

T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

MAKİNE ÖĞRENMESİ YÖNTEMLERİYLE
MÜŞTERİ DAVRANIŞ ANALİZİ

Rabia Gülizar TUNCER

YÜKSEK LİSANS TEZİ
Matematik Mühendisliği Anabilim Dalı
Matematik Mühendisliği Programı

Danışman
Doç. Dr. Serkan ONAR

Temmuz, 2025

T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

MAKİNE ÖĞRENMESİ YÖNTEMLERİYLE MÜŞTERİ
DAVRANIŞ ANALİZİ

Rabia Gülizar TUNCER tarafından hazırlanan tez çalışması 03.07.2025 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Matematik Mühendisliği Anabilim Dalı Matematik Mühendisliği Programı **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Doç. Dr. Serkan ONAR
Yildiz Technical University
Danışman

Jüri Üyeleri

Doç. Dr. Serkan ONAR, Danışman
Yildiz Technical University

Prof. Dr. İbrahim EMİROĞLU, Üye
Yildiz Technical University

Doç. Dr. Mücahit AKBIYIK, Üye
Beykent University

Danışmanım Doç. Dr. Serkan ONAR sorumluluğunda tarafımca hazırlanan Makine Öğrenmesi Yöntemleriyle Müşteri Davranış Analizi başlıklı çalışmada veri toplama ve veri kullanımında gerekli yasal izinleri aldığımı, diğer kaynaklardan aldığım bilgileri ana metin ve referanslarda eksiksiz gösterdiğimi, araştırma verilerine ve sonuçlarına ilişkin çarpıtma ve/veya sahtecilik yapmadığımı, çalışmam süresince bilimsel araştırma ve etik ilkelerine uygun davrandığımı beyan ederim. Beyanımın aksinin ispatı halinde her türlü yasal sonucu kabul ederim.

Rabia Gülizar TUNCER

İmza

TEŞEKKÜR

Tez çalışmamın her aşamasında bana rehberlik eden, değerli bilgi birikimi ve özverili destekleriyle yolumu aydınlatan danışman hocam Doç. Dr. Serkan ONAR'a en içten teşekkürlerimi sunarım. Onun vizyonu ve teşvikleri bu çalışmamın tamamlanmasında kilit rol oynamıştır.

Akademik gelişimime önemli katkılar sağlayan, değerli dersleriyle ve aydınlatıcı yorumlarıyla bana ilham veren Prof. Dr. Kevser ÖZDEN KÖKLÜ hocama özellikle teşekkür ederim.

Derslerinden istifade ettiğim okulumuzdaki kıymetli hocalarıma da teşekkürü borç bilirim. Tez jürimde yer alarak kıymetli zamanlarını ayıran, yapıcı eleştirileri ve değerli önerileriyle çalışmama katkı sağlayan tüm hocalarıma şükranlarımı sunarım.

Bu süreçte beni her daim destekleyen, sızlanmalarımı dinleyen, moralimi yüksek tutan sevgili aileme ve sevgili dostum Beyza Nur Sezgin'e, yol gösterici fikirler sunan iş arkadaşım Ferhat Aygün'e ve tüm dostlarıma sonsuz teşekkürler.

Ve tabii ki, geceleri uyanık kalmamı, sabahları hayata dönmemi sağlayan, her zor anımda yanımda olan, sadık çalışma arkadaşım kahve makinama da özel bir teşekkür borçluyum.

Rabia Gülizar TUNCER

Rabia Gülizar TUNCER

İÇİNDEKİLER

SİMGE LİSTESİ	vii
KISALTMA LİSTESİ	ix
ŞEKİL LİSTESİ	xi
TABLO LİSTESİ	xiii
ÖZET	xiv
ABSTRACT	xvi
1 GİRİŞ	1
1.1 Giriş	1
1.1.1 Çalışmanın Amacı ve Önemi	1
1.1.2 Çalışmanın Kapsamı	2
1.1.3 Tezin Yapısı	3
2 LİTERATÜR TARAMASI	4
2.1 Problemin Temelleri ve Veri Yönetimi	4
2.2 Tahminsel Modelleme Yaklaşımları	5
2.2.1 Sınıflandırma ve Regresyon Modelleri	5
2.2.2 Dengesiz Veri Seti Yönetiminde Modelleme	8
2.3 Zaman Serisi ve Davranışsal Akış Analizi	9
2.3.1 Zaman Serisi Tahmin Modelleri	9
2.3.2 Davranışsal Akış Analizi (Markov Zincirleri)	11
2.4 Model Yorumlanabilirliği (Explainable AI)	11
2.5 E-ticaret ve Müşteri Davranışları Alanındaki Uygulama Çalışmaları	11
2.6 Metodoloji	13
2.6.1 Veri Seti	13
2.6.2 Veri Ön İşleme ve Özellik Mühendisliği	14
3 METODOLOJİ	18
3.1 Metodoloji	18

3.1.1	Veri Seti	18
3.1.2	Veri Ön İşleme ve Özellik Mühendisliği	20
3.1.3	Uygulanan Analiz Yöntemleri	22
4	BULGULAR	39
4.1	Bulgular	39
4.1.1	Sınıflandırma Modelleri ve Dengesiz Veri Seti	39
4.1.2	Sınıflandırma Modelleri ve Açıklanabilirlik Bulguları	40
5	SONUÇ	64
5.1	Ana Bulguların Özeti	64
5.1.1	Sınıflandırma Modelleri ve Satın Alma Tahmini	64
5.1.2	Zaman Serisi ve Davranışsal Akış Analizi	65
5.1.3	Segmentasyon ve Öneri Sistemleri	65
5.1.4	İş Zekası ve Eyleme Dönüştürülebilir İçgörüler	66
5.2	Çalışmanın Katkıları	66
5.3	Çalışmanın Sınırlılıkları ve Gelecek Çalışmalar	66
	KAYNAKÇA	68
	TEZDEN ÜRETİLMİŞ YAYINLAR	72

SİMGE LİSTESİ

d	ARIMA modelinde Entegre (I) bileşenin (fark alma) derecesi
p	ARIMA modelinde Otoregresif (AR) bileşenin derecesi
q	ARIMA modelinde Hareketli Ortalama (MA) bileşenin derecesi
n_{min}	Azınlık sınıfındaki örnek sayısı
$P(C D)$	Bir veri kümesi (D) verildiğinde sınıfın (C) koşullu olasılık dağılımı
$\prod$	Çarpım (Produkt) operatörü
IQR	Çeyrekler Arası Aralık (InterQuartile Range)
w_t	Durağan hale getirilmiş zaman serisi
Δ	Fark alma operatörü (differencing)
γ	Holt-Winters modelinde mevsimsellik düzeltme parametresi
α	Holt-Winters modelinde seviye düzeltme parametresi
β	Holt-Winters modelinde trend düzeltme parametresi
argmax	Maksimum değeri döndüren argüman
k	K-Means algoritmasında optimum küme sayısı; K-NN algoritmasında komşu sayısı
β_0	Lojistik Regresyon modelinde kesme terimi (intercept)
β_1	Lojistik Regresyon modelinde tahminci katsayısı
X	Lojistik Regresyon modelinde tahminci değişken
μ_{jj}	Markov Zincirinde j durumunun yinelenme süresi
P_{ij}	Markov Zincirinde i durumundan j durumuna geçiş olasılığı
π_i	Markov Zincirinde i durumunda olma kararlı durum olasılığı
$\epsilon(t)$	Prophet modelinde hata terimi
$g(t)$	Prophet modelinde trend bileşeni

$h(t)$	Prophet modelinde tatil bileşeni
$s(t)$	Prophet modelinde mevsimsellik bileşeni
x	SMOTE algoritmasında orijinal azınlık sınıfı örneği
x^R	SMOTE algoritmasında rastgele seçilen en yakın komşu
u	SMOTE algoritmasında $\mathcal{U}(0, 1)$ tekdüze dağılımından rastgele seçilen değer
S	SMOTE ile türetilen yeni sentetik örnek
<code>loss='mse'</code>	Sinir ağı eğitiminde kullanılan kayıp fonksiyonu (Ortalama Kare Hata)
<code>int</code>	Tamsayı veri tipi
Y_t	Zaman serisi modellemesinde tahmin edilen değişkenin değeri
Y_{t-1}, Y_{t-2}	Zaman serisinde önceki gözlem değerleri
e_t	Zaman serisi modellemesinde t anındaki hata terimi
e_{t-1}, e_{t-2}	Zaman serisi modellemesinde önceki hata terimleri

KISALTMA LİSTESİ

ARIMA	AutoRegressive Integrated Moving Average
AUC	Area Under the Curve
CV	Cross-Validation
XGBoost	extreme Gradient Boosting
FastAPI	Fast API (Web Framework)
FN	False Negative
FP	False Positive
IQR	InterQuartile Range
K-Means	K-Means Clustering Algorithm
KNN	K-Nearest Neighbors
KPI	Key Performance Indicator
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MSE	Mean Squared Error
NB	Naive Bayes
OOB	Out-Of-Bag
OR	Odds Ratio
PR	Precision-Recall
ReLU	Rectified Linear Unit
RF	Random Forest
RMSE	Root Mean Squared Error
ROC	Receiver Operating Characteristic

