verilmiştir.

**Tablo 3.1:** Tablo Detayları

| Tablo Adı | Kolon Sayısı | Kayıt Sayısı |
|---|---|---|
| MUSTERILER | 34 | 525.000 |
| MUSTERI_TIPLERI | 11 | 2 |

| | | |
|---|---|---|
| MUSTERI_KIMLIK_TIPLERI | 11 | 2 |
| FATURA_KURUMLARI | 11 | 6.430 |
| ODEME_MERKEZLERI | 55 | 25.855 |
| ODEME_YONTEMLERI | 11 | 2 |
| FATURA_TIPLERI | 12 | 47 |
| FATURA_ODEMELERI | 60 | 60.000.000 |

- **FATURA_ODEMELERI:** Tüm fatura ödeme hareketlerinin yer aldığı tablodur. Fatura tipi, son ödeme tarihi, ödeme tutarı gibi bilgilerin tamamımı burada kayıtlıdır. Fatura ödemesi yapıldığında bu tabloya ilgili fatura ve ödeme bilgilerine ilişkin bilgiler bu tabloya eklenir. Ödeme, kurum müşteri veya kurum müşterisi olmayan kişiler tarafından yapılabilir. Eğer ödemeyi yapan kişi kurum müşterisi ise müşteri numarası eklenerek tabloya ekleme yapılır. Müşteri değilse, müşteri numarası boş bırakılır. Bu çalışmada kurum müşterileri dikkate alınmıştır.

- **FATURA_TIPLERI:** Kurumda ödeme yapılabilen fatura türlerinin yer aldığı tablodur. Bu türler Tablo 3.2’ de yer almaktadır.

**Tablo 3.2:** Fatura Tipleri

| **Fatura Kodu** | **Fatura Adı** |
|---|---|
| 1 | Elektrik |
| 2 | Su |
| 3 | Doğalgaz |
| 4 | GSM |
| 5 | İnternet |
| 6 | Telefon |
| 7 | Uydu/TV |

- **ODEME_YONTEMLERI:** Faturaların ödeme yöntemleri bu tabloda bulunmaktadır. Kurumda nakit veya kredi kartı ile ödeme yapılmaktadır.

- **ODEME_MERKEZLERI:** Fatura ödeme hizmeti sağlayan ödeme kuruluşuna bağlı bayilerin bulunduğu tablodur. Bu bayiler Türkiye'nin farklı il ve ilçelerinde yer almaktadır.

- **FATURA_KURUMLARI:** Her fatura tipine özelinde bu hizmeti sunan farklı şirketler bulunmaktadır. Örneğin, İstanbul şehrinde elektrik hizmeti sağlayan bir veya birden fazla şirket olabilir. Fatura tiplerine göre hizmet veren şirketlerin listesi bu tabloda bulunmaktadır.

- **MUSTERILER:** Kuruma kayıtlı müşteri bilgilerinin tutulduğu tablodur. Müşteriye ait detaylı bilgiler burada yer almaktadır. Ödeme kuruluşunda fatura ödeyen kişilerin, kurum müşterisi olma zorunluluğu yoktur. Bu tez çalışması kapsamında, kurum müşterisi olan ve dolayısıyla bu tabloda yer alan kişiler üzerinde işlemler yapılmıştır.

- **MUSTERI_TIPLERI:** Müşteri türleri bu tabloda bulunmaktadır. Ödeme merkezinde kurumsal ve bireysel olmak üzere 2 tipte müşteri tipi vardır. Çalışma kapsamında bireysel müşteriler yer almaktadır.

- **MUSTERI_KIMLIK_TIPLERI:** Müşterinin kayıt olurken beyan edebileceği kimlik türleri bu tablodadır. Kimlik tipi olarak; pasaport ve nüfus cüzdanı yer almaktadır.

### 3.1.2. Veri Setinin Hazırlanması

Bu bölümde bir önceki adımda detayları verilen tablolar incelenmiştir. Tez çalışması için uygun olmayan kısımlar belirlenerek, bu kısımların ayrıştırılması amaçlanmıştır. Veri setinin kaynağı kurumsal bir şirket olduğundan tablolarda birçok detay bilgi yer almaktaydı. Bu detaylar incelendiğinde aşağıdaki sonuçlara ulaşılmış ve bu sonuçlara uygun düzenlemeler yapılmıştır. Tez çalışmasında kullanılmayacak kısımların belirlenmesi aşağıdaki adımlar gözden geçirilerek yapılmıştır.

- İlişkisel veri tabanı mimarisine sahip olduğundan tüm tablolarda birincil ve ikincil anahtarlar yer almaktadır.

- Kurum içinde yapılan işlemlerin takibi için kullanılan tekil kolonlar bulunmaktadır. Bu kolonlar yalnızca kurum içinde bir anlam ifade etmektedir.

- Her kayıt için, bu kaydı oluşturan veya güncelleyen kullanıcıların bilgilerini ve bu işlemlerin gerçekleştirildiği tarih ve saatleri tutan alanlar mevcuttur.

- Hiç kullanılmamış kolonlarla karşılaşılmıştır. Bu kolonlar, herhangi bir sebeple oluşturulmuş, hiç kullanılmış veya bir süre sonra kullanımı bırakılmış alanlardır.

- Veri alışverişi sırasında kullanılan isteklerin ve dönüşlerin tutulduğu alanlar yer almaktadır.

Kullanılmayacak kısımlar belirlendikten sonra nihai veri seti için gerekli çalışmalara başlanmıştır ve var olan tablolar aşağıdaki kriterlere göre filtrelenmiştir:

- **Ödeme yapan kişinin kimliğinin bilinmesi ve kurum müşterisi olması:** Herhangi bir bayiden fatura ödemesi yapan kişinin kimliğini beyan etmesi veya ödeme kuruluşu müşterisi olma zorunluluğu yoktur. Kurum müşterisi değilse, veri tabanına anonim olarak kayıt atılır. Çalışmamız kapsamında kredi risk değerlendirmesi yapabilmek için fatura sahibinin kimliğini bilmek zorunludur. Bu nedenle veri setinde kimlik bilgisi bulunan müşteriler filtrelenmiştir.

- **Ödemesi yapılan faturanın sahibi ile müşterinin aynı olması:** Çeşitli nedenlerle herhangi bir faturayı ödeyen ile fatura sahibi farklılık gösterebilmektedir. Örneğin; kiracı olarak bulunulan bir eve ait elektrik faturası, kiracının değil hala ev sahibi üzerine kayıtlı olabilmektedir. Kiracının ödemeyi zamanında veya geç yapması ev sahibini değil kiracıyı bağlamalıdır. Bu nedenle çalışmamızda, müşterinin kendi adına kayıtlı faturalara yaptığı ödemeler baz alınmıştır.

- **Müşterinin gerçek kredi skorunun bilinmesi:** Kurumda fatura ödemesi yapan mevcut müşteriler arasından kredi risk puanı bilinen müşteriler seçilmiştir.

- **Müşterinin fatura ödeme hareketine sahip olması:** Ödeme alışkanlıklarının analizi için son 1 yılda içinde herhangi bir fatura türünden ödeme yapmış olması gerekmektedir.

- **Bireysel müşteri olması:** Ödeme kuruluşunda bireysel ve kurumsal ödemeler yapılmaktadır. Tez çalışması kapsamında bireysel müşterilerin kredi skorları

hesaplanacaktır. Bu yüzden yalnızca bireysel ödemeler baz alınmış olup, kurumsal ödemeler veri setinden çıkarılmıştır.

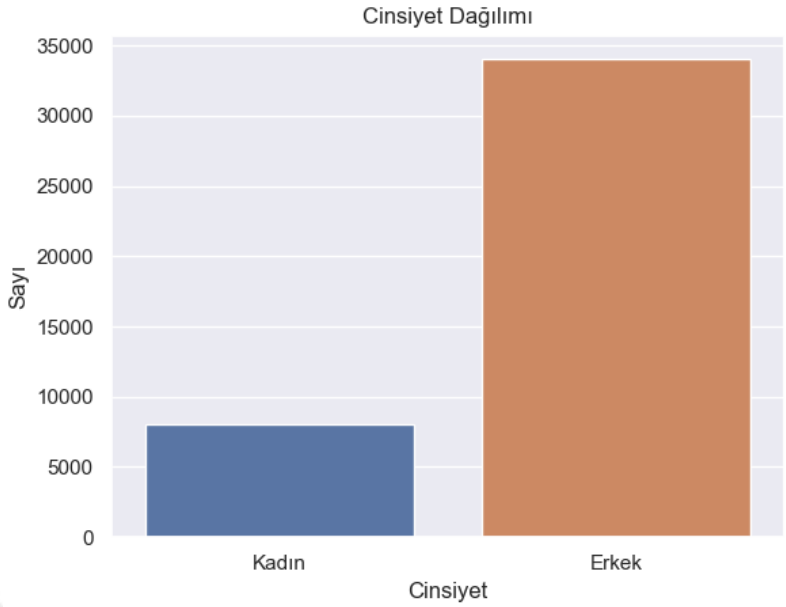
### 3.1.3. Fatura Ödeme Alışkanlıkları Veri Setinin Oluşturulması

Veriler uygun şekilde filtrelendikten sonra müşterilerin fatura ödeme alışkanlıklarının belirlenmesi aşamasına geçilmiştir. Bu aşamada Tablo 3.3' te gösterilen özellikler çıkarılmıştır. Tablo yer alan çoğu özellik mevcut özelliklerin işlenmesiyle elde edilmiştir.

**Tablo 3.3:** Nihai Veri Setinin Özellikleri

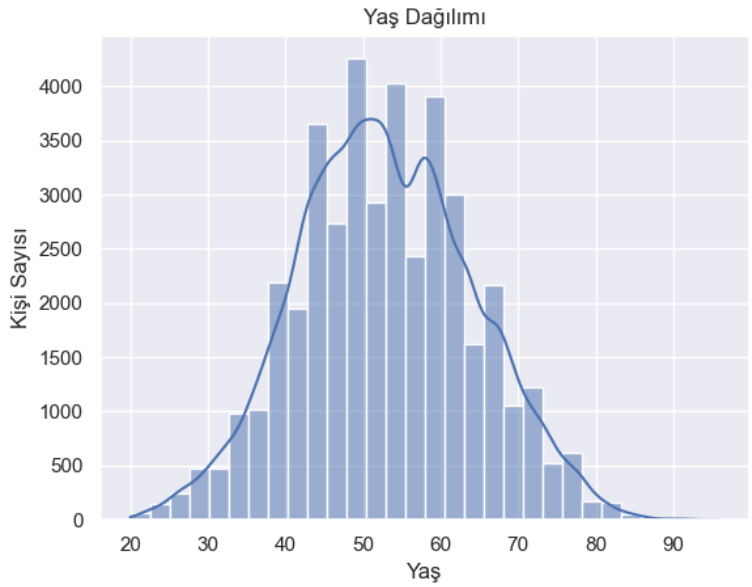
| **Sıra** | **Özellik Adı** | **Tipi** | **Yeni Özellik** |
|---|---|---|---|
| 1 | CINSIYET | Kategorik | Hayır |
| 2 | YAS | Sayısal | Evet |
| 3 | URUN_KODU | Kategorik | Hayır |
| 4 | FATURA_TIPI | Kategorik | Hayır |
| 5 | NAKIT_YUZDESI | Sayısal | Evet |
| 6 | ODENEN_FATURA_SAYISI | Sayısal | Evet |
| 7 | TOPLAM_ODEME_TUTARI | Sayısal | Evet |
| 8 | ODEME_TUTARI_ORT | Sayısal | Evet |
| 9 | GEC_ODEME_ORANI | Sayısal | Evet |
| 10 | GEC_ODENEN_ORT_GUN | Sayısal | Evet |
| 11 | ERKEN_ODENEN_ORT_GUN | Sayısal | Evet |
| 12 | FATURA_TIPI_SAYISI | Sayısal | Evet |
| **13** | **KREDI_SKORU** | **Kategorik** | **Hayır** |

- **CINSIYET:** Müşterinin cinsiyetini belirten alandır. Müşteri tablosundan alınmıştır. Erkek: 1 ve Kadın: 2 olacak şekilde kategorik bir özelliktir. Veri setindeki cinsiyet dağılımı Şekil 3.3'te gösterilmiştir.




**Şekil 3.3:** Cinsiyet Dağılımı

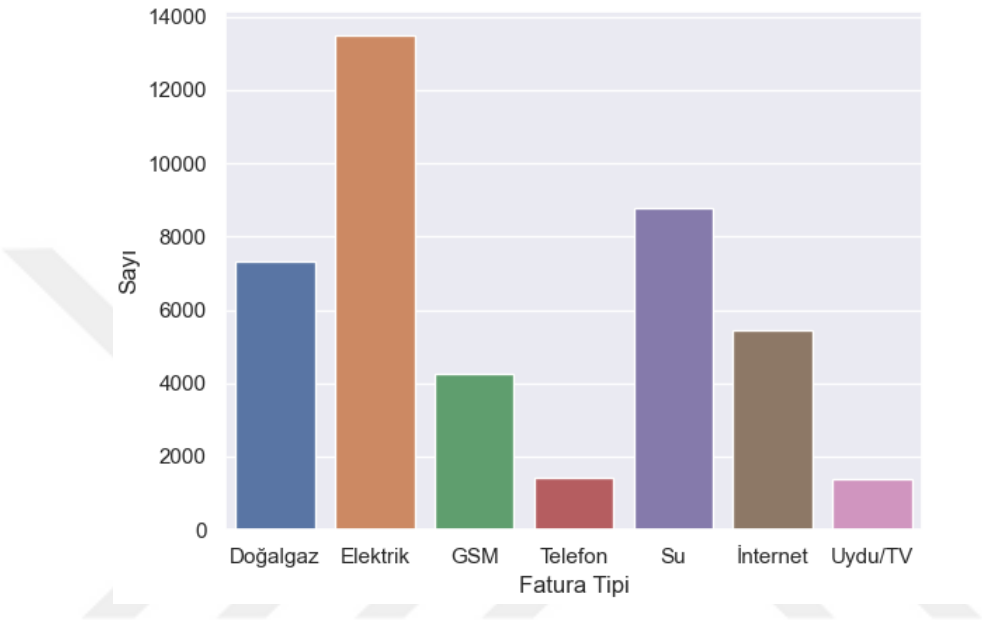
- **YAS:** Ocak 2022 itibariyle, müşteri tablosundan alınan Doğum Tarihi ile hesaplanmış olan müşteri yaşıdır. Veri setindeki yaş dağılımı Şekil 3.4'te gösterilmiştir.




**Şekil 3.4:** Yaş Dağılımı

- **URUN_KODU:** Her faturanın sahip olduğu bir ürün kodu bulunmaktadır.

- **FATURA_TIPI:** Ödemesi yapılmış fatura tipini belirtir (Elektrik: 1, Su: 2, Doğalgaz:3, İnternet:4, GSM: 5 Telefon:6, Uydu/TV: 7). FATURA_ODEMELERI ve FATURA_TIPLERI tabloları kullanılarak elde edilmiştir. Veri setindeki fatura tipleri dağılımı Şekil 3.5'te gösterilmiştir.

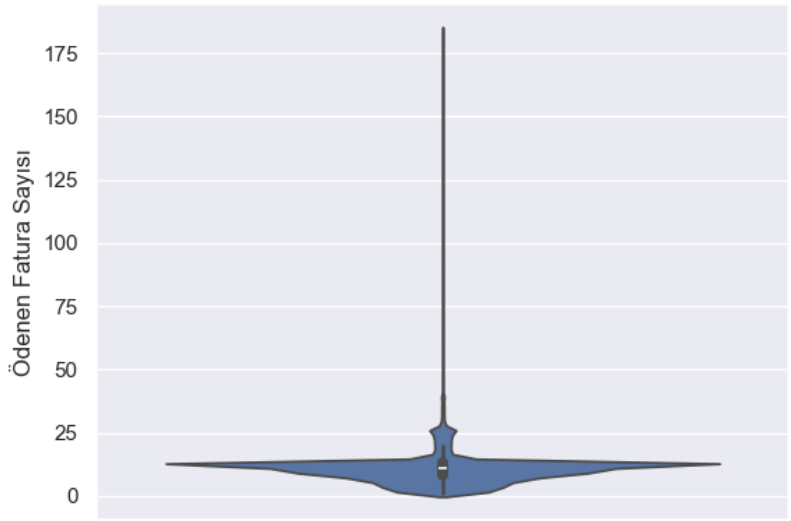



**Şekil 3.5:** Fatura Tipi Dağılımı

- **NAKIT_YUZDESI:** Faturanın ödenme biçimini gösterir. Faturalar, nakit veya kredi kartı ile ödenebilir. Bir müşterinin, ilgili fatura tipinde yapmış olduğu toplam ödeme sayısının, nakit ödeme sayısına oranı ile elde edilmiştir. FATURA_ODEMELERI ve ODEME_YONTEMLERI tabloları birleştirilerek elde edilmiştir.

$$NAKIT_YUZDESI = \frac{NAKIT_ODEME_SAYISI}{ODEME_SAYISI} \tag{3.1}$$

- **ODENEN_FATURA_SAYISI:** Müşterinin son 1 yılda ilgili fatura tipinde ödediği toplam fatura sayısıdır. FATURA_ODEMELERI ve FATURA_TIPLERI tabloları kullanılmıştır.