SHAP	Shapley Additive explanations
SMOTE	Synthetic Minority Over-sampling Technique
SVM	Support Vector Machine
TN	True Negative
TP	True Positive
WCSS	Within-Cluster Sum of Squares



ŞEKİL LİSTESİ

Şekil 4.1	SMOTE uygulaması öncesi ve sonrası veri setindeki sınıf dağılımları. Azınlık sınıfı olan "transaction" örneklerinin çoğaltılarak sınıflar arası dengesizliğin giderildiği görülmektedir. .	39
Şekil 4.2	Model Karşılaştırma Grafiği	40
Şekil 4.3	Model Karşılaştırma Grafiği	41
Şekil 4.4	Sınıflandırma Modellerinin SMOTE Sonrası karmaşıklık matrisleri	43
Şekil 4.5	Sınıflandırma modellerinin ROC Eğrileri.	44
Şekil 4.6	XGBoost Modeli SHAP Grafiği	45
Şekil 4.7	XGBoost modeli için SHAP özellik önem sıralaması	45
Şekil 4.8	Bireysel Zaman Serisi Modellerinin Günlük İşlem Sayısı Tahminleri	46
Şekil 4.9	Zaman Serisi Modellerinin Performans Metrikleri	47
Şekil 4.10	LSTM Olay Dizisi Tahminleme Öğrenme Eğrisi	48
Şekil 4.11	LSTM Olay Dizisi Tahminleme Karışıklık Matrisi	49
Şekil 4.12	LSTM Modelinin Günlük İşlem Sayısı Tahminleri	50
Şekil 4.13	En Geliştirilmiş LSTM Modelinin Günlük İşlem Sayısı Tahminleri	50
Şekil 4.14	Geliştirilmiş LSTM Model Öğrenme Eğrisi	51
Şekil 4.15	Zaman Serisi Tahmin Modellerinin Günlük İşlem Sayısı Tahminleri Karşılaştırması.	52
Şekil 4.16	Markov Zinciri Olay Geçiş Frekansları	53
Şekil 4.17	Markov Zinciri Olay Geçiş Olasılıkları Diyagramı	54
Şekil 4.18	KMeans Algoritması İçin Optimum Küme Sayısı Belirlemede Dirsek Metodu	55
Şekil 4.19	KMeans Algoritması İçin Silhouette Skoru Değerleri	56
Şekil 4.20	Müşteri Segmentlerinin Ortalama Etkileşim Metrikleri (Ölçeklendirilmiş)	57
Şekil 4.21	Müşteri Segmentlerinin Ortalama Alışveriş Yapma Oranları	58
Şekil 4.22	Müşteri Segmentlerinin Karşılaştırmalı Davranış Profilleri	59
Şekil 4.23	Haftanın Günlerine Göre Günlük İşlem Sayısı Dağılımı	59
Şekil 4.24	Prophet Modelinin Trend, Haftalık, Yıllık ve Günlük Bileşenleri .	60
Şekil 4.25	Apriori Algoritması ile Elde Edilen En Güçlü 10 Ürün İlişkisi (Lift Değerine Göre)	62

Şekil 4.26 Apriori Algoritması ile Çıkarılan Kuralların Destek, Güven ve Kaldıraç İlişkisi	63
---	----



TABLO LİSTESİ

Tablo 4.1	Modellerin SMOTE Öncesi ve Sonrası Performans Karşılaştırması	42
------------------	---	----



Makine Öğrenmesi Yöntemleriyle Müşteri Davranış Analizi

Rabia Gülizar TUNCER

Matematik Mühendisliği Anabilim Dalı
Yüksek Lisans Tezi

Danışman: Doç. Dr. Serkan ONAR

Bu çalışma, e-ticaret platformlarında müşteri davranışlarını daha iyi anlamak, tahmin etmek ve bu verileri aksiyona dönüştürebilmek için makine öğrenmesi ile zaman serisi analizlerinin bir arada kullanıldığı kapsamlı bir yaklaşım sunmaktadır. RetailRocket veri seti üzerinden yürütülen çalışmada, kullanıcı geçmişi temel alınarak üç aşamalı bir analiz süreci izlenmiştir.

İlk aşamada, kullanıcıların etkileşimlerinden davranışsal özellikler çıkarılmış ve veri dengesizliği SMOTE yöntemiyle giderilmiştir. Ardından, alışveriş yapma olasılıklarını tahmin etmek için Random Forest, Lojistik Regresyon, Naive Bayes ve XGBoost gibi farklı sınıflandırma algoritmaları uygulanmıştır. Modellerin başarımı F1 skoru ve ROC AUC gibi metriklerle ölçülürken, açıklanabilir yapay zeka (XAI) yöntemleri kullanılarak model kararları daha anlaşılır hale getirilmiştir.

İkinci aşamada, kullanıcı davranışlarının zaman içindeki değişimi Prophet, ARIMA, Holt-Winters ve LSTM gibi zaman serisi modelleriyle incelenmiştir. Aynı zamanda, kullanıcı eylemlerinin sırasal ilişkilerini analiz edebilmek için Markov zinciri modelleriyle olasılık tabanlı geçiş yapıları oluşturulmuştur.

Son aşamada ise kullanıcı segmentasyonu KMeans algoritmasıyla gerçekleştirilmiş, her segmentin davranışsal özetleri çıkarılmış ve Apriori algoritması yardımıyla ürün öneri kuralları elde edilmiştir. Tüm analiz çıktıları, FastAPI kullanılarak geliştirilen bir servis aracılığıyla gerçek zamanlı tahmin ve öneri sunacak şekilde entegre edilmiştir.

Genel olarak bu çalışma, müşteri davranışlarını çok yönlü analiz ederek anlamlı içgörüler üreten ve bu bilgileri karar destek sistemlerine entegre eden bütüncül bir veri bilimi çözümü ortaya koymaktadır.

Anahtar Kelimeler: Makine öğrenmesi, müşteri davranış analizi



ABSTRACT

Customer Behavior Analysis With Machine Learning Methods

Rabia Gülizar TUNCER

Department of Mathematical Engineering
Master of Science Thesis

Supervisor: Assoc. Prof. Serkan ONAR

This study presents a comprehensive data science approach that integrates machine learning and time series analysis to understand, predict, and act upon customer behavior on e-commerce platforms. Using the RetailRocket dataset, the project follows a three-phase methodology based on users' past interactions.

In the first phase, behavioral features were extracted for each user, and the issue of data imbalance was addressed using the SMOTE technique. Various classification algorithms—such as Random Forest, Logistic Regression, Naive Bayes, and XGBoost—were employed to predict whether a user would make a purchase. Model performances were evaluated using metrics like F1 score and ROC AUC, and explainable AI techniques were applied to interpret model outputs.

In the second phase, temporal patterns in user actions were analyzed using time series models like Prophet, ARIMA, Holt-Winters, and LSTM. Additionally, Markov chain models were constructed from event log data to capture the probabilistic transition structure between user interactions.

The final phase focused on user segmentation via the KMeans algorithm. Segment-specific behavioral statistics were extracted, and product recommendation rules were generated using the Apriori algorithm. All analyses were deployed in a real-time recommendation and prediction service built with FastAPI.

In summary, this study delivers a holistic data science framework that not only analyzes customer behavior from multiple dimensions but also transforms these

insights into actionable decision support systems.

Keywords: Machine learning, customer behavior analysis



1.1 Giriş

E-ticaret sektörünün küresel ölçekte büyümesiyle birlikte, şirketler benzeri görülmemiş miktarda müşteri etkileşim verisi toplamaktadır. Müşterilerin bir ürünü görüntülemesinden (*view*), sepete eklemesine (*addtocart*) ve son olarak satın alma (*transaction*) eylemini gerçekleştirmesine kadar uzanan dijital yolculuğu, işletmeler için değerli bilgiler barındıran karmaşık bir süreçtir. Bu sürecin doğru bir şekilde analiz edilmesi, müşteri davranışlarını anlamak, kişiselleştirilmiş deneyimler sunmak ve sonuç olarak gelirleri artırmak için hayati öneme sahiptir.

Bu tez çalışması, RetailRocket açık kaynak veri setindeki olay kayıtları (*event log*) verisi üzerinde bu problemi ele almaktadır. Karşılaşılan en temel ve kritik zorluk, veri setindeki aşırı dengesiz sınıf dağılımıdır. Veri setinde satın alma eylemini gerçekleştiren kullanıcıların oranı yalnızca %0.83 iken, satın alma yapmayanların oranı %99.17'dir. Bu durum, standart makine öğrenmesi modellerinin, nadir olan "*satın alma*" olayını tespit etmede yetersiz kalmasına ve ezici çoğunluğa sahip olan "*satın almama*" sınıfını tahmin etmeye eğilimli olmasına neden olur. Dolayısıyla, bu dengesizliği etkin bir şekilde yönetmek, çalışmanın temel metodolojik zorluklarından birini oluşturmaktadır.

1.1.1 Çalışmanın Amacı ve Önemi

Bu tezin temel amacı, e-ticaret müşteri davranışlarını, tek bir modelin sınırlarının ötesine geçen çok boyutlu ve bütüncül bir veri bilimi yaklaşımıyla analiz etmektir. Bu ana amaç doğrultusunda çalışma, dört temel hedef üzerine inşa edilmiştir:

- **Tahminleme:** Bu aşamada, aşırı dengesiz veri koşulları altında, bir kullanıcının satın alma yapıp yapmayacağını yüksek F1 skoru ile tahmin eden sınıflandırma modelleri (Random Forest, XGBoost vb.) geliştirmek ve bu modellerin karar mekanizmalarını SHAP ile açıklamak.