**Şekil 3.6:** Ödenen Fatura Sayısı Dağılımı

- **TOPLAM_ODEME_TUTARI:** Müşterinin her fatura tipinde ödediği toplam fatura tutarını ifade etmektedir.

- **ODEME_TUTARI_ORT:** Bir müşterinin son 1 yılda ilgili fatura tipinde ödediği toplam fatura tutarının, fatura sayısına bölünmesiyle hesaplanmıştır.

$$ODEME_TUTARI_ORT = \frac{TOPLAM_ODEME_TUTARI}{ODEME_SAYISI} \tag{3.2}$$

- **GEC_ODEME_ORANI:** Her fatura bir son ödeme tarihine sahiptir. Müşterinin ilgili fatura tipindeki tüm ödemelerinin sayısının, son ödeme tarihinden sonra yapılmış ödemeler sayısına bölünmesiyle elde edilmiştir. Bir fatura ya geç ya da erken ödenmiş olabilir.

$$GEC_ODEME_ORANI = \frac{GEC_ODEME_SAYISI}{ODEME_SAYISI} \tag{3.3}$$

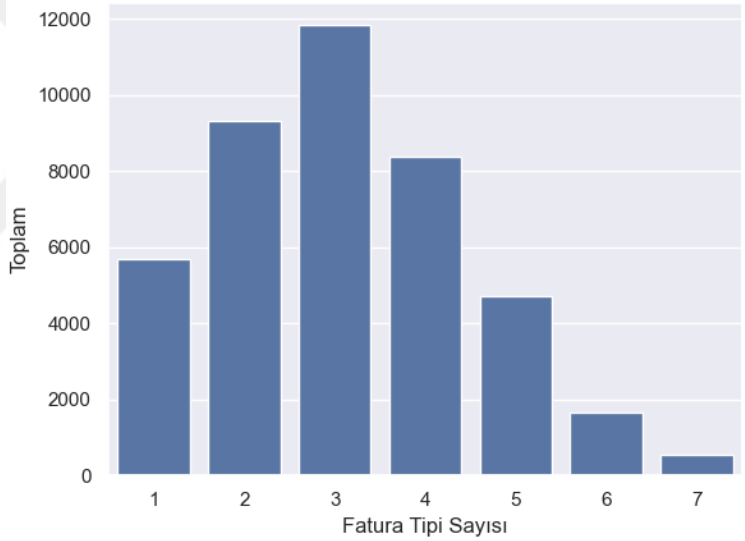
- **GEC_ODENEN_ORT_GUN:** Müşterinin geç ödediği fatura sayısının ve bu ödemelerin gecikme gün sayılarının oranıdır.

$$GEC_ODENEN_ORT_GUN = \frac{GEC_ODEME_GUN_TOPLAMI}{GEC_ODEME_SAYISI} \quad (3.4)$$

- **ERKEN_ODENEN_ORT_GUN:** Son ödeme tarihinden önce ödenmiş faturaların (ERKEN_ODEME_SAYISI), son ödeme tarihinden kaç gün önce ödendiğinin toplam sayısına (ERKEN_ODEME_GUN_TOPLAMI) bölünmesiyle elde edilmiştir.

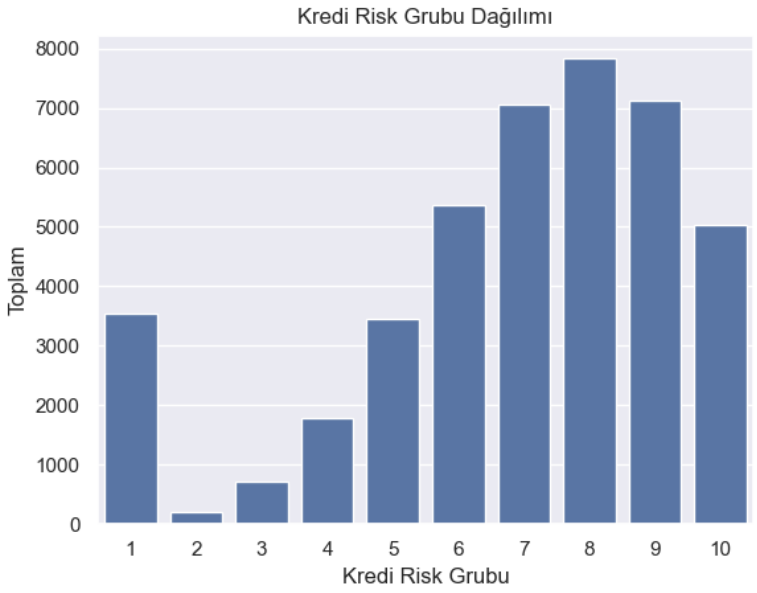
$$ERKEN_ODENEN_ORT_GUN = \frac{ERKEN_ODEME_GUN_TOPLAMI}{ERKEN_ODEME_SAYISI} \quad (3.5)$$

- **FATURA_TIPI_SAYISI:** Müşterinin ödeme kuruluşu aracılığıyla ödediği farklı fatura tiplerinin toplamı ile elde edilmiştir. Müşterinin ödeme yaptığı farklı fatura tipi sayılarının dağılımı Şekil 3.7'de gösterilmiştir.




**Şekil 3.7:** Fatura Tipi Sayısı Dağılımı

- **KREDI_SKORU:** Müşterinin KKB' den alınan gerçek kredi risk puanıdır. 1-10 arasında bir değere sahiptir. 1 en kötü ve 10 en iyi olacak şekilde gruplandırılmıştır. Şekil 3.8'de kredi skoru dağılımı gösterilmiştir.




**Şekil 3.8:** Kredi Skoru Dağılımı

Kullanıma hazır hale getirilen veri seti 13 özellik (12 girdi, 1 çıktı) ve 42.117 örnekten oluşmaktadır. Veri setinde yer alan her bir etiket için, birer örnek alınarak Tablo 3.4' te gösterilmiştir.

**Tablo 3.4:** Veri Setinden Örnekler

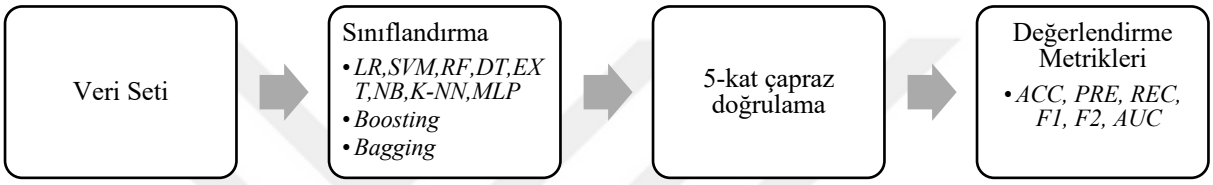
| G1 | G2 | G3 | G4 | G5 | G6 | G7 | G8 | G9 | G10 | G11 | G12 | Skor |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 45 | 4 | 5 | 0,62 | 1969,00 | 13 | 151,46 | 0,38 | 17,80 | 8,88 | 3 | **1** |
| 2 | 58 | 2 | 1 | 0,60 | 3007,86 | 11 | 273,44 | 0,27 | 13,00 | 7,38 | 4 | **2** |
| 1 | 51 | 1 | 2 | 0,70 | 869,70 | 11 | 79,06 | 0,00 | 0,00 | 0,00 | 3 | **3** |
| 1 | 26 | 3 | 4 | 0,80 | 955,30 | 3 | 318,43 | 0,33 | 12,00 | 5,50 | 3 | **4** |
| 1 | 46 | 4 | 5 | 0,54 | 1667,30 | 13 | 128,25 | 0,23 | 3,67 | 2,80 | 5 | **5** |
| 1 | 56 | 43 | 1 | 1,00 | 1660,95 | 13 | 127,77 | 0,77 | 15,90 | 6,67 | 1 | **6** |
| 1 | 66 | 187 | 1 | 1,00 | 1176,24 | 8 | 147.03 | 0.00 | 0.00 | 7,75 | 1 | **7** |
| 1 | 36 | 45 | 2 | 1,00 | 2845,2 | 10 | 284.52 | 0.10 | 3.00 | 9,89 | 3 | **8** |
| 2 | 49 | 220 | 3 | 0,92 | 5047,51 | 13 | 388,27 | 0,46 | 7,50 | 8,57 | 5 | **9** |
| 1 | 54 | 43 | 1 | 1,00 | 1858,92 | 12 | 154,91 | 0,08 | 2,00 | 8,55 | 2 | **10** |

## 3.2. YÖNTEM

Bölüm 3.1'de detaylı olarak verilen işlemler sonucunda elde edilen "Fatura Ödeme Alışkanlıkları" veri seti ile yapılan işlemler ve uygulanan yöntemlere ait detaylar bu başlıkta yer almaktadır.

### 3.2.1. Verinin Sınıflandırılması

Veri seti herhangi bir işlemden geçirilmeden önce sınıflandırma işlemi yapılmıştır. 5-kat çapraz doğrulama ile çeşitli değerlendirme metrikleri ile hesaplanmıştır. Bu adımda yapılan işlemler 3.9 'da özetlenmiştir.

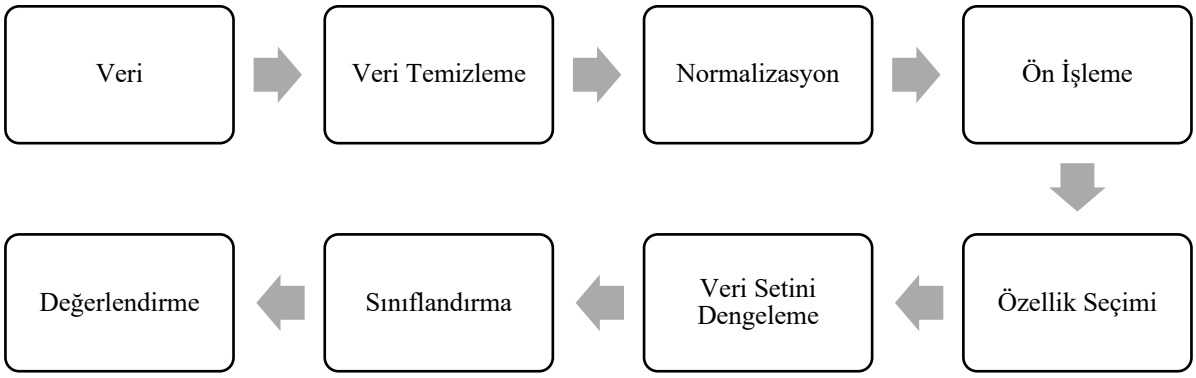



**Şekil 3.9:** Verinin Sınıflandırılması

Boosting algoritmalarından XGB için boosting tipi olarak; GBDT, DART ve GBLINEAR seçilmiştir. LGBM için ise GBDT, DART ve RF seçilmiştir. AdaBoost için öğrenici olarak; LR, DT, RF, EXT ve NB algoritmaları belirlenerek, sırayla testler yapılmıştır. Bagging yaklaşımı için temel öğrenci modelleri olarak; DT, RF, EXT, NB ve K-NN seçilmiştir. Bu adımda uygulanan işlemler sonrasında elde edilen sonuçlar Tablo 3.5' te detaylı olarak verilmiştir.

### 3.2.2. Verinin İşlenmesi ve Sınıflandırılması

Bu bölümde, veri seti Şekil 3.10'da gösterilen adımlar uygulanarak performans analizi yapılmıştır. Bu adımda veri seti makine öğrenmesi prensipleri kullanılarak işlenerek, verimli ve başarılı bir yöntem önerilmiştir.




**Şekil 3.10:** Verinin İşlenmesi ve Sınıflandırılması

**Veri Temizleme**

Veri seti temizleme aşaması aşağıdaki adımlardan oluşmaktadır.

**Eksik Veriler:** Çalışma kapsamında eksik veriler, Atama (Imputation) ve Silme (Deletion) yöntemleri kullanılarak işlenmiştir. Müşterilere ait bilgilerin bulunduğu tablodaki kayıtlarda, doğum tarihi alanlarının boş olması nedeniyle "Yaş" özelliği hesaplanamamıştır. Yaş özelliği eksik olan örnekler, ortalama yaş değeri ile doldurulmuştur. Herhangi bir nedenle bazı faturaların son ödeme tarihleri bulunmadığından, geç ödeme, erken ödeme gibi özelliklerin boş olması söz konusudur. Bu duruma sahip örnekler silinmiştir.

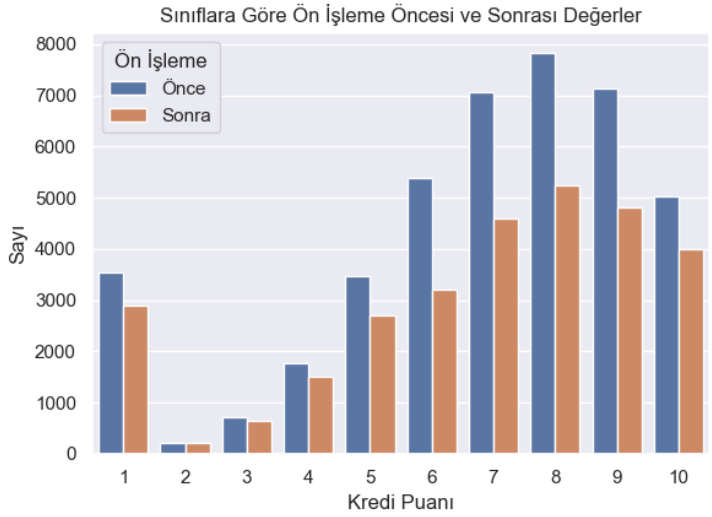
- **Tekrarlanan ve Aykırı Değerler:** Bazı müşterilerin ödeme yaptıkları faturalar için veri tabanına, birden fazla kayıt atıldığı görülmüş ve bu kayıtlar silinmiştir. Bazı müşterilerin birden fazla kayda sahip olduğu ve bu kayıtların birbirinden farklı bilgiler içerdiği görülmüştür. Bu kayıtlar silinerek veri setinden çıkarılmıştır. Müşteri tablosundaki Doğum Tarihi alanında yer alan bazı kayıtların, hatalı verilerle doldurulduğu gözlemlenmiştir. Bu neden Yaş özelliği aykırı ve yanlış hesaplanmıştır. Bu örnekler ortalama Yaş ile doldurulmuştur. Doğum tarihi alanında yer alan hatalı format veya tarihler nedeniyle aykırı (01.01.1800 vb.) olarak sınıflandırılan Yaş özelliğine ait örnekler ortalama değerlerle düzeltilmiştir. Fatura ödeme detaylarının yer aldığı tablolardaki bazı kayıtlarda ödenen tutar formatındaki bozukluklar nedeniyle aykırı hesaplanan tutarlara ait veriler silinmiştir. Çeşitli nedenlerle veri tabanına hatalı girilen, kayıtlar nedeniyle geç ödeme oranı ve nakit ödeme yüzdesi özelliklerinin aykırı

değerlere sahip olduğu görülmüştür (Nakit ödeme yüzdesi veya geç ödeme oranının %150 olması vb.). Bu kayıtlar silinerek veri setinden çıkarılmıştır.