- **Zaman Serisi ve Davranışsal Akış Analizi:** Toplam işlem sayısını zaman serisi modelleri (ARIMA, LSTM vb.) ile tahminleyerek gelecekteki satış trendlerini öngörmek ve Markov Zincirleri ile kullanıcıların *görün-tüleme*, *sepete ekleme* ve *satın alma* eylemleri arasındaki geçiş olasılıklarını modellemek.
- **Segmentasyon ve Kişiselleştirme:** Müşterileri davranışsal özelliklerine göre kümeler (segmentlere) ayırmak için KMeans algoritmasını kullanmak ve Apriori algoritması ile her bir segmente özel ürün tavsiye kuralları çıkarmak.
- **Eyleme Dönüştürülebilir İçgörüler:** Elde edilen tüm teknik bulguları, karar vericiler için anlamlı KPI'lar (dönüşüm oranı, sepet terk oranı vb.) içeren bir dashboard çatısı altında birleştirmek ve *Veri* → *İçgörü* → *Karar* → *Aksiyon* akışını somutlaştırmak.

Çalışmanın önemi, e-ticaret verisine yönelik izole analizler yapmak yerine; tahminleme, davranışsal modelleme, segmentasyon ve stratejik karar desteğini bir araya getiren entegre bir çerçeve sunmasından kaynaklanmaktadır.

1.1.2 Çalışmanın Kapsamı

Bu çalışma, RetailRocket veri seti kullanılarak gerçekleştirilmiştir. Araştırmanın kapsamı, yukarıda belirtilen dört ana hedef etrafında şekillendirilmiş ve aşağıdaki ana teknik ve analizleri içermektedir:

- **Sınıflandırma:** SMOTE ile veri dengeleme, Random Forest, XGBoost, Lojistik Regresyon ve Naive Bayes modellerinin uygulanması ve SHAP ile model yorumlanabilirliği.
- **Zaman Serisi ve Davranış Analizi:** Prophet, ARIMA, Holt-Winters ve LSTM modelleri ile tahminleme; Markov Geçiş Matrisleri ile davranış akışı analizi.
- **Segmentasyon ve Öneri Sistemleri:** KMeans algoritması ile müşteri segmentasyonu ve Apriori algoritması ile birliktelik kurallarına dayalı ürün önerileri.
- **İş Zekası:** Temel performans göstergelerini (KPI) içeren bir dashboard konsepti ve stratejik karar akış şeması.

1.1.3 Tezin Yapısı

Bu tez çalışması beş bölümden oluşmaktadır. Giriş bölümünün ardından, ikinci bölümde, çalışmanın temelini oluşturan konularla ilgili literatür taraması sunulmuştur. Üçüncü bölümde, kullanılan veri seti, uygulanan metodolojiler ve analiz adımları detaylı olarak açıklanmıştır. Dördüncü bölümde, her bir analizden elde edilen bulgular, grafikler ve tablolar eşliğinde sunulmuş ve yorumlanmıştır. Beşinci ve son bölümde ise çalışmanın genel sonuçları özetlenmiş, pratik çıkarımlar vurgulanmış ve gelecekteki araştırmalar için önerilerde bulunulmuştur.



2 LİTERATÜR TARAMASI

Bu bölümde, e-ticaret müşteri davranışlarını anlama, tahmin etme ve optimize etme amacıyla kullanılan makine öğrenmesi ve veri bilimi teknikleri ile ilgili literatür incelenmiştir. Çalışmanın temelini oluşturan metodolojiler ve bunların e-ticaret alanındaki uygulamaları, ilgili araştırmalar ışığında değerlendirilerek, mevcut çalışmanın literatürdeki yeri belirlenmiştir.

E-ticaret sektörünün küresel ölçekte büyümesiyle birlikte, şirketler benzeri görülmemiş miktarda müşteri etkileşim verisi toplamakta ve bu durum, satış tahminleme süreçlerinin karmaşıklığını artırmaktadır [1]. Müşteri çeşitliliği ve bunun yönetilmesi, perakendeciler için hem stratejik planlama hem de operasyonel düzeyde önemli bir güçlük oluşturmaktadır [1]. Bu bağlamda, müşteri davranışlarını anlamak, kişiselleştirilmiş deneyimler sunmak ve gelirleri artırmak hayati öneme sahiptir [1]. Makine öğrenmesi ve veri madenciliği (DM) teknikleri, müşteri verilerindeki "gizli" bilgileri keşfetmek ve müşteri ilişkileri yönetimini (CRM) geliştirmek için güçlü araçlar sunmaktadır [2].

2.1 Problemin Temelleri ve Veri Yönetimi

E-ticaret verileri genellikle, görüntüleme ('view') ve sepete ekleme ('addtocart') gibi eylemlerin satın alma ('transaction') eylemlerine kıyasla çok daha fazla olduğu, aşırı derecede dengesiz sınıf dağılımlarına sahiptir [1]. Bu durum, standart makine öğrenmesi modellerinin nadir görülen "satın alma" gibi olayları tespit etmede yetersiz kalmasına ve çoğunluk sınıfını tahmin etmeye eğilimli olmasına neden olur [1, 3]. Sınıf dengesizliği problemi, özellikle değişken sayısının örnek sayısından çok daha fazla olduğu yüksek boyutlu verilerde daha da şiddetlenir ve modelin yanlış tahminler üretmesine yol açabilir [3]. Bu tür sorunları gidermek için veri ön işleme, özellik mühendisliği ve veri dengeleme teknikleri hayati önem taşımaktadır [1].

2.2 Tahminsel Modelleme Yaklaşımları

E-ticaret ortamında müşteri davranışlarını tahmin etmek için çeşitli makine öğrenmesi modelleri kullanılmaktadır. Bu modeller, müşterinin satın alma yapıp yapmayacağı gibi ikili sınıflandırma problemlerinden, satış tutarlarını veya gelecekteki talepleri tahmin etmeye yönelik regresyon problemlerine kadar geniş bir yelpazeyi kapsar.

2.2.1 Sınıflandırma ve Regresyon Modelleri

Lojistik Regresyon: İkili sonuçlara sahip çıktıları analiz etmek için kullanılan temel bir regresyon modelidir. Doğrusal regresyonun aksine, tahminleri 0 ile 1 arasında kalır ve değişken varyans varsayımına takılmaz [4]. Lojistik regresyon, sürekli veya kategorik tahmincileri kullanmaya ve çoklu tahminciler için ayarlama yapmaya izin verir, bu da gözlemsel verilerdeki potansiyel yanlılığı azaltmak için faydalıdır [4]. Sonuçlar tipik olarak olasılık oranları (Odds Ratios - OR) olarak sunulur ve tahmincilerdeki değişimin olayın olasılığı üzerindeki etkisini belirtir [4]. Sahar F. Sabbeh (2018) tarafından yapılan bu çalışma, müşteri ilişkileri yönetiminde (CRM) müşteri kaybı (churn) tahminine yönelik makine öğrenmesi tekniklerinin karşılaştırmalı bir analizini sunmaktadır. Makale, telekomünikasyon sektörüne ait bir veri setini (3333 kayıt) kullanarak On farklı sınıflandırma algoritmasının performansını değerlendirmiştir. İncelenen algoritmalar arasında Karar Ağaçları (CART), K-En Yakın Komşular (k-NN), Destek Vektör Makineleri (SVM), Lojistik Regresyon, Naive Bayes, Çok Katmanlı Algılayıcı (MLP) ve topluluk öğrenme teknikleri (Random Forest, AdaBoost, Stokastik Gradyan Yükseltme) bulunmaktadır. Çalışmanın bulgularına göre, Random Forest ve AdaBoost modelleri, yaklaşık %96 doğrulukla diğer tüm tekniklerden daha iyi performans sergilemiştir. MLP ve SVM de yaklaşık %94 doğrulukla güçlü modeller olarak öne çıkmıştır. Çalışma, özellikle topluluk tabanlı öğrenme tekniklerinin müşteri kaybı tahmininde yüksek performans sunduğunu vurgulamaktadır.[5] Pelegrín et al. (2023), e-ticaret ortamlarında tüketici davranışlarını tahmin etmeye yönelik olarak çok modellemeli bir öneri sistemi geliştirmiştir. Çalışmada, kullanıcıların göz atma ve tıklama verileri dikkate alınarak zaman serisi analizi, öneri algoritmaları ve makine öğrenmesi yöntemleri birlikte uygulanmıştır. Özellikle hibrit sistemlerin, kullanıcı deneyimini iyileştirmede ve kişiselleştirilmiş öneriler sunmada etkili olduğu ortaya konmuştur.[6] Patel et al. (2020), müşteri segmentasyonu için çeşitli makine öğrenmesi algoritmalarını karşılaştırmalı olarak inceleyerek özellikle K-means kümeleme algoritmasının segment bazlı pazarlama stratejileri geliştirmede etkili olduğunu göstermiştir.[7] Bhagat et al.(2023) makalesi, müşteri davranışlarının anlaşılması ve segmentasyonuna yönelik olarak veri madenciliği ve

makine öğrenmesi tekniklerinin perakende sektörü için nasıl stratejik avantajlar sağlayabileceğini ortaya koymaktadır. Özellikle müşteri bağlılığı ve yaşam boyu değer analizleri, karar destek sistemleri açısından ön plana çıkarılmıştır.[8] Da Xu et al. (2012) çalışmasında, farklı veri türlerinin (zaman serileri, sembolik sıralar, metinsel veriler vb.) ayrı sistemlerde tutulduğu ortamlarda müşteri davranışlarının tahminine yönelik merkezi olmayan bir çözüm önerilmiştir. Geliştirilen C-MK-SVM yöntemi, dağınık veri yapılarında daha yüksek doğruluk ve hesaplama hızıyla müşteri davranışı tahmininde üstün performans sergilemiştir. [9] Tsai and Chiu (2005), müşteri davranışındaki değişimleri belirlemek amacıyla RFM analizini, demografik verileri ve işlem verilerini entegre eden bir veri madenciliği yöntemi önermiştir. Geliştirilen benzerlik ve beklenmedik değişim ölçütleriyle zaman içindeki davranışsal değişim kalıpları yakalanmış, böylece pazarlama stratejilerinin dinamik piyasa koşullarına göre optimize edilmesi sağlanmıştır. [10] Gürkan (2023) çalışmasında, gıda toptancılığı sektöründe müşteri kaybını analiz etmek amacıyla makine öğrenmesi teknikleri kullanılmıştır. Çalışmada SMOTE yöntemiyle veri dengelenmiş ve XGBoost algoritması ile %91.3 doğruluk oranına ulaşılmıştır. Bu yaklaşım, müşteri kaybının önceden tahmin edilmesi ve pazarlama stratejilerinin iyileştirilmesi açısından önemli bir katkı sunmaktadır. [11] Yağız Yaman'ın çalışmasında, sentetik bir veri kümesiyle desteklenen müşteri yolculuğu analitiği, CRISP-DM metodolojisi kapsamında modellenmiştir. Müşteri özellikleri ve aktiviteler üzerinden Random Forest gibi makine öğrenmesi algoritmalarıyla kurulan modeller sayesinde, yeni veya mevcut müşterilerin gelecekteki davranışları yüksek doğrulukla tahmin edilmiştir. Bu sayede firmalara, kayıp riskindeki müşterilere müdahale etme ve kârlılığı artırma yönünde stratejik avantajlar sunulmuştur. [12] Kaplan ve Uçkan (2020), Türkiye perakende sektöründeki müşteri ilişkileri yönetimini (CRM) inceleyerek dijitalleşmenin müşteri deneyimi üzerindeki etkisini değerlendirmiştir. Araştırma sonucunda, dijital araçların etkin kullanımının müşteri sadakatini artırdığı, CRM stratejilerinin ise veri odaklı yaklaşımlarla yeniden şekillendiği ortaya konmuştur. [13] Kurt (2021), müşteri ilişkileri yönetiminde veri madenciliği ve makine öğrenmesi tekniklerinin rolünü incelemiştir. Çalışmada, veri tabanlı yaklaşımlar kullanılarak müşteri sadakati, segmentasyonu ve kişiselleştirilmiş öneri sistemleri gibi alanlarda nasıl stratejik avantaj sağlandığı detaylandırılmıştır. [14]

Naive Bayes Sınıflandırıcısı: Olasılıksal bir makine öğrenmesi tekniği olup, Bayes teoremini temel alarak sınıflandırma yapar ve özelliklerin sınıf verildiğinde koşullu olarak bağımsız olduğunu varsayar [15]. Bu "naive" varsayım, pratikte özellikler arasında etkileşimler olsa bile Naive Bayes'in iyi sonuçlar verebilmesini sağlar; zira algoritma sadece en olası sınıfı doğru bir şekilde sıralamakla ilgilenir [15].

Flach ve Lachiche (2004) tarafından yapılan çalışma, geleneksel Naive Bayes'in yapılandırılmış verilerdeki sınırlamalarını aşmak için 1BC ve 1BC2 sistemlerini önermiştir.

Random Forest: L. Breiman tarafından 2001'de önerilen Random Forest algoritması, çok sayıda rastgele oluşturulmuş karar ağacını bir araya getirerek ve tahminlerini birleştirerek (regresyon için ortalama, sınıflandırma için çoğunluk oyu) çalışan güçlü bir topluluk öğrenmesi modelidir [16]. Özellikle değişken sayısının gözlem sayısından çok daha fazla olduğu yüksek boyutlu veri setlerinde mükemmel performans sergiler [16]. Yüksek doğruluğu, aşırı öğrenmeye karşı direnci ve özellik önem derecesi sağlama yeteneği ile öne çıkar [16]. Bu çalışma, sınıflandırma doğruluğunu artırırken kullanılan ağaç sayısını minimize eden, geliştirilmiş bir Random Forest algoritması önermektedir. Önerilen yöntem, önemsiz öznelilikleri iteratif olarak eleyerek sadece önemli özneliliklerle sınıflandırma yapar ve bu sayede hem daha az ağaçla daha yüksek doğruluk elde edilir hem de modelin genelleme kapasitesi artırılır. [17] Liaw ve Wiener (2002), "random forest" algoritmasının sınıflandırma ve regresyon görevlerinde nasıl etkili şekilde kullanılabileceğini açıklamaktadır. Makale, rastgele değişken alt kümeleri kullanılarak karar ağaçlarının nasıl geliştirileceğini ve bu yaklaşımın overfitting'e karşı dayanıklı olduğunu vurgular. Ayrıca, değişken önem derecelendirmeleri ve veri içi benzerlik ölçümleri gibi ek çıktılar sunan R paketine dair uygulama örnekleri de içermektedir. Bu yöntem, özellikle değişken sayısı fazla olan veri setlerinde güçlü performans göstermektedir. [18] Verbeke et al. (2007), finans sektöründe 100.000 müşterilik bir veri seti üzerinde rasgele ormanlar (random forests) ve regresyon ormanları tekniklerini kullanarak müşteri davranışlarının – özellikle yeniden satın alma, kısmi müşteri kaybı ve kârlılık evrimi – öngörüsünü araştırmaktadır. Sonuçlar, bu tekniklerin lojistik ve doğrusal regresyona göre daha başarılı tahmin gücü sunduğunu ve müşteri davranışlarında farklı değişkenlerin etkili olduğunu göstermektedir. [19] Verbeke et al. (2007), müşteri ilişkileri yönetimi kapsamında müşteri davranışlarının üç önemli çıktısını – yeniden satın alma (next buy), kısmi müşteri kaybı (partial-defection) ve kârlılık evrimi – analiz etmek için rastgele orman (random forest) tekniklerini kullanmıştır. 100.000 müşteriden oluşan gerçek bir finansal hizmet veri setiyle yapılan çalışmada, rastgele ormanlar hem sınıflandırma hem de regresyon bağlamında değerlendirilmiştir. Bulgular, bu tekniklerin doğrusal ve lojistik regresyona göre daha başarılı olduğunu ve aynı değişkenlerin farklı müşteri davranışları üzerinde farklı etkiler yarattığını göstermektedir. [20]

XGBoost: Gradyan yükseltme (Gradient Boosting) çerçevesinin optimize edilmiş bir uygulamasıdır. Ramraj S. ve arkadaşları (2016), XGBoost'un geleneksel

gradyan yükseltme algoritmalarına kıyasla çoklu iş parçacığı (multi-threaded) yaklaşımı ve veri organizasyonu optimizasyonları sayesinde önemli hız ve performans iyileştirmeleri sunduğunu belirtmektedir [21]. Seyrek verilere duyarlılığı, eksik değerleri otomatik olarak işleyebilmesi ve ağaç yapımını paralelleştirmeyi destekleyen blok yapısı gibi özellikleriyle öne çıkar [21]. Çalışmalar, XGBoost'un yapılandırılmış veya tablosal veri setlerinde sınıflandırma, regresyon ve tahminsel modelleme problemlerinde üstün performans gösterdiğini ortaya koymuştur [21]. Bu çalışma, eksik verilerle başa çıkmada XGBoost algoritmasının performansını analiz etmektedir. Araştırma, XGBoost'un eksik verileri etkili bir şekilde işleyebildiğini ve sınıflandırma doğruluğunda önemli iyileşmeler sağladığını göstermektedir. Modelin eksik veri içeren senaryolarda diğer bazı yöntemlerden daha üstün olduğu vurgulanmaktadır. [22] Bu makalede, ağ güvenliği için bir Saldırı Tespit Sistemi (IDS) oluşturmak amacıyla XGBoost algoritması kullanılmıştır. NSL-KDD veri seti üzerinde yapılan testler, XGBoost'un yüksek doğruluk oranı ve işlem hızıyla etkili bir çözüm sunduğunu göstermektedir. Bu model, güvenli ağ ortamlarının oluşturulmasında önemli katkılar sağlamaktadır. [23] Bu çalışma, müşteri davranışlarının tahmininde XGBoost algoritmasının kullanımıyla ilgili yeni yöntemler sunmaktadır. Büyük veri kümeleriyle yapılan deneyler, XGBoost'un müşteri analizinde yüksek doğruluk ve genellenebilirlik sağladığını ortaya koymuştur. Bu da perakende sektörü için güçlü veri odaklı karar destek sistemleri geliştirilmesine olanak tanımaktadır. [24]