- **Veri Yapısı ve Kalitesi Kontrolü:** Herhangi bir karşılığı olmayan ürün kodlarına ait örnekler silinmiştir. Herhangi bir fatura tipine karşılık gelmeyen değerlere sahip olan örnekler silinmiştir. Çeşitli nedenlerle ödeme tutarı, geç ödeme oranı, nakit ödeme yüzdesi alanlarında yer alan sayısal olmayan değerlere sahip örnekler silinmiştir.

**Normalizasyon:** Veri temizleme işleminde sonra, Min-Max normalizasyon ile büyük sayısal değerlere sahip özellikler, 0-1 aralığında olacak şekilde ölçeklendirilmiştir.

**Ön İşleme:** Veri temizleme ve ön işleme adımları sonrasında, veri seti 29.817 örnekten oluşmaktadır. Ön işleme adımı öncesi ve sonrasındaki etiket dağılımı Şekil 3.11' de gösterilmiştir.




**Şekil 3.11:** Ön İşleme Öncesi ve Sonrasındaki Etiket Dağılımı

**Özellik Seçimi:** Özellik seçim yöntemi olarak, ANOVA F-Test, Ki-Kare ve Karşılıklı Bilgi yöntemleri kullanılarak testler yapılmıştır.

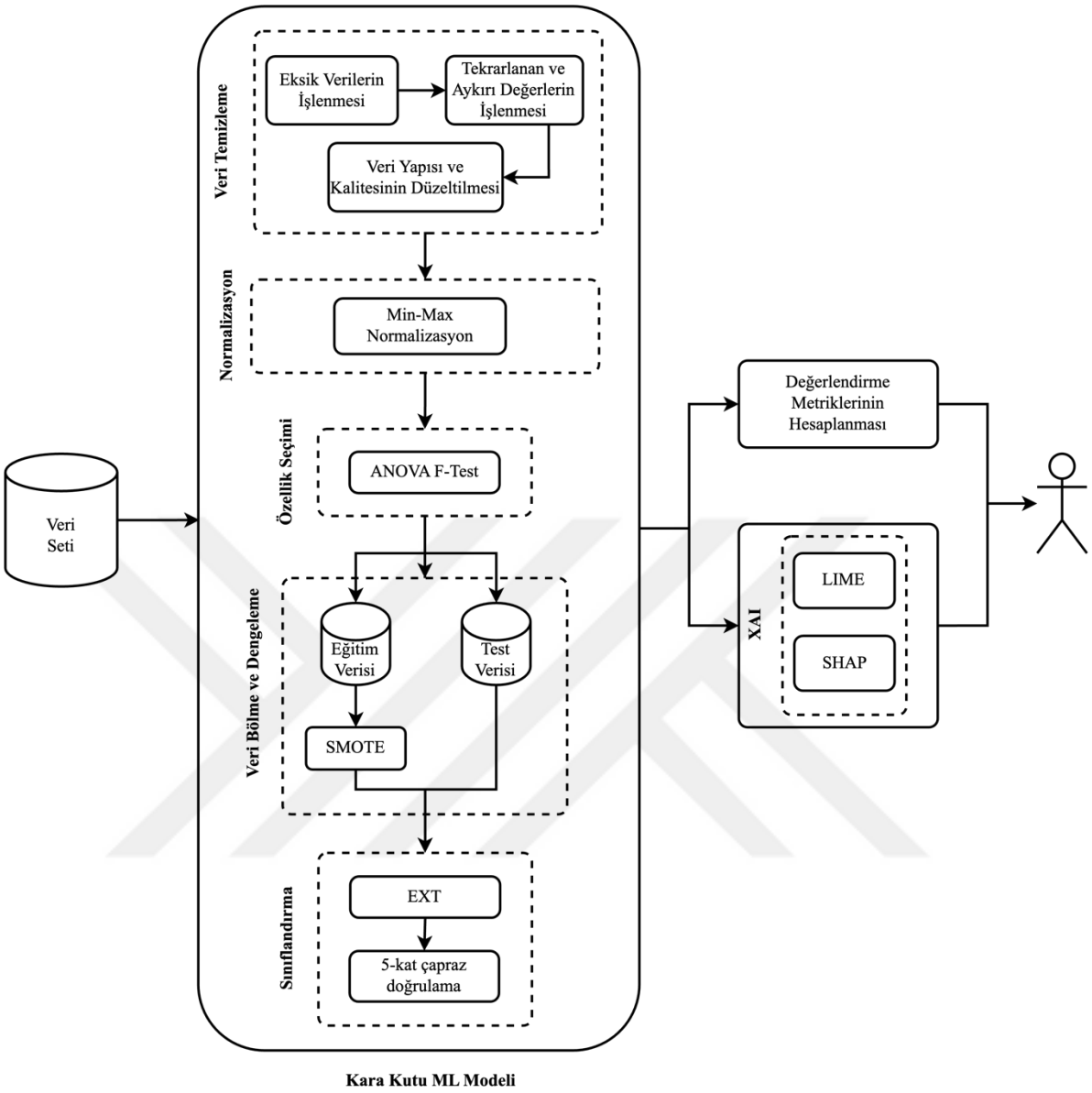
**Verinin Dengelenmesi:** Ön işleme adımları sonrasında veri seti, ham haline göre daha dengeli bir görünüme sahip olmasına rağmen, henüz dengeli bir dağılıma sahip değildir. Daha verimli sonuçlar elde etmek için eğitim verisine yapay örnekleme teknikleri uygulanarak, modellerin dengeli eğitilmesi amaçlanmıştır. Aşırı, yetersiz ve hibrit örnekleme yöntemleri kullanılmıştır.

**Sınıflandırma:** Modellerin değerlendirilmesi için, LR, SVM, RF, DT, EXT, NB, K-NN, MLP, Bagging ve Boosting sınıflandırma algoritmaları kullanılmıştır.

**Değerlendirme:** Yapılan deneyler sonrasında elde edilen en yüksek performans metriklerine değerlendirilmiştir.

### 3.2.3. XAI Yöntemleri ile En İyi ML Modelinin Açıklanması

Bir önceki adımda en yüksek performansa ait ML modeli, XAI yöntemlerinden LIME ve SHAP'e girdi olarak verilip, sonuçlar incelenmiştir. Bu başlıkta yapılan işlemler Şekil 3.12' de gösterilmiştir:




**Şekil 3.12:** En Başarılı ML Modelinin Açıklanması ve Yorumlanması Adımları

Şekil 3.12' de gösterilen Kara Kutu ML Modeli, en iyi performansa sahip modeldir. Bu modelin son kullanıcı tarafından anlaşılabilir ve yorumlanabilir bir forma dönüştürülmesi için XAI yöntemlerinden LIME ve SHAP kullanılmıştır. Her bir etiket için yapılan tahminlere ait LIME ve SHAP sonuçları incelenerek değerlendirmeler yapılmıştır.

## 3.3. ARAÇLAR

### 3.3.1. PyCharm Entegre Geliştirme Ortamı

PyCharm, Python için kullanılan ücretsiz ve profesyonel bir geliştirme ortamıdır. Windows, Linux ve MacOS ile uyumlu çalışabilmektedir. PyCharm, ihtiyaçlara veya kişisel tercihlere göre özelleştirilebilir. PyCharm, Python geliştirmeyi hızlandırmak ve verimliliği artırmak için çeşitli eklenti, uzantı, kit ve araç sağlar. Python programlama dili için kodlama, hata ayıklama, test etme ve analiz etme gibi özellikler sunar [87].

PyCharm sunduğu başlıca özellikler aşağıdaki gibi sıralanabilir:

- **Kod Tarama:** Kod tarama, yazılım geliştiricilerin bir sınıfa, özelliğe veya dosyaya erişimini kolaylaştırır. Sınıflar, dosyalar ve yöntemler arasında geçiş yapılabilir. Geliştiricilerin, Python kodunu düzenlemek ve incelemek için harcadıkları eforu en aza indirmeye yardımcı olur.

- **Kod Düzenleyici:** Kod düzenleyici, otomatik tamamlama, kod inceleme, hata ayıklama ve yeniden düzenleme gibi özellikler sunar. Böylece, kodun kalitesi ve doğruluğu arttırılarak, yazılım geliştirme sürecini hızlandırır.

- **Çoklu Teknoloji Geliştirme:** PyCharm, JavaScript, HTML, CSS ve TypeScript gibi yaygın kullanılan web teknolojilerini destekler. Ayrıca birçok veri tabanı yönetim sistemini de destekleyerek veri tabanı işlemlerinin gerçekleştirilmesine yardımcı olur.

- **Kod Analizi:** Geliştirilen uygulamalardaki kodların analizi ve hataların ayıklanmasını sağlayarak, hataların önceden tespit edilerek, çözülmesine yardımcı olur.

- **Entegrasyon Araçları Desteği:** Programlama için kolaylaştırıcı ve yararlı araçların entegrasyonuna izin vererek, kod yönetimi ve üretkenliğini arttırmaya yardımcı olur.

- **Versiyon Kontrol:** Git başta olmak üzere popüler versiyon kontrol sistemlerini destekler.

- **Test Otomasyonu:** Uygulamadaki otomatik testleri çalıştırır ve sonuçların incelenmesi için bir dizi araç sağlar.

- **Uzaktan Geliştirme:** PyCharm, yazılımları uzak ana veya sanal bilgisayarlarda çalıştırma olanağı sağlar.

PyCharm avantajları ve dezavantajları aşağıdaki gibi özetlenebilir:

Avantajlar;

- Kullanım kolaylığı,
- Çoklu platform desteği,
- Kod geliştirme ve verimliliği arttırma,
- Geliştiricilere sunulan birçok kodlama aracı,
- Gelişmiş hata analiz ve ayıklama yöntemleri,
- Geniş bir topluluğa sahip olması,
- Birçok sürüm kontrol sistemini desteklemesi,
- Çoklu dil ve veri tabanı sistemini desteklemesi.

Dezavantajlar;

- Profesyonel kullanım için kullanılacak sürümü ücretlidir.
- Yüksek depolama ve bellek alanına ihtiyaç duyabilir.
- Python Sanal Env. gibi araçlara düzenlemek karmaşık olabilir.

### 3.3.2. Python Programlama Dili

Pyhton; özellikle veri bilimi ve makine öğreniminde yaygın olarak kullanılan bir programlama dilidir. Yazılım geliştiriciler, etkili ve öğrenmesi kolay olduğu ve birçok farklı platformda çalıştırılabildiği için Python' ı tercih etmektedir. Python yazılımı ücretsiz olarak indirilebilir, her türlü sistemle iyi bir entegrasyon sağlar ve geliştirme hızını artırır [87], [88].

Python programlama dilinin en önemli avantajları şunlardır:

- Kolay anlaşılır bir söz dizimine sahip olduğundan Python programını kolayca okuyabilir ve anlayabilir.

- Python birçok görev için yeniden kullanılabilir geniş bir kitaplığa sahiptir. Geliştiriciler birçok işlemi bu kitaplıkları kullanarak gerçekleştirebilir.

- Birçok programlama diline kıyasla daha az kod satırı ile program yazabilme imkânı sunduğundan daha efektif bir ortam sağlar.
- Java, C ve C++ gibi diğer programlama dillerine kıyasla daha kolay öğrenebilir bir dildir.
- Geniş bir topluluğa sahiptir ve sürekli geliştirilmeye devam edilmektedir.
- Farklı işlem sistemlerinde çalışabilir.

Veri bilimi ve makine öğrenmesi konularında sıklıkla tercih edilen bir dildir. Tercih edilme sebeplerinden bazıları aşağıdaki gibidir:

- Veri temizleme, ayıklama, özellik seçme, veri etiketleme gibi makine öğrenmesi aşamaları için birçok araç sunmaktadır.
- Veri görselleştirme ile ilgili son derece gelişmiş yöntem ve araç desteği vardır.
- Makine öğrenmesi modellerini eğitmek ve verileri doğru sınıflandıran sınıflandırıcıların derlenmesi için hazır kitaplıklar bulundurmaktadır.
- Makine öğrenmesi ile ilgili farklı alanlar için (görüntü ve metin sınıflandırma, görüntü işleme vb.) birçok yöntem sunmaktadır.
- Derin öğrenme gibi gelişmiş yaklaşımlar için tercih edilmektedir.

Python programlama dilini öne çıkaran ve benzersiz kalın özellikler ise şunlardır:

- Python, yorumlanan bir dildir. Başka bir deyişle kodu doğrudan satır satır çalıştırmaktadır. Yazılan kodda hata çıktığında çalışmayı durdurur ve programcının hatayı hızlıca tespit etmesini sağlamaktadır.
- İngilizceye benzer ve kolay bir dildir. Diğer dillerde kullanılan süslü ayraçlar yerine girintileme kullanır.
- Değişken türlerini geliştirme aşamasında belirlemeye gerek yoktur. Çalışma anında otomatik olarak belirlenir. Böylece daha hızlı geliştirilebilen bir ortam sağlar.

- Diğer programlama dillerine göre günlük hayatta kullanılan dile daha yakındır. Yazılım geliştiricilere, mimari ve bellek yönetimi gibi temel işlevler konusunda kolaylık sağlar.

- Python, nesne tabanlı ve işlevsel programlama gibi birçok programlama tipini destekler.

Python'ı özellikle makine öğrenmesi alanında öne çıkaran sebeplerin biri de kendine özgü güçlü ve zengin kütüphanelere sahip olmasıdır. Bu çalışma kapsamında kullanılan kütüphaneler aşağıda listelenmiştir:

- **NumPy (Numerical Python– Sayısal Python):** Python'da bilimsel hesaplama için temel pakettir. Çok boyutlu bir dizi nesnesi, çeşitli türetilmiş nesneler (maskeli diziler ve matrisler gibi) ve matematiksel, mantıksal, şekil işleme, sıralama, seçme, G/Ç, ayrık Fourier dönüşümleri, temel doğrusal cebir, temel istatistiksel işlemler, rastgele simülasyon ve çok daha fazlası dahil olmak üzere diziler üzerinde hızlı işlemler için çeşitli rutinler sağlayan bir Python kütüphanesidir.

- **SciPy (Scientific tools for Python– Python için Bilimsel Araçlar):** Bilimsel ve teknik hesaplama için kullanılan bir Python kitaplığıdır. SciPy, optimizasyon, lineer cebir, entegrasyon, enterpolasyon, özel fonksiyonlar, FFT, sinyal ve görüntü işleme, ODE çözücüler ve bilim ve mühendislikte yaygın olan diğer görevler için modüller içerir.

- **Matplotlib (Mathematical Plot Library– Matematiksel Çizim Kütüphanesi):** Python programlama dili ve NumPy için bir çizim kitaplığıdır. Tkinter, wxPython, Qt veya GTK gibi genel amaçlı grafiksel kullanıcı arayüzü araç setlerini kullanan uygulamalara grafikleri yerleştirmek için nesne yönelimli bir API sağlar.

- **SciKit-Learn (Science Kit/Kits Learn– Bilim Öğrenim Kiti):** Makine öğrenmesi modelleri oluşturmak ve değerlendirmek için kullanılan bir Python kitaplığıdır. Sınıflandırma, kümeleme, regresyon, özellik ve model seçimi, ölçekleme ve ön işleme gibi özellikler sunmaktadır.

- **Pandas:** Veri analiz ve işleme için kullanılan bir kitaplıktır. Zaman serileri ve sayısal tabloları işlenmesi için bir yapı oluşturmak ve analiz işlemlerinde sıklıkla kullanılır.

- **Seaborn:** İstastiksel veri görselleştirme amacıyla kullanılan bir kitaplıktır. Matplotlib üzerine inşa edilmiştir. Veri analizi ve görselleştirme için sıklıkla kullanılır.

- **Imblearn (Imbalanced-learn– Dengesiz Öğrenme):** Makine öğrenmesi uygulamalarında karşılaşılan sınıf dengesizliği sorunları için kullanılan bir kitaplıktır. Bu sorunları çözmek için farklı yöntemler sunar.

- **LIME (Local Interpretable Model Agnostic Explanations- Yerel Yorumlanabilir Model Agnostik Açıklamalar):** Makine öğrenmesi modelleri tarafından üretilen çıktıları anlamak ve açıklamak için kullanılan bir Python kütüphanesidir. Karmaşık makine öğrenmesi modelleri, genellikle "kara kutu" olarak adlandırılır. Bu modellerin kararlarının anlaşılması ve yorumlanması zor olabilir. LIME, bu tür modellerin çıktılarının anlaşılması için bir kitaplık sunar.