2.2.2 Dengesiz Veri Seti Yönetiminde Modelleme

Dengesiz veri setleri, modellerin azınlık sınıfını doğru tahmin etme kabiliyetini ciddi şekilde kısıtlar. Bu zorluğun üstesinden gelmek için çeşitli dengeleme teknikleri uygulanmaktadır. Rok Blagus ve Lara Lusa (2013) tarafından yapılan kapsamlı bir araştırma, SMOTE (Synthetic Minority Oversampling Technique) algoritmasının bu problemdeki rolünü incelemiştir. SMOTE, azınlık sınıfından sentetik örnekler oluşturarak veri setini dengelemeyi amaçlar. Araştırma, SMOTE'un düşük boyutlu verilerde çoğu sınıflandırıcı için faydalı olduğunu, ancak yüksek boyutlu verilerde çoğu zaman çoğunluk sınıfına yönelik yanlılığı azaltmada etkili olmadığını ve alt örneklemeden daha az verimli olduğunu bulmuştur. Yüksek boyutlu verilerde SMOTE'un sadece k-NN sınıflandırıcıları için, ancak değişken seçimi yapılması koşuluyla önemli fayda sağladığı vurgulanmıştır. Bu makale, dengesiz veri setlerinden sınıflandırıcı üretimi için SMOTE (Synthetic Minority Over-sampling Technique) adlı yeni bir yöntem önermektedir. Yöntem, azınlık sınıfı örneklerini sentetik olarak çoğaltırken, çoğunluk sınıfını azaltarak sınıflandırma performansını artırmayı amaçlar. SMOTE'un, ROC

alanında diğerklasik yöntemlere kıyasla daha başarılı sonuçlar verdiği deneysel olarak kanıtlanmıştır. [25] Bu çalışma, dengesiz veri setleriyle sınıflandırma problemlerinde yaygın olarak kullanılan SMOTE ve türevlerini incelemektedir. SMOTE'un temel prensibi, azınlık sınıf örnekleri ile onların K-en yakın komşuları arasında sentetik örnekler oluşturarak sınıflar arası karar sınırlarını daha iyi tanımlamaktır. Makale, imbalanced data sorunlarını, performans değerlendirme ölçütlerini ve gelecekteki araştırma alanlarını kapsamlı biçimde ele almaktadır. [26] Makale, dengesiz sınıf dağılımının derin öğrenme modelleri üzerindeki olumsuz etkilerini ele alarak bu soruna yönelik DeepSMOTE adlı yeni bir oversampling yöntemi önermektedir. Encoder-decoder mimarisi, SMOTE tabanlı örnekleme ve özel bir ceza fonksiyonu içeren bu yöntem, GAN'lere kıyasla discriminator gerektirmeden yüksek kaliteli sentetik görseller üretme kapasitesi ile öne çıkmaktadır. [27]

2.3 Zaman Serisi ve Davranışsal Akış Analizi

E-ticaret ortamında gelecekteki satış trendlerini öngörmek, müşteri davranışlarının zaman içindeki dinamiklerini anlamak ve müşteri yolculuklarını analiz etmek, işletmeler için kritik stratejik değer taşır.

2.3.1 Zaman Serisi Tahmin Modelleri

Gelecekteki satış veya işlem hacmi gibi zaman serisi verilerini tahmin etmek için çeşitli modeller kullanılmaktadır. Klasik zaman serisi modelleri arasında **Holt-Winters (HW)** ve **ARIMA (AutoRegressive Integrated Moving Average)** öne çıkar. Claudimar Pereira Da Veiga ve arkadaşları (2014) tarafından yapılan bir çalışma, perakende sektöründe talep tahmini için bu iki modeli karşılaştırmıştır. Holt-Winters metodu, seviye, trend ve mevsimsellik bileşenlerini ayrı ayrı ele alan, üstel düzeltmenin bir uzantısıdır; basitliği ve pratik uygulamalardaki iyi performansı ile bilinir [28]. ARIMA modeli ise, geçmiş değerlere (AR), hata terimlerine (MA) ve seriyi durağanlaştırmak için fark alma (I) işlemine dayanarak tahmin yapar; güçlü teorik temellere sahiptir [28]. Araştırma, HW modelinin genellikle ARIMA'ya kıyasla daha düşük hata metrikleri ile daha iyi tahminler sunduğunu belirtmiştir [28]. Zaman serisi veri madenciliği üzerine yapılan bu kapsamlı çalışmada, temsil ve indeksleme, benzerlik ölçümü, segmentasyon, görselleştirme ve madencilik gibi temel alanlar incelenmiştir. Özellikle zaman serilerinin yüksek boyutlu, sürekli güncellenen ve bütünsel olarak ele alınması gereken doğası, geleneksel veri madenciliği yöntemlerinden farklı stratejilerin geliştirilmesini gerekli kılmaktadır. Makale, bu alanlardaki son gelişmeleri

derleyerek, araştırmacılar için yol gösterici bir kaynak olmayı amaçlamaktadır. Ayrıca veri akışları, çok nitelikli zaman serileri ve gizlilik gibi güncel araştırma sorunlarına da dikkat çekilmektedir. Bu yönüyle çalışma, hem literatür taraması hem de gelecek çalışmalara yön verme potansiyeli taşımaktadır. [29]

Daha gelişmiş ve hibrit yaklaşımlar da zaman serisi tahminlerinde etkinliğini göstermiştir. Zhoufan Chen ve arkadaşları (2023), dağıtım şebekelerindeki tepe yük tahmini için Prophet ve LSTM algoritmalarını birleştiren **Prophet-LSTM** hibrit modelini önermişlerdir. Prophet modeli, doğrusal olmayan eğilimleri modellemek ve dönersellik (haftalık, aylık, yıllık) ile tatil etkilerini otomatik olarak yönetmek için eklemeli bir yapı kullanır ve eksik verilerde sağlamdır [30]. LSTM (Long Short-Term Memory) ise, tekrarlayan sinir ağlarının (RNN) bir türü olarak, özellikle uzun vadeli bağımlılıkları öğrenme ve karmaşık zaman serisi desenlerini modelleme yeteneğine sahiptir; geleneksel RNN'lerdeki gradyan kaybolması/patlaması sorunlarını çözmek için unutma, giriş ve çıkış kapıları gibi mekanizmalar içerir [30]. Chen ve arkadaşları (2023), Prophet-LSTM algoritmasının bireysel Prophet ve LSTM modellerinden ve diğer derin öğrenme modellerinden (GRU, TCN, Transformer) daha iyi performans gösterdiğini kanıtlamıştır [30]. Karim et al. (2019), tek değişkenli zaman serisi sınıflandırmasında başarı gösteren LSTM-FCN ve Attention LSTM-FCN modellerini çok değişkenli zaman serisi sınıflandırması için geliştirerek MLSTM-FCN ve MALSTM-FCN modellerini önermiştir. Bu yeni modeller, fully convolutional bloklara squeeze-and-excitation blokları eklenerek genişletilmiştir. 35 veri setinde yapılan deneylerde bu modeller, minimal ön işleme ihtiyaç duyarken birçok durumda mevcut en iyi yöntemlerden daha yüksek doğruluk sağlamıştır. Ayrıca, bu yapılar test zamanında da oldukça verimli olup, bellek kısıtlı sistemlerde dağıtılabılır niteliktedir. [31] Bu çalışmada, finansal zaman serilerinin doğruluğunu artırmak amacıyla iki yeni melez tahmin modeli önerilmiştir: EMD-LSTM ve CEEMDAN-LSTM. EMD (Empirical Mode Decomposition) ve gelişmiş versiyonu CEEMDAN (Complete Ensemble Empirical Mode Decomposition with Adaptive Noise), karmaşık ve gürültülü finansal verileri ayrıştırarak her bir bileşeni Long Short-Term Memory (LSTM) ile ayrı ayrı tahmin eder. Ardından, bu tahminler birleştirilerek nihai tahmin elde edilir. Model, SP500, HSI, DAX ve SSE endekslerine uygulanmış ve tek başına kullanılan LSTM, SVM, MLP gibi modellere göre üstün sonuçlar vermiştir. Bu yaklaşım, gürültü azaltma ve zaman serisinin çok katmanlı yapısını daha iyi analiz etme açısından oldukça etkilidir. [32]

2.3.2 Davranışsal Akış Analizi (Markov Zincirleri)

Müşterilerin e-ticaret platformlarındaki dijital yolculuklarını ve davranışsal geçişlerini modellemek, dönüşüm hunisindeki potansiyel darboğazları ve kritik etkileşim noktalarını belirlemek için Markov Zincirleri güçlü bir analitik araç olarak kullanılmaktadır. Bu yöntem, kullanıcıların bir eylem durumundan diğerine geçiş olasılıklarını nicel olarak ölçmeye olanak tanır (örneğin, bir ürünü görüntüledikten sonra sepete ekleme veya doğrudan satın alma). Elde edilen geçiş matrisi, müşteri yolculuğundaki temel akışları ve potansiyel terk alanlarını anlamak için değerli içgörüler sunar. Bu analizler, işletmelere müşteri deneyimini optimize etme ve dönüşüm oranlarını artırma konusunda stratejik bilgiler sağlar. Bu çalışmada, Borsa İstanbul'da orman ürünleri ve mobilya sektöründe faaliyet gösteren firmaların hisse senedi fiyatları, Markov zincirleri yöntemiyle analiz edilmiştir. Altı farklı firmanın 1 yıllık kapanış fiyatları kullanılarak oluşturulan geçiş olasılıkları yardımıyla, dört firmanın fiyat hareketleri başarılı şekilde tahmin edilmiştir. Bulgular, Markov zincirleri yönteminin bu sektördeki hisse senedi fiyat tahmini için etkili bir araç olduğunu ortaya koymaktadır. [33]

2.4 Model Yorumlanabilirliği (Explainable AI)

Makine öğrenmesi modellerinin, özellikle Random Forest ve XGBoost gibi karmaşık "kara kutu" algoritmaların, tahminlerini hangi özelliklere dayanarak yaptığını anlamak, modelin güvenilirliği ve iş kararları açısından kritik bir gerekliliktir [1]. Bu şeffaflık ihtiyacını karşılamak için **SHAP** (SHapley Additive exPlanations) gibi yorumlanabilirlik yöntemleri geliştirilmiştir. SHAP, oyun teorisi tabanlı Shapley değerlerini kullanarak her bir özelliğin bir tahmine olan katkısını belirler. Bu sayede, model kararlarının hem yerel (tekil tahminler için) hem de global (genel model davranışı için) düzeyde açıklanması sağlanır [34]. Jiangtao Li ve arkadaşları (2022) tarafından yapılan bir çalışmada, XGBoost modelinin özellik katkılarının SHAP değerleri teorisiyle nasıl analiz edildiği gösterilmiştir; bu tür analizler, modelin iç mekanizmalarını şeffaf hale getirerek, işletmelere hangi müşteri davranışlarının satın alma olasılığını artırdığına veya azalttığına dair eyleme dönüştürülebilir içgörüler sağlar [34].

2.5 E-ticaret ve Müşteri Davranışları Alanındaki Uygulama Çalışmaları

Literatürdeki birçok çalışma, çeşitli makine öğrenmesi ve veri madenciliği tekniklerinin e-ticaret ve müşteri davranışları analizi alanındaki pratik

uygulamalarını ve sağladığı faydaları ortaya koymaktadır. Bu çalışmalar, teorik modellerin gerçek dünya problemlerine nasıl uygulandığına dair önemli örnekler sunar.

- **Satış Tahminlemesi Uygulamaları:** Vahid Sinap (2024), Black Friday satış tahmini gibi kritik iş süreçlerinde regresyon algoritmalarını karşılaştırarak XGBoost'un üstün performansını ve hiperparametre ayarlamasının önemini vurgulamıştır [1]. Zeenia Singla ve arkadaşları (2017) ise, müşteri ürün incelemelerinin duygu analizinde SVM'nin yüksek doğrulukla öne çıktığını ve online incelemelerin tüketici satın alma davranışlarını nasıl etkilediğini göstermiştir [35].
- **Müşteri Segmentasyonu ve CRM Uygulamaları:** Narendra Singh ve arkadaşları (2020), müşteri ilişkileri yönetiminde (CRM) demografik özelliklere dayalı müşteri segmentasyonu ve sınıflandırmasında Çok Katmanlı Algılayıcı (MLP) gibi modellerin etkinliğini göstermiştir [2]. Patel ve arkadaşları (2020) da müşteri segmentasyonu için KMeans gibi kümeleme algoritmalarının önemini vurgulamış ve bu segmentasyonun hedefli pazarlama stratejileri için nasıl kullanılabileceğini belirtmişlerdir [7].
- **E-ticaret Kullanımı Analizi ve Etkileyen Faktörler:** Yunus Emre GÜR ve arkadaşları (2024), Türkiye'deki e-ticaret kullanımını etkileyen demografik (yaş, eğitim durumu), teknolojik (internet bankacılığı kullanımı, internet kullanım sıklığı) ve sosyal medya (Instagram, YouTube kullanımı) faktörlerini makine öğrenmesiyle analiz etmiştir. Çalışma, özellikle Gradient Boosting modelinin yüksek doğrulukla e-ticaret kullanımını sınıflandırmada etkin olduğunu ve eğitim durumu gibi faktörlerin belirleyici rol oynadığını ortaya koymuştur [36].
- **Geniş Kapsamlı ve Stratejik Uygulamalar:** Makine öğrenmesi, ürün sınıflandırma, dolandırıcılık tespiti, müşteri davranışlarını öngörme, kişiselleştirilmiş pazarlama ve satış/iş planlaması gibi kritik e-ticaret işlevlerinde etkin bir araç olarak öne çıkmaktadır. Bu çalışmaların tümü, makine öğrenmesi tekniklerinin e-ticaret operasyonlarını dönüştürme ve işletmelere veriye dayalı karar verme, güvenlik önlemlerini güçlendirme ve müşteri deneyimlerini geliştirme fırsatları sunma potansiyelini açıkça ortaya koymaktadır. Chopra ve arkadaşları (2022) tarafından yapılan bir çalışma, FastAPI'nin makine öğrenmesi modellerini sunmak için asenkron yetenekleri ve hafif yapısı nedeniyle ideal bir seçim olduğunu belirtmektedir. Ancak, doğrudan model yürütmesine kıyasla API istek yönetiminin getirdiği ek

yük nedeniyle çıkarım gecikmesini (latency) bir miktar artırabileceği ve verimi (throughput) hafifçe azaltabileceği de gösterilmiştir. Buna rağmen, model niceleme (quantization) ve toplu işleme (batch processing) gibi optimizasyon tekniklerinin uygulanmasıyla bu ek yükün hafifletilebileceği ve genel performansın artırılabilceği vurgulanmaktadır [37].

2.6 Metodoloji

Bu bölümde, RetailRocket platformundan elde edilen olay kayıtları (event log) verisi üzerinde e-ticaret müşteri davranışlarını analiz etmek için izlenen adımlar ve uygulanan teknik metodolojiler detaylı olarak açıklanacaktır. Çalışmanın dört ana hedefi doğrultusunda kullanılan veri ön işleme, özellik mühendisliği, modelleme ve değerlendirme yaklaşımları adım adım sunulacaktır.

2.6.1 Veri Seti

Bu tez çalışmasında, **RetailRocket platformundan elde edilen gerçek zamanlı olay kayıtları (event log) verisi** kullanılmıştır. Veri seti, bir e-ticaret sitesindeki kullanıcıların etkileşimlerini kronolojik olarak kaydetmekte olup, yaklaşık **4.5 aylık bir dönem** boyunca toplanmıştır. Sağlanan ham veri, gizlilik nedenleriyle tüm değerleri karma (hashed) formda sunmaktadır.

- **Veri Kaynağı ve Yapısı:** Kullanılan veri seti, RetailRocket platformu üzerinden akademik çalışmalara özel olarak sunulmuş, anonimleştirilmiş gerçek dünya e-ticaret verilerinden oluşmaktadır. Veri seti, temelde üç ayrı dosyadan oluşmaktadır:

1. `events.csv` (Davranış Verisi): Kullanıcı etkileşimlerini içeren ana olay kayıtları dosyasıdır. Her bir satır, benzersiz bir kullanıcı etkileşimini temsil etmekte ve aşağıdaki temel sütunları içermektedir:
 - `timestamp`: Olayın gerçekleştiği Unix zaman damgası. (`int` formatında, işleme sırasında `datetime`'a dönüştürülmüştür)
 - `visitorid`: Etkileşimi gerçekleştiren kullanıcının benzersiz karma (hashed) kimliği. (`str` formatında)
 - `event`: Gerçekleşen etkileşim türü. Temel olarak **"view"** (ürün görüntüleme), **"addtocart"** (sepete ekleme) ve **"transaction"** (satın alma) olmak üzere üç ana olay tipi bulunmaktadır.
 - `itemid`: Etkileşimde bulunan ürünün benzersiz karma (hashed) kimliği. (`str` formatında)

- `transactionid` (Sadece 'transaction' olayları için): Satın alma işleminin benzersiz karma (hashed) kimliği.
 - 2. `item_properties.csv` (Ürün Özellikleri Verisi): Ürünlere ait özellikleri (örn. fiyat, kategori vb.) içeren dosyadır. Bu dosya, **20.275.902** satır ile **417.053** tekil öğenin farklı özelliklerini tanımlamaktadır. Bir öğenin özellikleri zaman içinde değişebileceğinden (örn. fiyat), her bir satır ilgili zaman damgasını içermektedir ve `events.csv` dosyasındaki davranış verisiyle eşzamanlı olarak haftalık anlık görüntüler şeklinde birleştirilmiştir.
 - 3. `category_tree.csv` (Kategori Ağacı Verisi): Ürün kategorileri arasındaki hiyerarşik ilişkiyi (üst-alt kategori) tanımlayan dosyadır. Bu dosya, `itemid` ve `categoryid` sütunları aracılığıyla `item_properties.csv` dosyasındaki kategori kimlikleriyle ilişkilendirilerek, ürün kategorilerinin daha derinlemesine anlaşılmasına olanak tanımıştır.
- **Veri Büyüklüğü ve Dağılımı:** `events.csv` dosyası, toplamda **2.756.101** olay kaydı içermektedir. Bu olaylar arasında **2.664.312** görüntüleme (%96.67), **69.332** sepete ekleme (%2.52) ve **22.457** satın alma (%0.81) eylemi bulunmaktadır. Bu olaylar, **1.407.580** tekil ziyaretçi tarafından gerçekleştirilmiştir. Giriş bölümünde de vurgulandığı gibi, `transaction` (satın alma) olaylarının (%0.81), tüm olayların çok küçük bir kısmını oluşturması, **aşırı dengesiz sınıf dağılımı** problemine yol açmıştır. Bu durum, özellikle satın alma tahmini probleminde, modellerin azınlık sınıfı ("satın alma") doğru bir şekilde öğrenmesini sağlamak için özel metodolojik yaklaşımlar (örn. SMOTE) gerektirmiştir.
 - **İlk Gözlemler ve Veri Kalitesi:** Veri setinin ilk incelenmesinde, `item_properties.csv` dosyasındaki bazı özelliklerde (örneğin fiyat bilgisi) potansiyel aykırı değerler gözlemlenmiştir. Ayrıca, olay verilerinin yaklaşık %90'ı için `item_properties.csv` dosyasında ilgili özellik bilgilerinin bulunması, analize dahil edilecek veri setinin kapsamını ve potansiyel bilgi yoğunluğunu göstermiştir. Bu durumlar, veri ön işleme aşamasında uygun stratejilerle ele alınmıştır.