- **SHAP (Shapley Additive Explanations- Shapley Katkı Açıklamaları):** Makine öğrenmesi modellerinin açıklanabilirliği ile ilgili bir diğer Python kütüphanesidir. Shapley değerleri adı verilen ve oyun teorisine dayana bir yöntem kullanarak, her özelliğin model tahminine ne kadar katkıda bulunduğunu hesaplar. Böylece anlaşılır ve yorumlanabilir bir çıktı sağlar.

### 3.3.3. Microsoft SQL Server

Microsoft SQL Server, Microsoft tarafından geliştirilen ve yönetilen İlişkisel Veri Tabanı Yönetim Sistemi (Relational Database Management System- RDMS)' dir. İlişkisel veri tabanı sistemlerinde veriler tablolar halinde tutulur ve kullanılan tablolar arasında mantıksal bir bağlantı bulunur. Küçük veya büyük ölçekli uygulamalar için sıklıklar tercih edilir. Veri depolama, işleme, analiz ve raporlama gibi ihtiyaçları karşılar [89].

Bazı kullanım alanları aşağıdaki gibidir:

- **İşletme Veri tabanları:** Kurumsal şirketlerin iş süreçleri için merkezi yönetilen veri tabanları oluşturmak.

- **Veri Ambarları:** Büyük verilerin analiz ve raporlanması için ver ambarları oluşturmak.

- **Web Tabanlı Uygulamalar:** Web uygulamalarında kullanılmak üzere arka uç veri tabanları sağlamak.

- **Veri Entegrasyonu:** Dağıtık veri kaynaklarının entegrasyonunu yapmak ve veri alış-verişini yönetmek.

- **İş Zekâsı:** Veri analizi ve raporlama uygulamaları ile işletmelere çözümler sunmak.

# 4. BULGULAR

Bir önceki başlıkta detaylı olarak açıklanan malzeme ve yöntemler kullanılarak elde edilen bulgular bu başlık altında incelenmiştir.

Veri seti herhangi bir ön işleme adımı uygulanmadan sınıflandırıldığında Tablo 4.1' deki sonuçlar elde edilmiştir.

**Tablo 4.1:** Ham Veri ile Sınıflandırma Sonuçları

| Sınıflandırma Yöntemi | Değerlendirme Metrikleri (%) | | | | | |
|---|---|---|---|---|---|---|
| | ACC | PRE | REC | F1 | F2 | AUC |
| LR | 29,12 | 25,55 | 29,12 | 26,56 | 27,91 | 72,73 |
| SVM | 30,19 | 26,46 | 30,19 | 26,40 | 28,25 | 68,32 |
| DT | 42,72 | 43,02 | 42,85 | 42,67 | 42,74 | 68,12 |
| RF | 47,65 | 48,38 | 47,52 | 47,06 | 47,18 | 85,93 |
| EXT | 47,07 | 47,31 | 47,11 | 46,60 | 46,87 | 85,46 |
| NB | 20,60 | 14,70 | 20,60 | 13,57 | 16,19 | 69,38 |
| K-NN | 33,77 | 32,86 | 33,77 | 32,86 | 33,29 | 71,03 |
| MLP | 44,37 | 45,71 | 44,48 | 43,98 | 44,38 | 85,57 |
| **GB** | | | | | | |
| GBDT | 50,69 | 53,65 | 50,69 | 50,53 | 50,38 | 85,75 |
| **XGB** | | | | | | |
| GBT | 49,65 | 53,57 | 49,65 | 49,53 | 49,27 | 85,10 |
| DART | 49,65 | 53,57 | 49,65 | 49,53 | 49,27 | 85,10 |
| GBLIN. | 18,59 | 3,46 | 18,59 | 5,83 | 9,91 | 66,79 |
| **LGBM** | | | | | | |
| GBDT | 52,86 | 54,68 | 52,86 | 52,73 | 52,63 | 87,23 |
| DART | 52,85 | 55,42 | 52,85 | 52,70 | 52,56 | 87,11 |
| RF | 48,99 | 53,13 | 48,99 | 48,75 | 48,57 | 82,13 |
| **AdaBoost** | | | | | | |
| LR | 18,23 | 16,61 | 18,23 | 12,43 | 14,62 | 58,56 |
| DT | 42,79 | 42,92 | 42,75 | 42,83 | 42,81 | 66,62 |
| RF | 47,42 | 48,30 | 47,43 | 46,92 | 47,07 | 69,48 |
| EXT | 46,99 | 47,44 | 47,13 | 46,35 | 46,85 | 69,15 |
| NB | 19,71 | 15,50 | 19,71 | 13,82 | 16,04 | 58,34 |
| **Bagging** | | | | | | |
| DT | 46,12 | 46,28 | 46,42 | 46,30 | 46,71 | 79,79 |
| RF | 48,02 | 49,94 | 47,98 | 47,39 | 47,66 | 83,50 |
| EXT | 47,43 | 49,12 | 47,68 | 47,03 | 47,29 | 82,68 |
| NB | 20,47 | 14,38 | 20,63 | 13,55 | 16,08 | 66,75 |
| K-NN | 34,10 | 33,58 | 34,12 | 33,51 | 33,85 | 71,64 |

Tablo 4.1 incelendiğinde en iyi performansın LGBM ile elde edildiği görülmüştür. Bu yöntemle %53 doğruluk ve %87 AUC hesaplanmıştır. En yüksek ikinci doğruluk ise %50 ile GB yöntemiyle elde edilmiştir. RF ve EXT algoritmaları da birbirine yakın ve iyi performans göstersen sınıflandırma yöntemleri olarak öne çıkmıştır.

Ham veri ile elde edilen sonuçlar incelendikten sonra, veri setine Şekil 3.10' da gösterilen veri temizleme, normalizasyon, özellik seçimi ve yapay örnekleme adımları uygulanmıştır. Kullanılan özellik seçimi yöntemleri ile elde edilen en yüksek oranlar Tablo 4.2' de özetlenmiştir.

**Tablo 4.2:** Ön İşleme ve Örnekleme Adımları Sonrası Elde Edilen En İyi Sonuçlar

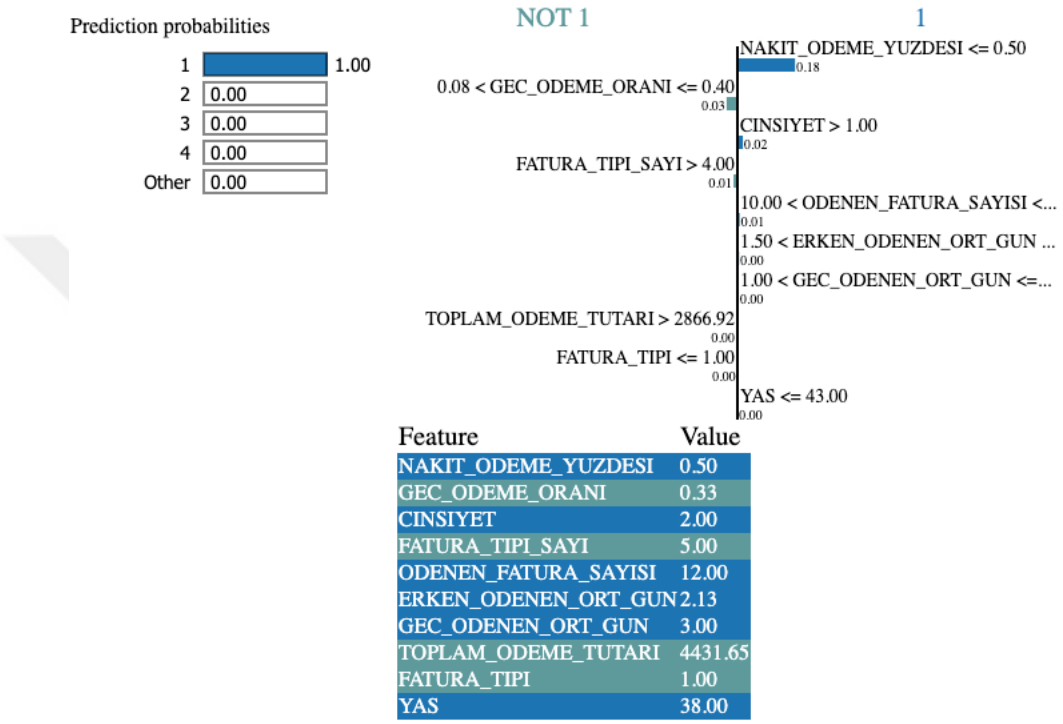
| **Özellik Seçimi** | **Seçilen Özellikler** | **Örnekleme Yöntemi** | **Sınıflandırma Yöntemi** | **ACC (%)** | **PRE (%)** | **REC (%)** | **F1 (%)** | **F2 (%)** | **AUC (%)** |
|---|---|---|---|---|---|---|---|---|---|
| ANOVA F-Test | Cinsiyet, Yaş, Fatura Tipi, Nakit Ödeme Yüzdesi, Ödenen Fatura Sayısı, Toplam Ödeme Tutarı, Geç Ödeme Oranı, Geç Ödenen Ort. Gün, Erken Ödenen Ort. Gün, Fatura Tipi Sayısı. | SMOTE | EXT | 80,49 | 79,89 | 80,42 | 79,83 | 80,11 | 97,04 |
| Ki-Kare | Cinsiyet, Yaş, Ürün Kodu, Fatura Tipi, Nakit Ödeme Yüzdesi, Ödenen Fatura | SMOTE | EXT | 75,88 | 75,89 | 75,88 | 75,39 | 75,58 | 96,01 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Sayısı, Geç Ödeme Oranı, Geç Ödenen Ort., Gün, Erken Ödenen Ort. Gün, Fatura Tipi Sayısı. | | | | | | | | |
| Karşılıklı Bilgi | Yaş, Ürün Kodu, Nakit Ödeme Yüzdesi, Ödenen Fatura Sayısı, Toplam Ödeme Tutarı, Geç Ödenen Ort., Gün, Erken Ödenen Ort. Gün, Fatura Tipi Sayısı, Ödeme Tutar Ort., Geç Ödeme Oranı, | SMOTE | EXT | 79,09 | 78,40 | 78,97 | 78,35 | 78,58 | 96,73 |

Tabloya göre en iyi performans ANOVA F-Test özellik seçimi ile elde edilmiştir. Bu yöntemle 12 adet özellikten 10 tanesi kullanılmıştır. Ürün kodu ve ödenen ortalama tutar özellikleri veri setinden çıkarılmıştır. Örnekleme yöntemi olarak ise SMOTE kullanılarak eğitim verisi dengelenmiştir. Tüm özellik seçim yöntemlerinde en iyi performans EXT sınıflandırma algoritması ile hesaplanmıştır. Doğruluk oranı, %80,49, kesinlik %79,89, duyarlılık %80,49, F1 ve F2 skorlar ise 79,83 ve %80,11 olarak ölçülmüştür. AUC ise %97,04' tür.

## LIME Yöntemi ile Değerlendirmeler

Bu yöntemde veri setinden seçilecek örnekler için, ilgili sınıf değerine tahmin olasılıkları ve bu tahmini gerçekleştirmek için faydalanılan özellik ve değerleri incelenebilmektedir. Veri setinde bulunan 10 adet sınıf için, doğru tahmin edilen örnekler rastgele seçilerek LIME sonuçlarına yer verilmiştir.




**Şekil 4.1:** Kredi Skoru=1 Olan Örnek için LIME Sonucu

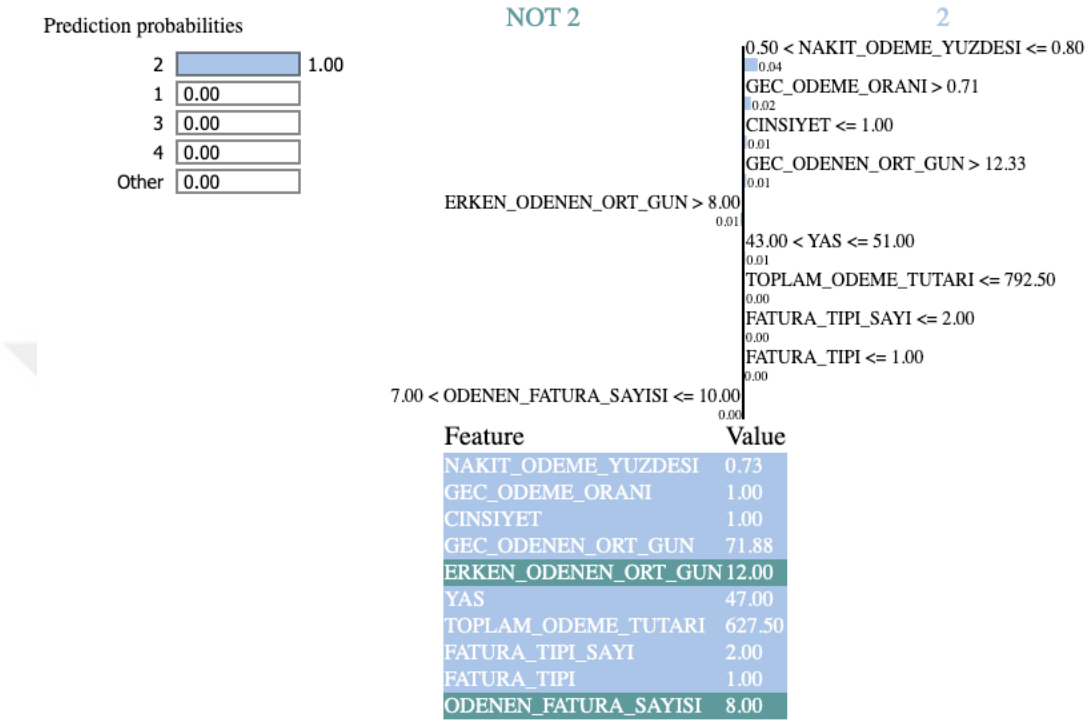
LIME çıktısı, soldan sağa üç ana bilgi içermektedir: (1) modelin tahminleri, (2) özelliklerin katkıları ve (3) her özelliğin, seçilen örnekteki gerçek değeridir.

(1) Seçilen örneğin kredi skorunun; %100 olasılıkla "1" olabileceği tahmin edilmiştir.

(2) Bu alanda yer alan bilgilerde modelin "1" tahminini yaparken örnekteki değerlere göre karar durumu özetlenmiştir. Yeşil renk ile "NOT 1" başlığı altındaki özellikler, modelin "1" olmama durumuna işaret eden özelliklerdir. Mavi renk ile yazılan "1" başlığındaki özellikler ise modelin tahmin yaparken "1" olma olasılığına katkı sağlayan özelliklerdir. Model incelendiğinde; nakit ödeme yüzdesi, geç ödeme oranı, cinsiyet, fatura tipi sayısı ve ödenen fatura sayısı en önemli beş özelliktir. Modele göre, nakit ödeme yüzdesinin 0,50'den düşük veya eşit olması, cinsiyetin Kadın, ödenen fatura sayısının 10 ve ödenen fatura sayısının 10-13 aralığında olması tahminde

en önemli rolleri üstlenmiştir. Geç ödeme oranının 0,08 ve 0,40 olması, fatura tipi sayısının 4'ten büyük olması ise model tahminin "1" olmamasına katkı sağlayan özelliklerdir.