2.6.2 Veri Ön İşleme ve Özellik Mühendisliği

Ham veri seti üzerinde doğrudan analiz yapmak yerine, tutarlılık sağlamak, eksik değerleri yönetmek ve modelleme için uygun özellikleri türetmek amacıyla kapsamlı bir ön işleme ve özellik mühendisliği süreci uygulanmıştır. Bu süreç, ham

veriden eyleme dönüştürülebilir içgörüler elde etmek ve modellerin performansını optimize etmek için hayati öneme sahiptir.

Veri Setlerinin Yüklenmesi ve İlk İnceleme: Çalışma kapsamında kullanılan RetailRocket olay verisi (`events.csv`), ürün özellikleri (`item_properties.csv`) ve kategori ağacı (`category_tree.csv`) dosyaları, Python programlama dili ve **pandas** kütüphanesi kullanılarak belleğe yüklenmiştir. Yükleme sonrası her bir veri setinin genel yapısı, boyutu ve sütun özellikleri hakkında ilk izlenimler edinmek amacıyla **head()**, **info()** ve **describe()** gibi temel pandas fonksiyonları kullanılmıştır. Bu aşamada, veri setlerinin sütun isimleri, veri tipleri ve temel istatistiksel özetleri incelenmiştir.

Sütun İsimlerinin Düzenlenmesi: Veri setlerindeki sütun isimlerinde olası tutarsızlıklar, boşluklar veya özel karakterler kontrol edilerek, standart ve anlaşılır bir formata dönüştürülmüştür. Bu düzenleme, sonraki kodlama ve analiz adımlarında olası hataları önlemek için kritik bir adımdır.

Eksik Değer Analizi ve Yönetimi: Tüm sütunlarda **eksik değerlerin varlığı** `isnull().sum()` fonksiyonu ile **sistemik olarak kontrol edilmiştir**. Özellikle `itemid` ve `visitorid` gibi temel tanımlayıcı sütunlardaki eksik değerler içeren satırlar, veri bütünlüğünü korumak adına `events.csv` veri setinden çıkarılmıştır. `item_properties.csv` dosyasında ise, eksik özellik değerleri bulunan kayıtlar, analizin amacına göre uygun bir strateji (örneğin, kategorik değerler için "Bilinmiyor" ataması veya analizin dışında bırakma) ile ele alınmıştır.

Veri Tiplerinin Kontrolü ve Dönüştürülmesi: Veri setlerindeki sütunların veri tipleri detaylıca incelenmiştir. Özellikle `timestamp` sütunu, zaman serisi analizleri ve süre hesaplamaları için **pandas datetime formatına dönüştürülmüştür**. `visitorid`, `itemid` ve `transactionid` gibi benzersiz tanımlayıcılar, `string (object)` olarak işlenerek sayısal bir manipülasyona maruz kalmamaları sağlanmıştır. Diğer sayısal sütunlar (örneğin `price`), ilgili analizler için uygun sayısal formatlara (`float` veya `int`) dönüştürülmüştür.

Aykırı Değer Analizi ve Yönetimi: Sayısal özelliklerdeki (örn. `price`, türetilen etkileşim sayıları) **potansiyel aykırı değerler**, **kutu grafikleri (boxplot)** ve **histogramlar** gibi görselleştirme teknikleri ile birlikte **çeyrekler arası aralık (IQR) tabanlı istatistiksel yöntemlerle incelenmiştir**. Tespit edilen mantıksız derecede yüksek veya düşük değerler, model performansını olumsuz etkilememesi için ya belirli eşik değerlerle sınırlandırılmış ya da ilgili analiz dışı bırakılmıştır.

Benzersiz Değerlerin ve Kategorilerin İncelenmesi: `visitorid`, `itemid`, `event` ve `categoryid` gibi kategorik ve tanımlayıcı sütunlardaki benzersiz değerlerin sayıları ve dağılımları analiz edilmiştir. Bu analizler, veri setlerinin çeşitliliğini ve kapsamını anlamada yardımcı olmuştur. Örneğin, platformdaki toplam tekil kullanıcı, ürün ve kategori sayıları belirlenerek veri setinin kapsamına dair net bir tablo oluşturulmuştur.

Sınıf Dağılımının ve Dengesizliğin Analizi: Çalışmanın en kritik bulgularından biri, `events.csv` dosyasındaki `event` sütununda yer alan `transaction` (satın alma) sınıfının **aşırı derecede dengesiz dağılımı** olmuştur. Veri setindeki satın alma olaylarının oranı, tüm olayların yalnızca %0.81'i olarak tespit edilmiştir. Bu durum, makine öğrenmesi modellerinin azınlık sınıfı öğrenme kabiliyeti veya genel performansını olumsuz etkileyeceğinden, sonraki adımlarda **SMOTE** gibi dengeleme tekniklerinin uygulanmasını zorunlu kılmıştır.

Event Tiplerine Göre Veri Dağılımı ve Zaman İçindeki Dağılım: Her bir olay türünün (`view`, `addtocart`, `transaction`) genel ve oransal dağılımı hesaplanarak müşteri yolculuğundaki temel eylemlerin yoğunluğu belirlenmiştir. Ayrıca, olayların **zaman içindeki dağılımı** (**günlük, haftalık, aylık**) çizgi grafikleri ile görselleştirilerek, platformdaki yoğun dönemler, mevsimsel eğilimler veya kampanya etkileri gibi zamana bağlı davranışsal paternler tespit edilmiştir. Bu analiz, zaman serisi tahminleme modelleri için önemli ön bilgiler sağlamıştır.

Temel Metriklerin Hesaplanması: Veri setlerinden, e-ticaret analitiği için temel teşkil eden bazı özet metrikler türetilmiştir. Bunlar arasında **ortalama sepet boyutu** (satın alma başına düşen ortalama ürün sayısı) ve **kullanıcı başına ortalama etkileşim sayıları** (görüntüleme, sepete ekleme, satın alma) yer almaktadır. Bu metrikler, müşteri davranışlarına dair ilk nicel içgörülerini sunmuştur.

Kullanıcı Bazlı Özelliklerin Oluşturulması (Feature Engineering): Model tabanlı analizler için her bir kullanıcıyı temsil eden zengin bir özellik seti oluşturulmuştur. Bu özellikler, kullanıcının site içindeki genel davranışlarını ve eğilimlerini yansıtmaktadır. Bu süreçte, `events.csv` ve `item_properties.csv` dosyaları `itemid` ve `timestamp` üzerinden birleştirilerek, ürün özelliklerinin kullanıcı davranışlarına dahil edilmesi sağlanmıştır. Oluşturulan özellikler:

- **Etkileşim Sayıları:** Her bir kullanıcı için toplam `view`, `addtocart` ve `transaction` olay sayıları hesaplanmıştır.
- **Dönüşüm Oranları:** `view`'dan `transaction`'a ve `addtocart`'tan

transaction'a dönüşüm oranları türetilmiştir.

- **Zaman Bazlı Metrikler:** Kullanıcının ilk ve son etkileşimi arasındaki geçen süre, ortalama oturum süresi gibi metrikler hesaplanmıştır.
- **RFM Metrikleri:** Yeterli fiyat bilgisi `item_properties.csv` dosyasından elde edildiğinden, kullanıcı başına **Recency** (son satın almadan bu yana geçen süre), **Frequency** (satın alma sıklığı) ve **Monetary** (toplam harcanan miktar) değerleri hesaplanarak RFM analizi için temel oluşturulmuştur. Bu metrikler, müşteri yaşam boyu değerini anlamak için kritik öneme sahiptir.
- **Ürün Kategori Tercihleri:** Kullanıcıların en çok etkileşimde bulunduğu kategoriler veya belirli kategorilerdeki etkileşim yoğunlukları gibi özellikler, `categoryid` ve `category_tree.csv` kullanılarak türetilmiştir.

Gereksiz Sütunların Kaldırılması ve Veri Setinin Son Hali: Analiz ve modelleme süreçleri için doğrudan gereksinim duyulmayan veya yüksek oranda eksik değer içeren sütunlar, veri setinin boyutsallığını azaltmak ve modelleme verimliliğini artırmak amacıyla çıkarılmıştır. Tüm bu ön işleme ve özellik mühendisliği adımlarının ardından, analiz ve modelleme için kullanılacak olan nihai ve temizlenmiş veri seti oluşturulmuş ve kaydedilmiştir.