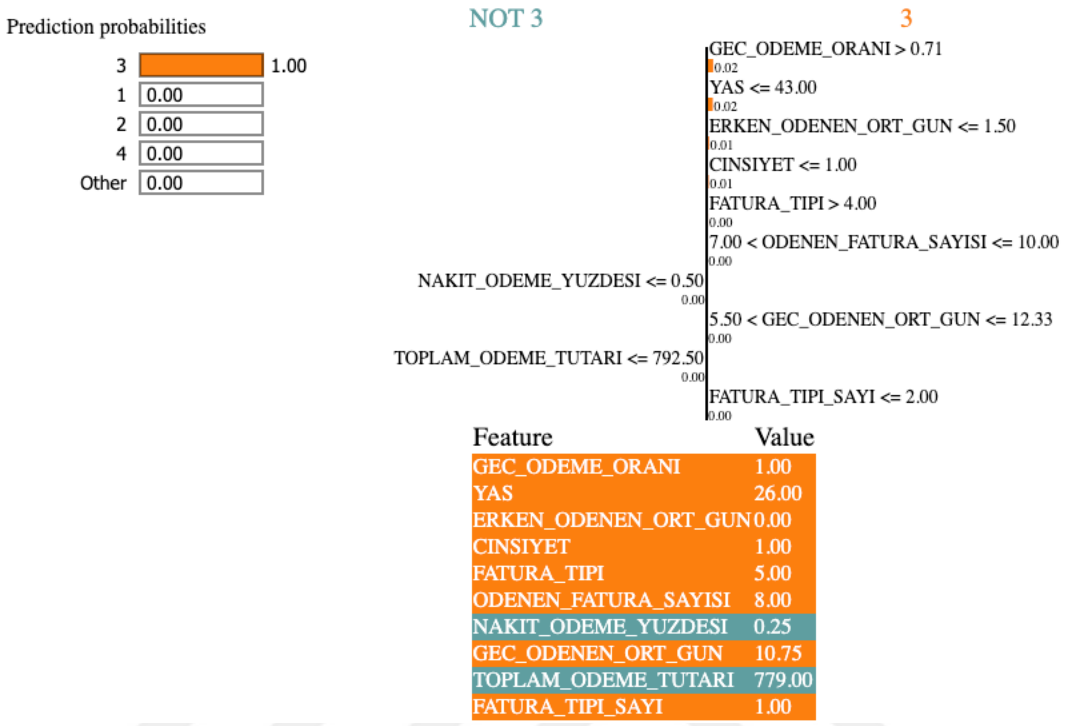
(3) Üçüncü kısımda ise örneğin gerçek değerleri yer almaktadır.




**Şekil 4.2:** Kredi Skoru=2 Olan Örnek için LIME Sonucu

Şekil 4.2' de LIME çıktısı verilen örnek için tahmin %100 oranında "2" dir. Nakit ödeme yüzdesinin 0,50 ve 0,80 arasında, geç ödeme oranının 0,71'den büyük, cinsiyetin Erkek, geç ödenen ortalama gün sayısının 12,33'den büyük ve yaşın 43-51 arasında olması en önemli

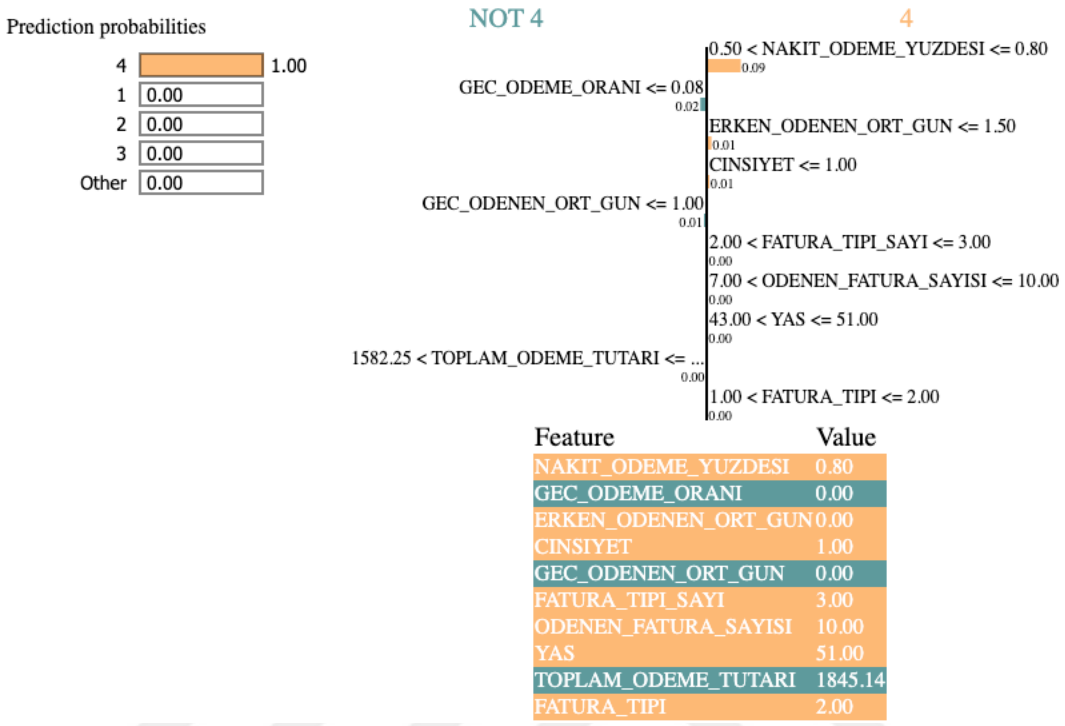
özelliklerdir. Erken ödenen ortalama gün sayısı ve ödenen fatura sayısı özellikleri ise tahmine negatif etkili olmuştur.




**Şekil 4.3:** Kredi Skoru=3 Olan Örnek için LIME Sonucu

Şekil 4.3' teki çıktıya göre tahmin kesin olarak "3" tür. Geç ödeme oranının 0,71'den büyük, yaşın 43'ten küçük, erken ödenen ortalama gün sayısının 1,5' ten küçük ve cinsiyetin Erkek

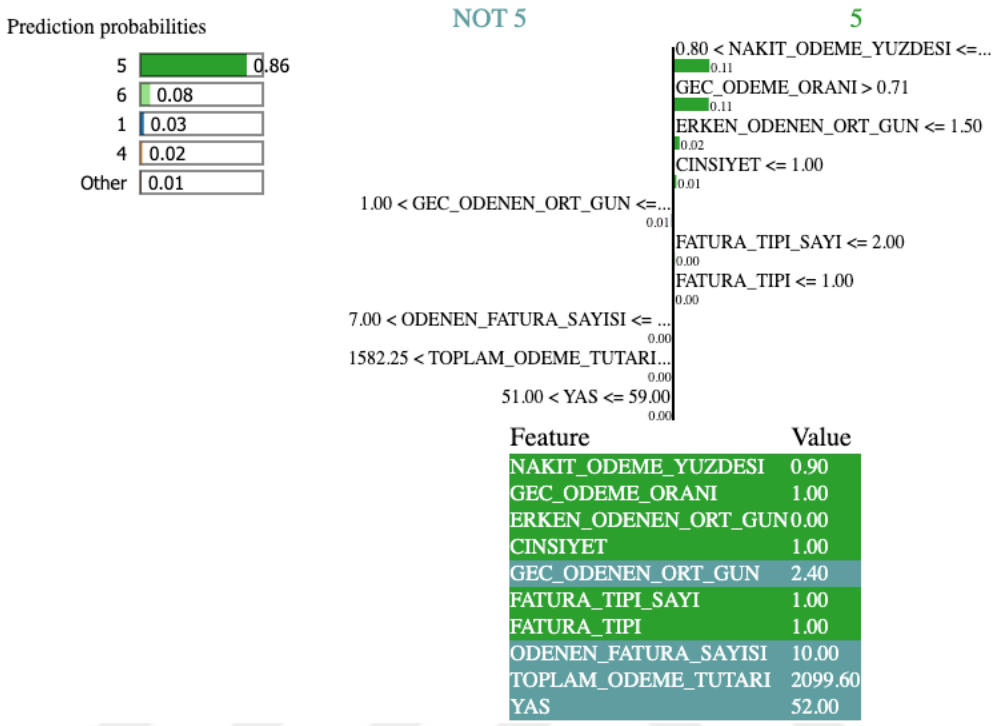
olması en önemli özellikler olarak sıralanmıştır. Nakit ödeme yüzdesi ve toplam ödeme tutarı özellikleri ise model tahminin "3" olmama durumuna katkıda bulunmuştur.




**Şekil 4.4:** Kredi Skoru=4 Olan Örnek için LIME Sonucu

Bu örnek incelendiğinde, tahminin "4" olduğu görülmektedir. Bu tahmin yapılırken kullanılan en önemli özellikler turuncu renk ile belirtilmiştir. Nakit ödeme yüzdesinin 0,80'e küçük veya eşit, erken ödenen ortalama gün sayısının 1,5' ten küçük ve cinsiyetin Erkek olması en önemli

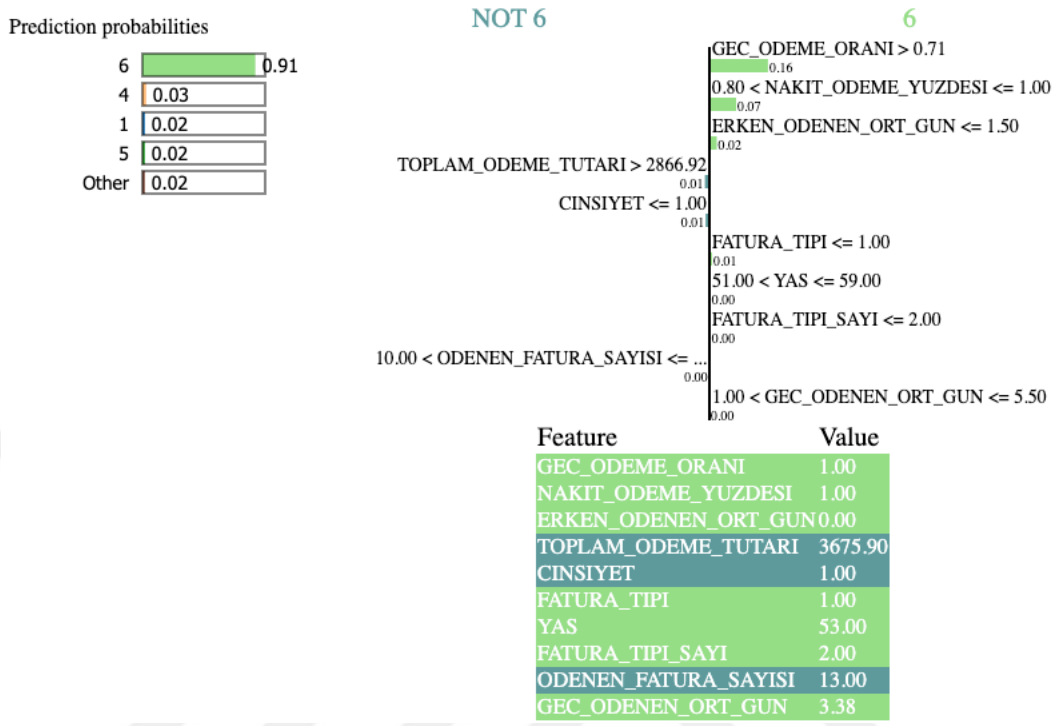
üç özelliktir. Geç ödeme oranı, geç ödenen ortalama gün sayısı ve toplam ödeme tutarı özellikleri modelin "4" kararının verirken kullanmadığı özelliklerdir.




**Şekil 4.5:** Kredi Skoru=5 Olan Örnek için LIME Sonucu

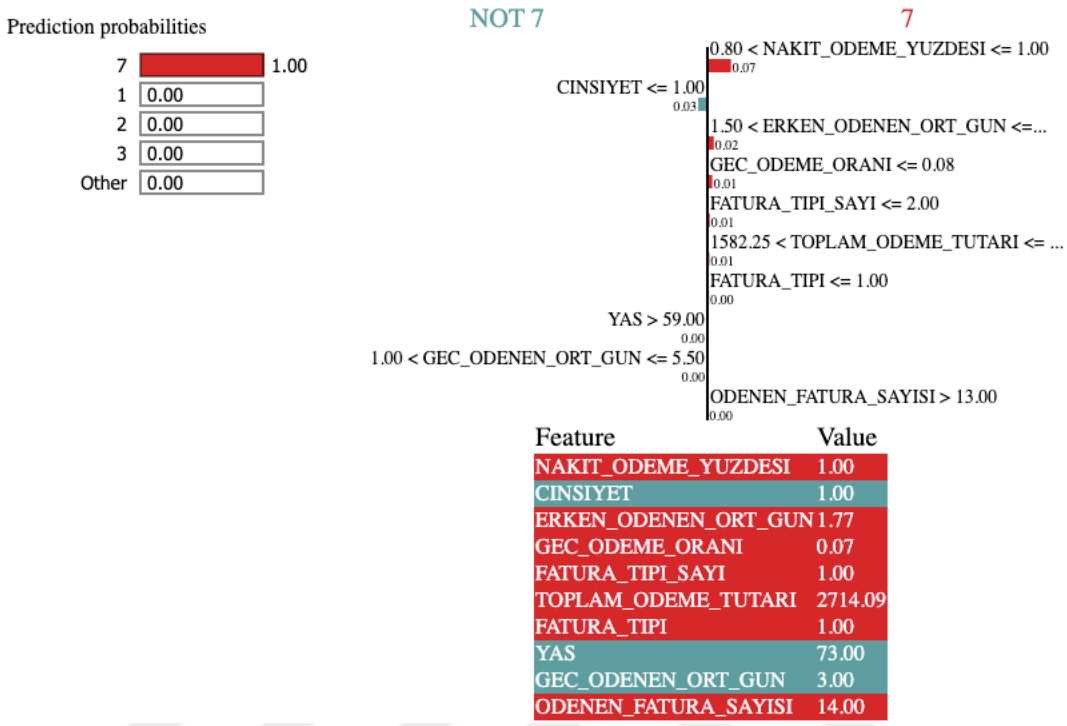
Modele göre, seçilen örnek %86 oranında "5", %8 oranında "6", %3 oranında "1" ve %2 oranında "4" sınıfına aittir. "5" sınıfı için tahmin yaparken; nakit ödeme yüzdesinin 0,8 ve 1 aralığının, geç ödeme oranın 0,71'den büyük, erken ödenen ortalama gün sayısının 1,50'den küçük ve cinsiyetin Erkek olması en önemli özelliklerdir. Geç ödenen ortalama gün sayısı,

ödenen fatura sayısı, toplam ödeme tutarı ve yaş özellikleri ise seçilen örneğin, diğer sınıflara ait olma olasılığına katkı sağlamıştır.




**Şekil 4.6:** Kredi Skoru=6 Olan Örnek için LIME Sonucu

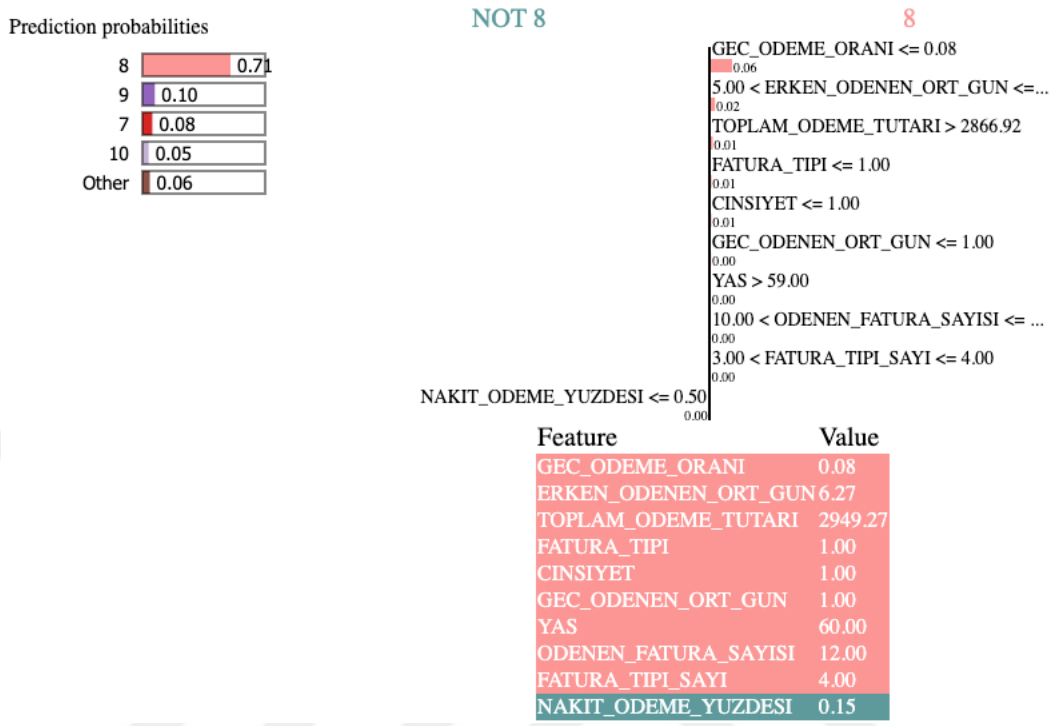
Bu örnekte ise modelin tahmini %91 oranında “6” olmuştur. Bu örnekte yer alan özelliklere göre tahmin, %3 oranında “4”, %2 oranında “1”, “5” veya diğer sınıflara aittir.




**Şekil 4.7:** Kredi Skoru=7 Olan Örnek için LIME Sonucu

Şekil 4.7’te gösterilen örnekte ise tahmin %100 olarak “7” sınıfı olarak yapılmıştır. Nakit ödeme yüzdesinin 0,80-1 aralığında, erken ödenen ortalama günün 1,50- 5 aralığında, geç ödeme oranının 0,08’e küçük veya eşit, fatura tipi sayısının 2’den küçük, toplam ödeme tutarının 1582,25- 2866,92 aralığında olması en önemli özellikler olarak sıralanmıştır. “NOT

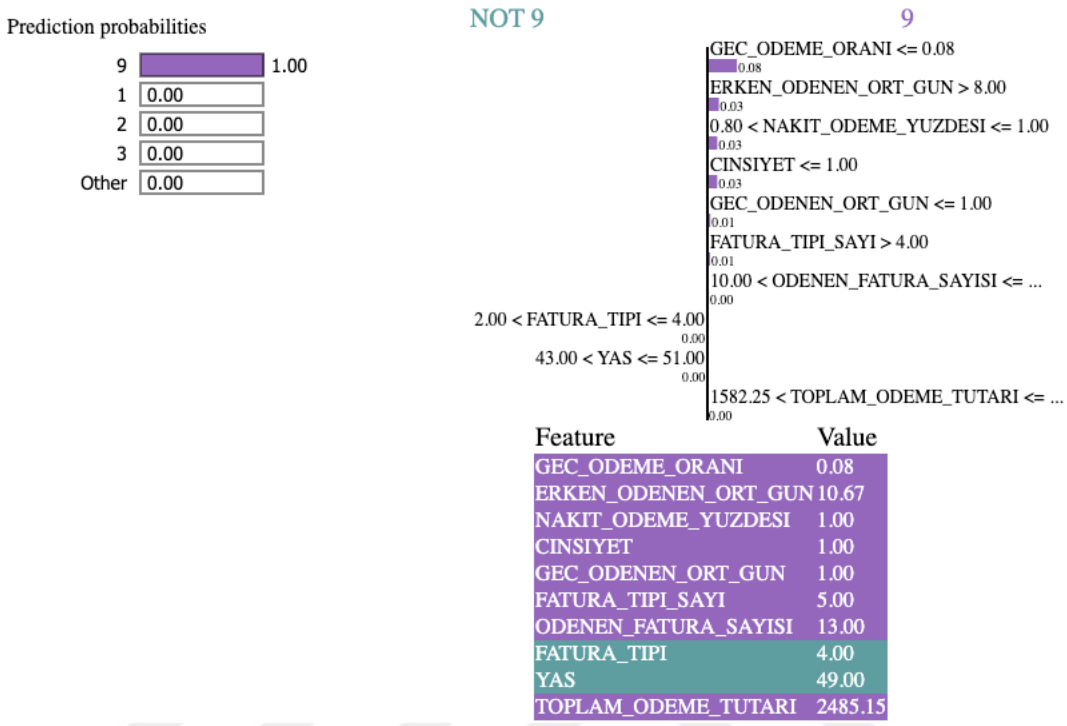
7” başlığı altında verilen özellik ve değerler ise “7” dışındaki sınıfların tahmini için kullanılmıştır.




**Şekil 4.8:** Kredi Skoru=8 Olan Örnek için LIME Sonucu

Bu örnekte ise model, %71 oranında “8” tahmininde bulunmuştur. Seçilen örnek, %10 olasılıkla “9”, %8 olasılıkla “7”, %5 olasılıkla “10” ve %6 olasılıkla diğer sınıflara ait olabilir. Geç ödeme oranının 0,08’den küçük, erken ödenen ortalama günün 5- 8 aralığında, toplam ödeme tutarının

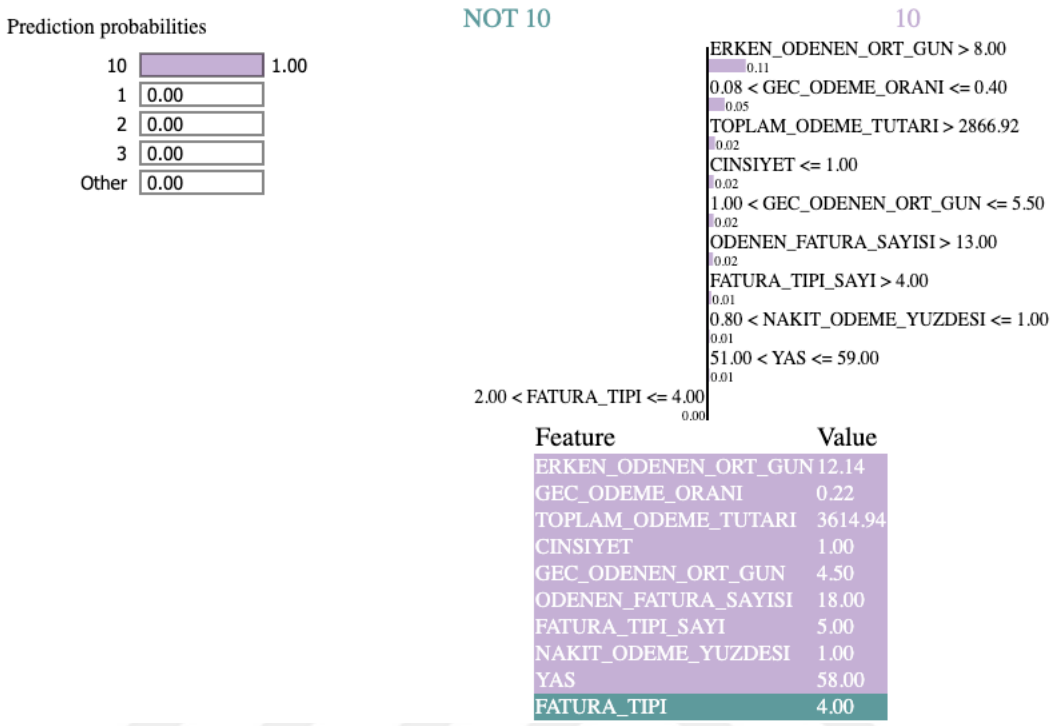
2866,92'den büyük, fatura tipinin Elektrik ve cinsiyetin Erkek olması modelin tahminin "8" olmasını sağlayan özellik ve değerlerdir.




**Şekil 4.9:** Kredi Skoru=9 Olan Örnek için LIME Sonucu

Bu örnek için ise model, %100 ihtimalle "9" sınıfı için tahmin yapmıştır. Geç ödeme oranı, erken ödenen ortalama gün sayısı, nakit ödeme yüzdesi, cinsiyet, geç ödenen ortalama gün

sayısı ve fatura tipi sayısı en verimli özellikler olarak sıralanmıştır. Fatura tipinin İnternet ve yaşın 43-51 arasında olması tahmine katkı sağlamamıştır.



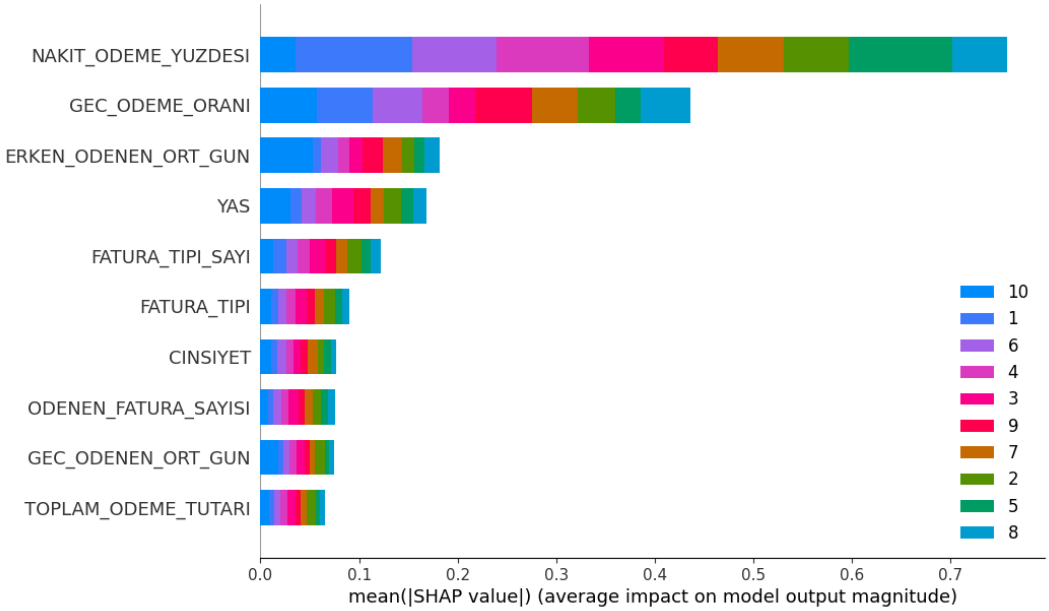

**Şekil 4.10:** Kredi Skoru=10 Olan Örnek için LIME Sonucu

Şekil 4.10'daki örnek için model "10" tahminini yapmıştır. En önemli özellikler, Erken Ödenen Ortalama Gün Sayısı, Geç Ödeme Oranı, Toplam Ödeme Tutarı ve Cinsiyet olarak sıralanmıştır. Fatura tipinin GSM olması için bu tahmin için kullanılmamıştır.

### SHAP Yöntemi ile Değerlendirmeler

SHAP yöntemi ile makine öğrenmesi modeli ile ilgili genel açıklamalar yapılarak, ilgili modelin açıklanabilirliği ve yorumlanabilirliği incelenebilir.

**Özellik Önem Grafiği:** Bu yöntemde modele dair temel bir fikir oluşturmak amacıyla özelliklerin önemi incelenmiştir. SHAP değerleriyle hesaplanan özellik önemi (büyük mutlak Shapley değerlerine sahip özellikler önemlidir) ile her ağaçtaki ortalama ve standart sapması olarak hesaplanan özellik önemi arasındaki karşılaştırmak amaçlanmıştır.

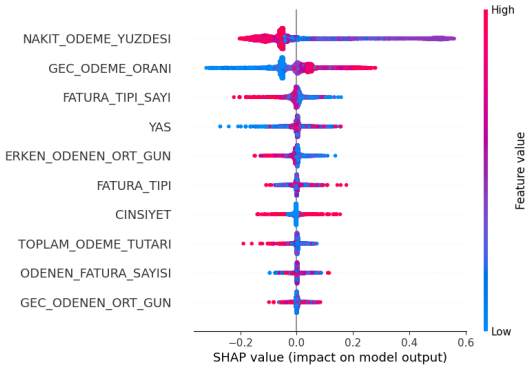



**Şekil 4.11:** Veri Setine Ait SHAP Özellik Önem Grafiği

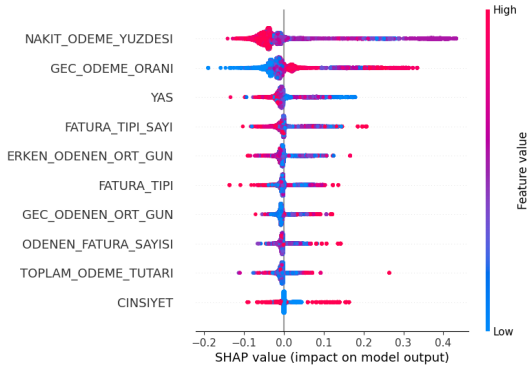
Önem grafiğinin Y ekseninde önem sırasına göre özellikler yer almaktadır. Her sınıfa ait renkler ile ilgili özelliğin önemi gözlemlenebilmektedir.

Özet grafiği incelendiğinde Nakit Ödeme Yüzdesi, Geç Ödeme Oranı, Erken Ödenen Ortalama Gün Sayısı, Yaş ve Fatura Tipi Sayısı en önemli ilk beş özellik olmuştur. Müşterinin ödediği toplam fatura tutarı ise önem sırası bakımından son sırada yer almıştır.

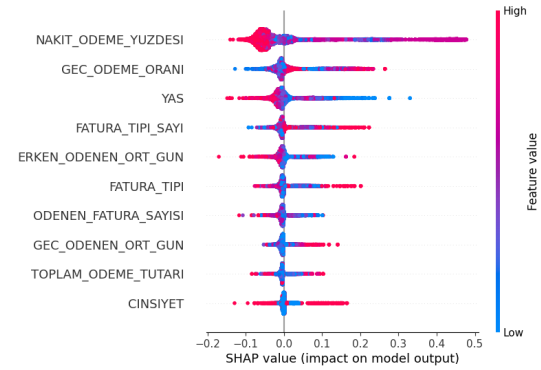
**Belirli Bir Etiket Değeri için Özet Grafiği:** Bu yaklaşımı kullanmak, her özelliğin belirli bir sonuç (etiket) üzerindeki etkisine ilişkin daha ayrıntılı bir genel bakış sağlayabilir. Her bir etiket değeri için özellik önemi incelenebilir. Çalışmamızda 1-10 arasında değişen 10 adet etiket değeri mevcuttur. 1 en kötü, 10 en iyi kredi skoru olacak şekilde bir dağılım söz konusudur. Her bir etiket için özellik önem grafikleri hazırlanmıştır.



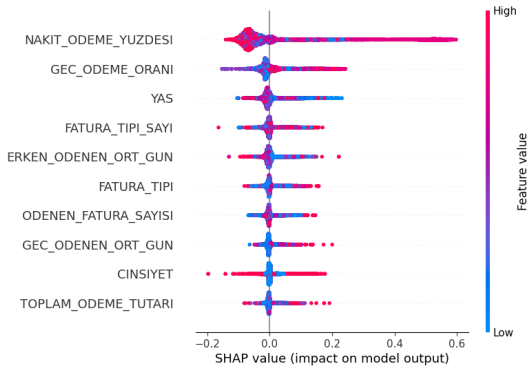

**a.** Kredi Skoru=1



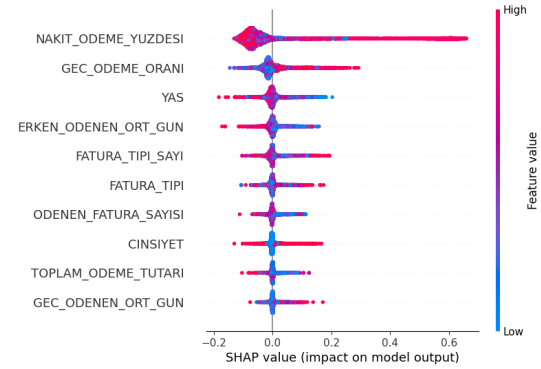

**b.** Kredi Skoru=2



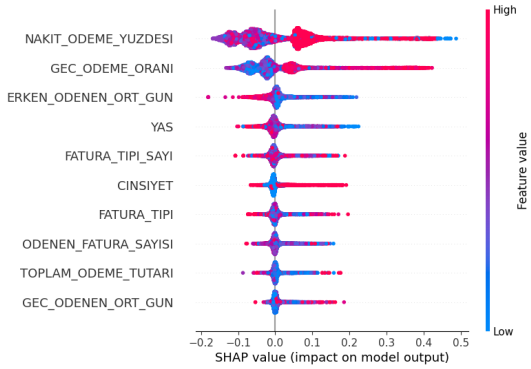

**c.** Kredi Skoru=3



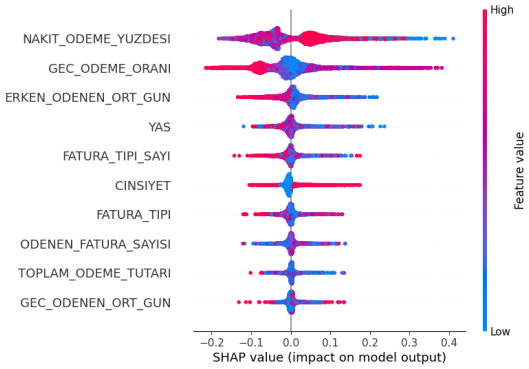

**d.** Kredi Skoru=4



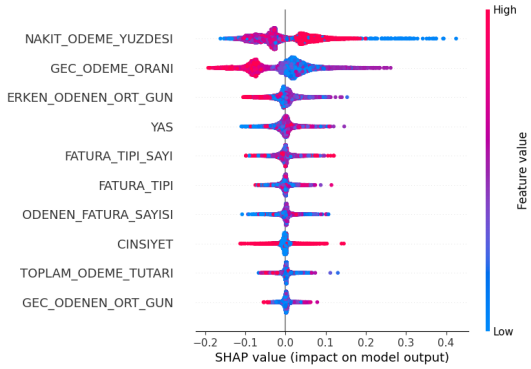

**e.** Kredi Skoru=5



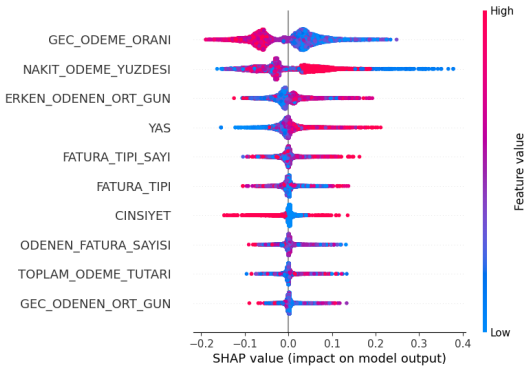

**f.** Kredi Skoru=6



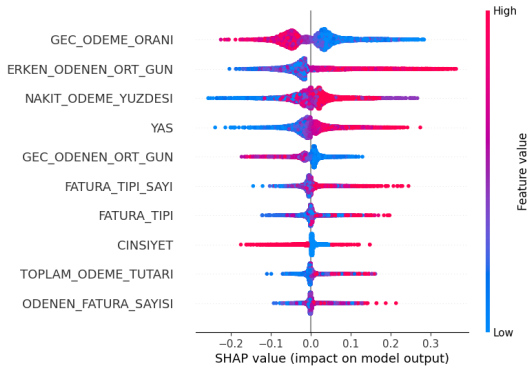

**g.** Kredi Skoru=7




**h.** Kredi Skoru=8




**i.** Kredi Skoru=9




**j.** Kredi Skoru=10

**Şekil 4.12:** Sınıf Etiketleri için SHAP Özet Grafikleri

Grafik detayları aşağıdaki gibi özetlenebilir:

- Y ekseni, model eğitiminde kullanılan özellikleri belirtmektedir. Sonuca en çok etki eden özellikler önem sırasına göre yukarıdan aşağıya doğru sıralanmıştır.
- X ekseninde yer alan noktaların her biri hesaplanan Shapley değerleridir. X eksenindeki pozitif değerler, ilgili etiket değeri tahminine olumlu katkı sağlayan değerlerdir. Negatif değerler ise tahmine olumsuz katkı sağlayan değerlerdir.
- X eksenindeki noktaların renkleri ise verinin sayısal değeri hakkında yorum yapabilme yeteneği kazandırmaktadır. Maviden kırmızıya yaklaştıkça verinin sayısal değerinin arttığından bahsedilebilir.

- Kredi skoru 1-8 arasında olan müşterilerin tahmininde en önemli iki özellik Nakit Ödeme Yüzdesi ve Geç Ödeme Oranı'dır. Kredi skoru değeri arttıkça Nakit Ödeme Yüzdesinin de arttığı görülmektedir. Bu aralıktaki müşterilere bakıldığında faturasını geç ödeyenlerin, kredi skorunun daha düşük olduğu sonucu çıkarılabilir.

- Kredi skoru 9 ve 10 olan müşterilere ait grafikler incelendiğinde, Geç Ödeme Oranının en önemli özellik olduğu görülmüştür. En yüksek skorlara sahip kişilerin geç ödeme alışkanlığı olmadığı söylenebilir. Buna ek olarak, bu kişiler faturalarını son ödeme tarihinden önce ödeme eğilimindedir. Ödemelerini ise ağırlıklı olarak nakit yapmaktadırlar.

- Tüm kredi puanlarına ait grafiklere göre, müşterinin ödeme yaptığı farklı fatura türlerinin sayısı, fatura tipi ve müşteri yaşının önemli özellikler arasında yer aldığı gözlemlenmiştir.

LIME ve SHAP çıktıları incelendiğinde aşağıdaki sonuçlar çıkarılabilir:

- Geç ödeme alışkanlığı olan müşterilerin kredi skoru daha düşüktür.

- Kredi skoru yüksek olan müşteriler, faturalarını Nakit olarak ödemeyi tercih etmektedir.

- Erken ödenen gün sayısı arttıkça kredi skoru iyileşmektedir.

- Her özellik farklı etikette farklı önem derecesine sahiptir.

- İlgili ödeme kuruluşunda farklı fatura türlerindeki fatura sayıları önemli bir özellik olarak göze çarpmaktadır.

# 5. TARTIŞMA VE SONUÇ

Finans ve bankacılık sektörünün büyümesi ve gelişmesiyle birlikte finansal risklerin yönetebilmesi de önem kazanmaktadır. Bu risklerin öngörülmesi ve yönetilmesi ile ilgili sık kullanılan yöntemlerden biri kredi skorudur. FICO ve Vantage gibi geleneksel yöntemler dışında farklı skorlama yaklaşımları da ihtiyacı ve arayışları da mevcuttur.

Bu çalışmada, fatura ödeme alışkanlıkları verisi ve makine öğrenmesi yöntemleri kullanılarak, kredi skoru tahmini yapılması konusunda araştırmalar yapılmıştır. Çalışmaya konu olan fatura ödeme verileri, Türkiye'de bulunan bir ödeme kuruluşuna ait gerçek verilerden oluşmaktadır. Ödeme kuruluşundan alınan veriler öncelikle, veri tabanı seviyesinde ön analiz işlemine tabii tutulmuştur. Fatura ödeme hareketleri incelenerek gerekli ilişkisel veri tabanı işlemleri ile kişi bazlı fatura ödeme alışkanlığı veri seti oluşturulmuştur. Fatura ödeme alışkanlıkları bulunan kişilerin gerçek kredi skorları da bu veri seti ile birleştirilmiştir.

Sonuçlar incelendiğinde ön işleme adımlarının performansa olumlu katkılarının olduğu görülmüştür. En iyi performans ölçümleri ise ağaç tabanlı sınıflandırma algoritmaları (RF, EXT) ile edilmiştir.

Çok sınıftan oluştan veri seti için ACC, PRE, REC, F1 skoru, F2 skoru ve AUC gibi farklı metrikler kullanılarak performans ölçümleri yapılmış ve modellerin güvenirlilik açısından da değerlendirilmesi yapılmıştır. Bu metrikler göz önüne alındığında ağaç tabanlı sınıflandırma algoritmalarının en iyi performansa sahip sonuçları ürettiği çıkarımı yapılmıştır.

Makine öğrenmesi modellerinin karmaşıklığı nedeniyle son kullanıcı tarafından açıklanabilir ve yorumlanabilir olmaması sorunu ile ilgi araştırmalar yapılmıştır. XAI yöntemlerinden LIME ve SHAP ile yorumlanabilir ve açıklanabilir çıktılar elde edilmiştir. Bu çıktılar üzerinde incelemeler yapılarak önerilen makine öğrenmesi modeli yorumlanmıştır. Bu çıktıların tutarlı oluşu makine öğrenmesi modelinin güvenirliliği açısından önemlidir.

Önerilen modelin XAI yöntemleri ile çıktıları incelendiğinde, farklı kredi skoruna bağlı olarak özellik önem derecelerinin değiştiği gözlemlenmiştir.

Önerilen model, kredi veya kredi kartı geçmişi baz alınarak hesaplanan geleneksel kredi skoruyla birlikte kullanılarak güvenilirliği arttırabilir. Ek olarak, herhangi bir kredi geçmişi olmayan ve kredi riski değerlendirilmek istenen kişilerin, risk değerlendirmesi konusunda bir yöntem olabilir.

Gelecek çalışmalarda, farklı borç ödeme türlerine ait alışkanlıklar veya alternatif veri setleri ile kredi risk değerlendirmesi tahmini yapılabilir. Bu veri setleri, geleneksel modelle hesaplanmış skorlar veya geçmiş kredi ödemeleri bilgileri ile birleştirilerek zenginleştirilebilir. Farklı makine öğrenmesi yöntemlerinin kullanılması da yeni bir araştırma konusu olabilir. Ayrıca farklı XAI yöntemleri ile açıklanabilirlik ve yorumlanabilirlik ile ilgili araştırmalar için referans olabilir. Ek olarak, dengesiz ve çok sınıflı veri setleriyle çalışma konusunda motivasyon sağlayabilir.

# KAYNAKLAR


[1] L. Thomas, J. Crook, ve D. Edelman, "Chapter 1: The History and Philosophy of Credit Scoring", içinde *Credit Scoring and Its Applications, Second Edition*, içinde Mathematics in Industry. , Society for Industrial and Applied Mathematics, 2017, ss. 1-10. doi: 10.1137/1.9781611974560.ch1.

[2] D. Durand, "Risk Elements in Consumer Instalment Financing, Technical Edition". NBER, 1941. Erişim: 19 Şubat 2024. [Çevrimiçi]. Erişim adresi: https://www.nber.org/books-and-chapters/risk-elements-consumer-instalment-financing-technical-edition

[3] D. E. Kieso, J. J. Weygandt, ve T. D. Warfield, "Intermediate Accounting: IFRS Edition".

[4] F. Louzada, A. Ara, ve G. B. Fernandes, "Classification methods applied to credit scoring: Systematic review and overall comparison", *Surveys in Operations Research and Management Science*, c. 21, sy 2, ss. 117-134, Ara. 2016, doi: 10.1016/j.sorms.2016.10.001.

[5] Y. Xia, C. Liu, Y. Li, ve N. Liu, "A boosted decision tree approach using Bayesian hyper-parameter optimization for credit scoring", *Expert Systems with Applications*, c. 78, ss. 225-241, Tem. 2017, doi: 10.1016/j.eswa.2017.02.017.

[6] G. Zhou, Y. Zhang, ve S. Luo, "P2P Network Lending, Loss Given Default and Credit Risks", *Sustainability*, c. 10, sy 4, Art. sy 4, Nis. 2018, doi: 10.3390/su10041010.

[7] Q. Chen *vd.*, "An Empirical Research on Bank Client Credit Assessments", *Sustainability*, c. 10, sy 5, Art. sy 5, May. 2018, doi: 10.3390/su10051406.

[8] S. Ali *vd.*, "Explainable Artificial Intelligence (XAI): What we know and what is left to attain Trustworthy Artificial Intelligence", *Information Fusion*, c. 99, s. 101805, Kas. 2023, doi: 10.1016/j.inffus.2023.101805.

[9] Ö. Arslan ve M. B. Karan, "Consumer Credit Risk Characteristics: Understanding Income and Expense Differentials", *Emerging Markets Finance & Trade*, c. 46, sy 2, ss. 20-37, 2010.

[10] G. Sabato, "Credit Risk Scoring Models". Rochester, NY, 02 Şubat 2010. doi: 10.2139/ssrn.1546347.

[11] J. N. Crook, D. B. Edelman, ve L. C. Thomas, "Recent developments in consumer credit risk assessment", *European Journal of Operational Research*, c. 183, sy 3, ss. 1447-1465, Ara. 2007, doi: 10.1016/j.ejor.2006.09.100.

[12] S. Arya, C. Eckel, ve C. Wichman, "Anatomy of the credit score", *Journal of Economic Behavior & Organization*, c. 95, ss. 175-185, Kas. 2013, doi: 10.1016/j.jebo.2011.05.005.

[13] R. B. Avery, K. P. Brevoort, ve G. B. Canner, "Credit Scoring and Its Effects on the Availability and Affordability of Credit", *Journal of Consumer Affairs*, c. 43, sy 3, ss. 516-537, 2009, doi: 10.1111/j.1745-6606.2009.01151.x.

[14] L. Cao, "Data Science: A Comprehensive Overview", *ACM Comput. Surv.*, c. 50, sy 3, s. 43:1-43:42, Haziran 2017, doi: 10.1145/3076253.

[15] P. Cunningham, M. Cord, ve S. J. Delany, "Supervised Learning", içinde *Machine Learning Techniques for Multimedia: Case Studies on Organization and Retrieval*, M. Cord ve P. Cunningham, Ed., içinde Bilişsel Teknolojiler. , Berlin, Heidelberg: Springer, 2008, ss. 21-49. doi: 10.1007/978-3-540-75171-7_2.

[16] W. Bao, N. Lianju, ve K. Yue, "Integration of unsupervised and supervised machine learning algorithms for credit risk assessment", *Expert Systems with Applications*, c. 128, ss. 301-315, Ağu. 2019, doi: 10.1016/j.eswa.2019.02.033.
[17] M. Mohammed, M. B. Khan, ve E. B. M. Bashier, *Machine Learning: Algorithms and Applications*. CRC Press, 2016.
[18] L. P. Kaelbling, M. L. Littman, ve A. W. Moore, "Reinforcement Learning: A Survey", *Journal of Artificial Intelligence Research*, c. 4, ss. 237-285, May. 1996, doi: 10.1613/jair.301.
[19] D. Pyle, *Data Preparation for Data Mining*. Morgan Kaufmann, 1999.
[20] "Statistical Analysis with Missing Data, Third Edition | Wiley Series in Probability and Statistics". Erişim: 19 Şubat 2024. [Çevrimiçi]. Erişim adresi: https://onlinelibrary.wiley.com/doi/book/10.1002/9781119482260
[21] X. Zhu ve X. Wu, "Class Noise vs. Attribute Noise: A Quantitative Study", *Artificial Intelligence Review*, c. 22, sy 3, ss. 177-210, Kas. 2004, doi: 10.1007/s10462-004-0751-8.
[22] I. Guyon ve A. Elisseeff, "An introduction to variable and feature selection", *J. Mach. Learn. Res.*, c. 3, sy null, ss. 1157-1182, Mar. 2003.
[23] R. Bro ve A. K. Smilde, "Principal component analysis", *Analytical Methods*, c. 6, sy 9, ss. 2812-2831, 2014, doi: 10.1039/C3AY41907J.
[24] J. A. Olvera-López, J. A. Carrasco-Ochoa, J. F. Martínez-Trinidad, ve J. Kittler, "A review of instance selection methods", *Artif Intell Rev*, c. 34, sy 2, ss. 133-143, Ağu. 2010, doi: 10.1007/s10462-010-9165-y.
[25] B. De Schutter ve T. J. J. van den Boom, "Model predictive control for max-min-plus-scaling systems", içinde *Proceedings of the 2001 American Control Conference. (Cat. No.01CH37148)*, Haz. 2001, ss. 319-324 c.1. doi: 10.1109/ACC.2001.945564.
[26] J. Han, J. Pei, ve H. Tong, *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2022.
[27] S. B. Kotsiantis, "Decision trees: a recent overview", *Artif Intell Rev*, c. 39, sy 4, ss. 261-283, Nis. 2013, doi: 10.1007/s10462-011-9272-4.
[28] L. Breiman, "Bagging predictors", *Mach Learn*, c. 24, sy 2, ss. 123-140, Ağu. 1996, doi: 10.1007/BF00058655.
[29] L. Breiman, "Random Forests", *Machine Learning*, c. 45, sy 1, ss. 5-32, Eki. 2001, doi: 10.1023/A:1010933404324.
[30] S. Lessmann, B. Baesens, H.-V. Seow, ve L. C. Thomas, "Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research", *European Journal of Operational Research*, c. 247, sy 1, ss. 124-136, Kas. 2015, doi: 10.1016/j.ejor.2015.05.030.
[31] D. W. H. Jr, S. Lemeshow, ve R. X. Sturdivant, *Applied Logistic Regression*. John Wiley & Sons, 2013.
[32] J. D. M. Rennie, L. Shih, J. Teevan, ve D. R. Karger, "Tackling the Poor Assumptions of Naive Bayes Text Classifiers", *In Proceedings of the 20th international conference on machine learning (ICML-03)*, ss. 616-623, 2003.
[33] K. Q. Weinberger, J. Blitzer, ve L. Saul, "Distance Metric Learning for Large Margin Nearest Neighbor Classification", içinde *Advances in Neural Information Processing Systems*, MIT Press, 2005. Erişim: 15 Şubat 2024. [Çevrimiçi]. Erişim adresi: https://proceedings.neurips.cc/paper/2005/hash/a7f592cef8b130a6967a90617db5681b-Abstract.html
[34] S. A. Dudani, "The Distance-Weighted k-Nearest-Neighbor Rule", *IEEE Transactions on Systems, Man, and Cybernetics*, c. SMC-6, sy 4, ss. 325-327, Nis. 1976, doi: 10.1109/TSMC.1976.5408784.

[35] C. Cortes ve V. Vapnik, "Support-vector networks", *Mach Learn*, c. 20, sy 3, ss. 273-297, Eyl. 1995, doi: 10.1007/BF00994018.
[36] "An Introduction to Support Vector Machines and Other Kernel-based Learning Methods". Erişim: 15 Şubat 2024. [Çevrimiçi]. Erişim adresi: https://www.cambridge.org/core/books/an-introduction-to-support-vector-machines-and-other-kernelbased-learning-methods/A6A6F4084056A4B23F88648DDBFDD6FC
[37] J. Tang, C. Deng, ve G.-B. Huang, "Extreme Learning Machine for Multilayer Perceptron", *IEEE Transactions on Neural Networks and Learning Systems*, c. 27, sy 4, ss. 809-821, Nis. 2016, doi: 10.1109/TNNLS.2015.2424995.
[38] M.-C. Popescu, V. Balas, L. Perescu-Popescu, ve N. Mastorakis, "Multilayer perceptron and neural networks", *WSEAS Transactions on Circuits and Systems*, c. 8, Tem. 2009.
[39] P. Geurts, D. Ernst, ve L. Wehenkel, "Extremely randomized trees", *Mach Learn*, c. 63, sy 1, ss. 3-42, Nis. 2006, doi: 10.1007/s10994-006-6226-1.
[40] Y. Freund ve R. E. Schapire, "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting", *Journal of Computer and System Sciences*, c. 55, sy 1, ss. 119-139, Ağu. 1997, doi: 10.1006/jcss.1997.1504.
[41] T. Hastie, J. Friedman, ve R. Tibshirani, *The Elements of Statistical Learning*. içinde Springer Series in Statistics. New York, NY: Springer, 2001. doi: 10.1007/978-0-387-21606-5.
[42] T. Hastie, S. Rosset, J. Zhu, ve H. Zou, "Multi-class AdaBoost", *Statistics and Its Interface*, c. 2, sy 3, ss. 349-360, 2009, doi: 10.4310/SII.2009.v2.n3.a8.
[43] J. H. Friedman, "Greedy Function Approximation: A Gradient Boosting Machine", *The Annals of Statistics*, c. 29, sy 5, ss. 1189-1232, 2001.
[44] T. Chen ve C. Guestrin, "XGBoost: A Scalable Tree Boosting System", içinde *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, içinde KDD '16. New York, NY, USA: Association for Computing Machinery, Ağustos 2016, ss. 785-794. doi: 10.1145/2939672.2939785.
[45] G. Ke *vd.*, "LightGBM: A Highly Efficient Gradient Boosting Decision Tree", içinde *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2017. Erişim: 14 Şubat 2024. [Çevrimiçi]. Erişim adresi: https://proceedings.neurips.cc/paper_files/paper/2017/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html
[46] M. Skurichina ve R. P. W. Duin, "Bagging, Boosting and the Random Subspace Method for Linear Classifiers", *Pattern Anal Appl*, c. 5, sy 2, ss. 121-135, Haz. 2002, doi: 10.1007/s100440200011.
[47] G. Louppe ve P. Geurts, "Ensembles on Random Patches", içinde *Machine Learning and Knowledge Discovery in Databases*, c. 7523, P. A. Flach, T. De Bie, ve N. Cristianini, Ed., içinde Lecture Notes in Computer Science, vol. 7523. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, ss. 346-361. doi: 10.1007/978-3-642-33460-3_28.
[48] *Pattern Recognition and Machine Learning*. Erişim: 15 Şubat 2024. [Çevrimiçi]. Erişim adresi: https://link.springer.com/book/9780387310732
[49] G. James ve T. Hastie, "The Error Coding Method and PICTs", *Journal of Computational and Graphical Statistics*, c. 7, sy 3, ss. 377-387, 1998, doi: 10.2307/1390710.
[50] T. G. Dietterich ve G. Bakiri, "Solving Multiclass Learning Problems via Error-Correcting Output Codes", *Journal of Artificial Intelligence Research*, c. 2, ss. 263-286, 1994, doi: 10.1613/jair.105.

[51] A. Fernández, S. García, M. Galar, R. C. Prati, B. Krawczyk, ve F. Herrera, *Learning from Imbalanced Data Sets*. Cham: Springer International Publishing, 2018. doi: 10.1007/978-3-319-98074-4.

[52] H. He ve E. A. Garcia, "Learning from Imbalanced Data", *IEEE Transactions on Knowledge and Data Engineering*, c. 21, sy 9, ss. 1263-1284, Eyl. 2009, doi: 10.1109/TKDE.2008.239.

[53] A. J. Ferreira ve M. A. T. Figueiredo, "Boosting Algorithms: A Review of Methods, Theory, and Applications", içinde *Ensemble Machine Learning: Methods and Applications*, C. Zhang ve Y. Ma, Ed., New York, NY: Springer, 2012, ss. 35-85. doi: 10.1007/978-1-4419-9326-7_2.

[54] B. W. Yap, K. A. Rani, H. A. A. Rahman, S. Fong, Z. Khairudin, ve N. N. Abdullah, "An Application of Oversampling, Undersampling, Bagging and Boosting in Handling Imbalanced Datasets", içinde *Proceedings of the First International Conference on Advanced Data and Information Engineering (DaEng-2013)*, T. Herawan, M. M. Deris, ve J. Abawajy, Ed., içinde Lecture Notes in Electrical Engineering. Singapore: Springer, 2014, ss. 13-22. doi: 10.1007/978-981-4585-18-7_2.

[55] Z. Xu, D. Shen, T. Nie, ve Y. Kou, "A hybrid sampling algorithm combining M-SMOTE and ENN based on Random forest for medical imbalanced data", *Journal of Biomedical Informatics*, c. 107, s. 103465, Tem. 2020, doi: 10.1016/j.jbi.2020.103465.

[56] G. Menardi ve N. Torelli, "Training and assessing classification rules with imbalanced data", *Data Min Knowl Disc*, c. 28, sy 1, ss. 92-122, Oca. 2014, doi: 10.1007/s10618-012-0295-5.

[57] N. V. Chawla, K. W. Bowyer, L. O. Hall, ve W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique", *jair*, c. 16, ss. 321-357, Haz. 2002, doi: 10.1613/jair.953.

[58] H. Han, W.-Y. Wang, ve B.-H. Mao, "Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning", içinde *Advances in Intelligent Computing*, D.-S. Huang, X.-P. Zhang, ve G.-B. Huang, Ed., içinde Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2005, ss. 878-887. doi: 10.1007/11538059_91.

[59] Y. Tang, Y.-Q. Zhang, N. V. Chawla, ve S. Krasser, "SVMs Modeling for Highly Imbalanced Classification", *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, c. 39, sy 1, ss. 281-288, Şub. 2009, doi: 10.1109/TSMCB.2008.2002909.

[60] F. Last, G. Douzas, ve F. Bacao, "Oversampling for Imbalanced Learning Based on K-Means and SMOTE", *Information Sciences*, c. 465, ss. 1-20, Eki. 2018, doi: 10.1016/j.ins.2018.06.056.

[61] H. He, Y. Bai, E. A. Garcia, ve S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning", içinde *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, Haz. 2008, ss. 1322-1328. doi: 10.1109/IJCNN.2008.4633969.

[62] M. Alamri ve M. Ykhlef, "Survey of Credit Card Anomaly and Fraud Detection Using Sampling Techniques", *Electronics*, c. 11, sy 23, Art. sy 23, Oca. 2022, doi: 10.3390/electronics11234003.

[63] Q. Kang, X. Chen, S. Li, ve M. Zhou, "A Noise-Filtered Under-Sampling Scheme for Imbalanced Classification", *IEEE Transactions on Cybernetics*, c. 47, sy 12, ss. 4263-4274, Ara. 2017, doi: 10.1109/TCYB.2016.2606104.

[64] P. M. Rezende, J. S. Xavier, D. B. Ascher, G. R. Fernandes, ve D. E. V. Pires, "Evaluating hierarchical machine learning approaches to classify biological databases", *Briefings in Bioinformatics*, c. 23, sy 4, s. bbac216, Tem. 2022, doi: 10.1093/bib/bbac216.

[65] "Two Modifications of CNN", *IEEE Transactions on Systems, Man, and Cybernetics*, c. SMC-6, sy 11, ss. 769-772, Kas. 1976, doi: 10.1109/TSMC.1976.4309452.
[66] D. L. Wilson, "Asymptotic Properties of Nearest Neighbor Rules Using Edited Data", *IEEE Transactions on Systems, Man, and Cybernetics*, c. SMC-2, sy 3, ss. 408-421, Tem. 1972, doi: 10.1109/TSMC.1972.4309137.
[67] "An Experiment with the Edited Nearest-Neighbor Rule", *IEEE Transactions on Systems, Man, and Cybernetics*, c. SMC-6, sy 6, ss. 448-452, Haz. 1976, doi: 10.1109/TSMC.1976.4309523.
[68] P. Hart, "The condensed nearest neighbor rule (Corresp.)", *IEEE Transactions on Information Theory*, c. 14, sy 3, ss. 515-516, May. 1968, doi: 10.1109/TIT.1968.1054155.
[69] J. Laurikkala, "Improving Identification of Difficult Small Classes by Balancing Class Distribution", içinde *Artificial Intelligence in Medicine*, S. Quaglini, P. Barahona, ve S. Andreassen, Ed., içinde Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2001, ss. 63-66. doi: 10.1007/3-540-48229-6_9.
[70] G. E. A. P. A. Batista, R. C. Prati, ve M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data", *SIGKDD Explor. Newsl.*, c. 6, sy 1, ss. 20-29, Haziran 2004, doi: 10.1145/1007730.1007735.
[71] G. Batista, A. Bazzan, ve M.-C. Monard, "Balancing Training Data for Automated Annotation of Keywords: a Case Study.", program adı: the Proc. Of Workshop on Bioinformatics, Oca. 2003, ss. 10-18.
[72] A. Das ve P. Rad, "Opportunities and Challenges in Explainable Artificial Intelligence (XAI): A Survey". arXiv, 22 Haziran 2020. doi: 10.48550/arXiv.2006.11371.
[73] D. V. Carvalho, E. M. Pereira, ve J. S. Cardoso, "Machine Learning Interpretability: A Survey on Methods and Metrics", *Electronics*, c. 8, sy 8, Art. sy 8, Ağu. 2019, doi: 10.3390/electronics8080832.
[74] G. Elkhawaga, O. Elzeki, M. Abuelkheir, ve M. Reichert, "Evaluating Explainable Artificial Intelligence Methods Based on Feature Elimination: A Functionality-Grounded Approach", *Electronics*, c. 12, sy 7, Art. sy 7, Oca. 2023, doi: 10.3390/electronics12071670.
[75] A. Barredo Arrieta *vd.*, "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI", *Information Fusion*, c. 58, ss. 82-115, Haz. 2020, doi: 10.1016/j.inffus.2019.12.012.
[76] V. Belle ve I. Papantonis, "Principles and Practice of Explainable Machine Learning", *Frontiers in Big Data*, c. 4, 2021, Erişim: 26 Şubat 2024. [Çevrimiçi]. Erişim adresi: https://www.frontiersin.org/articles/10.3389/fdata.2021.688969
[77] M. T. Ribeiro, S. Singh, ve C. Guestrin, "'Why Should I Trust You?': Explaining the Predictions of Any Classifier". arXiv, 09 Ağustos 2016. doi: 10.48550/arXiv.1602.04938.
[78] K. J. Arrow *vd.*, *Contributions to the Theory of Games (AM-28), Volume II*. Princeton University Press, 1953. Erişim: 27 Şubat 2024. [Çevrimiçi]. Erişim adresi: https://www.jstor.org/stable/j.ctt1b9x1zv
[79] S. M. Lundberg ve S.-I. Lee, "A Unified Approach to Interpreting Model Predictions", içinde *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2017. Erişim: 27 Şubat 2024. [Çevrimiçi]. Erişim adresi: https://papers.nips.cc/paper_files/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html
[80] R. Rodríguez-Pérez ve J. Bajorath, "Interpretation of machine learning models using shapley values: application to compound potency and multi-target activity predictions", *J Comput Aided Mol Des*, c. 34, sy 10, ss. 1013-1026, Eki. 2020, doi: 10.1007/s10822-020-00314-0.

[81] T. Fushiki, “Estimation of prediction error by using K-fold cross-validation”, *Stat Comput*, c. 21, sy 2, ss. 137-146, Nis. 2011, doi: 10.1007/s11222-009-9153-8.
[82] C. Sammut ve G. I. Webb, *Encyclopedia of Machine Learning*. Springer Science & Business Media, 2011.
[83] B. Prasetiyo, Alamsyah, M. A. Muslim, ve N. Baroroh, “Evaluation performance recall and F2 score of credit card fraud detection unbalanced dataset using SMOTE oversampling technique”, *J. Phys.: Conf. Ser.*, c. 1918, sy 4, s. 042002, Haz. 2021, doi: 10.1088/1742-6596/1918/4/042002.
[84] D. Chicco ve G. Jurman, “The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation”, *BMC Genomics*, c. 21, sy 1, s. 6, Oca. 2020, doi: 10.1186/s12864-019-6413-7.
[85] P. Baldi, S. Brunak, Y. Chauvin, C. A. F. Andersen, ve H. Nielsen, “Assessing the accuracy of prediction algorithms for classification: an overview”, *Bioinformatics*, c. 16, sy 5, ss. 412-424, May. 2000, doi: 10.1093/bioinformatics/16.5.412.
[86] G. I. Ivchenko ve S. A. Honov, “On the jaccard similarity test”, *J Math Sci*, c. 88, sy 6, ss. 789-794, Mar. 1998, doi: 10.1007/BF02365362.
[87] Q. Nguyen, *Hands-On Application Development with PyCharm: Accelerate your Python applications using practical coding techniques in PyCharm*. Packt Publishing Ltd, 2019.
[88] M. Lutz, *Programming Python*. O’Reilly Media, Inc., 2001.
[89] R. Mistry ve S. Misner, *Introducing Microsoft SQL Server 2014*. Microsoft Press, 2014.

# İNTİHAL RAPORU İLK SAYFASI

**UYARI: Bu bölümde intihal programında çıkan benzerlik yüzdesinin bulunduğu detaylı raporun ilk sayfasının yer alması tüm öğrenciler için zorunludur!** Bu uyarıyı okuduktan sonra siliniz!

# ETİK KURUL İZİN YAZISI

**Uyarı:** Canlı denekler üzerinde yapılan tüm araştırmalar için Etik Kurul Belgesi alınması zorunludur.

☐ **Etik Kurul izni gerekmektedir.**

☒ **Etik Kurul izni gerekmemektedir.**

Cem BULUT

# KURUM İZNİ YAZILARI

**Uyarı:** Canlı ve cansız deneklerle yapılan tüm çalışmalar için kurum izin belgelerinin eklenmesi zorunludur. Gizlilik ve mahremiyet içeren durumlarda kurum adı kapatılmalıdır.

☐ **Kurum izni gerekmektedir.**

☒ **Kurum izni gerekmemektedir.**

Cem BULUT

# ÖZGEÇMİŞ

| Kişisel Bilgiler | |
|---|---|
| Adı Soyadı | Cem Bulut |
| Doğum Tarihi | Tarih girmek için tıklayın veya dokunun. |
| Uyruğu | ☑ T.C. ❒ Diğer: |
| E-Posta Adresi | |
| Web Adresi | |

| Eğitim Bilgileri | |
|---|---|
| **Lisans** | |
| Üniversite | Trakya Üniversitesi |
| Fakülte | Mühendislik Fakültesi |
| Bölümü | Bilgisayar Mühendisliği |
| Mezuniyet Yılı | 2013 |
| **Yüksek Lisans** | |
| Üniversite | Altınbaş Üniversitesi |
| Enstitü Adı | Lisansüstü Eğitim Enstitüsü |
| Anabilim Dalı | Elektrik ve Bilgisayar Mühendisliği |
| Programı | |

| Makale ve Bildiriler |
|---|
| |