3.1 Metodoloji

Bu bölümde, e-ticaret müşteri davranışlarını anlamak ve tahminlemek amacıyla gerçekleştirilen analizlerin **bilimsel çerçevesi ve uygulama adımları** detaylandırılmıştır. Çalışmanın problemleri ve hedefleri doğrultusunda, izlenen veri işleme süreçleri, geliştirilen makine öğrenmesi modelleri ve bu modellerin performans değerlendirme metrikleri sistematik bir yaklaşımla sunulmuştur. Bu bölüm, araştırmanın **nasıl** yürütüldüğünü ve elde edilen bulguların **güvenilirliğini ve tekrarlanabilirliğini** sağlayacak tüm metodolojik ayrıntıları içermektedir.

3.1.1 Veri Seti

Bu tez çalışmasında, gerçek zamanlı olay kayıtları (event log) içeren ve açık kaynak bir dataseti olan RetailRocket veri seti kullanılmıştır. Bu veri seti, bir e-ticaret sitesindeki kullanıcıların etkileşimlerini kronolojik olarak kaydetmekte olup yaklaşık 4.5 aylık bir dönem boyunca toplanmıştır. Sağlanan ham veri, gizlilik nedenleriyle tüm değerleri karma (hashed) formda sunmaktadır.

- **Veri Kaynağı ve Yapısı:** Kullanılan RetailRocket veri seti, Kaggle platformu üzerinden akademik çalışmalara özel olarak sunulmuş, anonimleştirilmiş gerçek dünya e-ticaret verilerinden oluşmaktadır. Veri seti, temelde üç ayrı dosyadan oluşmaktadır:

1. `events.csv` (Davranış Verisi): Kullanıcı etkileşimlerini içeren ana olay kayıtları dosyasıdır. Her bir satır, benzersiz bir kullanıcı etkileşimini temsil etmekte ve aşağıdaki temel sütunları içermektedir:
 - `timestamp`: Olayın gerçekleştiği Unix zaman damgası. (`int` formatında, işleme sırasında `datetime`'a dönüştürülmüştür)
 - `visitorid`: Etkileşimi gerçekleştiren kullanıcının benzersiz karma (hashed) kimliği. (`str` formatında)

- **event**: Gerçekleşen etkileşim türü. Temel olarak **"view"** (ürün görüntüleme), **"addtocart"** (sepete ekleme) ve **"transaction"** (satın alma) olmak üzere üç ana olay tipi bulunmaktadır.
 - **itemid**: Etkileşimde bulunulan ürünün benzersiz karma (hashed) kimliği. (str formatında)
 - **transactionid** (Sadece 'transaction' olayları için): Satın alma işleminin benzersiz karma (hashed) kimliği.
2. **item_properties.csv** (Ürün Özellikleri Verisi): Ürünlere ait özellikleri (örn. fiyat, kategori vb.) içeren dosyadır. Bu dosya, **20.275.902** satır ile **417.053** tekil öğenin farklı özelliklerini tanımlamaktadır. Bir öğenin özellikleri zaman içinde değişebileceğinden (örn. fiyat), her bir satır ilgili zaman damgasını içermektedir ve **events.csv** dosyasındaki davranış verisiyle eşzamanlı olarak haftalık anlık görüntüler şeklinde birleştirilmiştir.
3. **category_tree.csv** (Kategori Ağacı Verisi): Ürün kategorileri arasındaki hiyerarşik ilişkiyi (üst-alt kategori) tanımlayan dosyadır. Bu dosya, **itemid** ve **categoryid** sütunları aracılığıyla **item_properties.csv** dosyasındaki kategori kimlikleriyle ilişkilendirilerek, ürün kategorilerinin daha derinlemesine anlaşılmasına olanak tanımıştır.
- **Veri Büyüklüğü ve Dağılımı**: **events.csv** dosyası, toplamda **2.756.101** olay kaydı içermektedir. Bu olaylar arasında **2.664.312** görüntüleme (%96.67), **69.332** sepete ekleme (%2.52) ve **22.457** satın alma (%0.81) eylemi bulunmaktadır. Bu olaylar, **1.407.580** tekil ziyaretçi tarafından gerçekleştirilmiştir. Giriş bölümünde de vurgulandığı gibi, **transaction** (satın alma) olaylarının (%0.81), tüm olayların çok küçük bir kısmını oluşturması, **aşırı dengesiz sınıf dağılımı** problemine yol açmıştır. Bu durum, özellikle satın alma tahmini probleminde, modellerin azınlık sınıfı ("satın alma") doğru bir şekilde öğrenmesini sağlamak için özel metodolojik yaklaşımlar (örn. SMOTE) gerektirmiştir.
 - **İlk Gözlemler ve Veri Kalitesi**: Veri setinin ilk incelenmesinde, **item_properties.csv** dosyasındaki bazı özelliklerde (örneğin fiyat bilgisi) potansiyel aykırı değerler gözlemlenmiştir. Ayrıca, olay verilerinin yaklaşık %90'ı için **item_properties.csv** dosyasında ilgili özellik bilgilerinin bulunması, analize dahil edilecek veri setinin kapsamını ve potansiyel bilgi yoğunluğunu göstermiştir. Bu durumlar, veri ön işleme aşamasında uygun stratejilerle ele alınmıştır.

3.1.2 Veri Ön İşleme ve Özellik Mühendisliği

Ham veri seti üzerinde doğrudan analiz yapmak yerine, tutarlılık sağlamak, eksik değerleri yönetmek ve modelleme için uygun özellikleri türetmek amacıyla kapsamlı bir ön işleme ve özellik mühendisliği süreci uygulanmıştır. Bu süreç, ham veriden eyleme dönüştürülebilir içgörüler elde etmek ve modellerin performansını optimize etmek için hayati öneme sahiptir.

Veri Setlerinin Yüklenmesi ve İlk İnceleme: Çalışma kapsamında kullanılan RetailRocket olay verisi (`events.csv`), ürün özellikleri (`item_properties.csv`) ve kategori ağacı (`category_tree.csv`) dosyaları, Cursor (Yapay Zeka Destekli Kod Yazma Editörü) üzerinde, Python programlama dili ve **pandas** kütüphanesi kullanılarak belleğe yüklenmiştir. Yükleme sonrası her bir veri setinin genel yapısı, boyutu ve sütun özellikleri hakkında ilk izlenimler edinmek amacıyla **head()**, **info()** ve **describe()** gibi temel pandas fonksiyonları kullanılmıştır. Bu aşamada, veri setlerinin sütun isimleri, veri tipleri ve temel istatistiksel özetleri incelenmiştir.

Sütun İsimlerinin Düzenlenmesi: Veri setlerindeki sütun isimlerinde olası tutarsızlıklar, boşluklar veya özel karakterler kontrol edilerek, standart ve anlaşılır bir formata dönüştürülmüştür. Bu düzenleme, sonraki kodlama ve analiz adımlarında olası hataları önlemek için kritik bir adımdır.

Eksik Değer Analizi ve Yönetimi: Tüm sütunlarda **eksik değerlerin varlığı** **isnull().sum()** fonksiyonu ile **sistemik olarak kontrol edilmiştir**. Özellikle `itemid` ve `visitorid` gibi temel tanımlayıcı sütunlardaki eksik değerler içeren satırlar, veri bütünlüğünü korumak adına `events.csv` veri setinden çıkarılmıştır. `item_properties.csv` dosyasında ise, eksik özellik değerleri bulunan kayıtlar, analizin amacına göre uygun bir strateji (örneğin, kategorik değerler için "Bilinmiyor" ataması veya analizin dışında bırakma) ile ele alınmıştır.

Veri Tiplerinin Kontrolü ve Dönüştürülmesi: Veri setlerindeki sütunların veri tipleri detaylıca incelenmiştir. Özellikle `timestamp` sütunu, zaman serisi analizleri ve süre hesaplamaları için **pandas datetime formatına dönüştürülmüştür**. `visitorid`, `itemid` ve `transactionid` gibi benzersiz tanımlayıcılar, `string (object)` olarak işlenerek sayısal bir manipülasyona maruz kalmamaları sağlanmıştır. Diğer sayısal sütunlar (örneğin `price`), ilgili analizler için uygun sayısal formatlara (`float` veya `int`) dönüştürülmüştür.

Aykırı Değer Analizi ve Yönetimi: Sayısal özelliklerdeki (örn. `price`, türetilen

etkileşim sayıları) **potansiyel aykırı değerler, kutu grafikleri (boxplot) ve histogramlar gibi görselleştirme teknikleri ile birlikte çeyrekler arası aralık (IQR) tabanlı istatistiksel yöntemlerle incelenmiştir.** Tespit edilen mantıksız derecede yüksek veya düşük değerler, model performansını olumsuz etkilememesi için ya belirli eşik değerlerle sınırlandırılmış ya da analiz dışı bırakılmıştır.

Benzersiz Değerlerin ve Kategorilerin İncelenmesi: `visitorid`, `itemid`, `event` ve `categoryid` gibi kategorik ve tanımlayıcı sütunlardaki benzersiz değerlerin sayıları ve dağılımları analiz edilmiştir. Bu analizler, veri setlerinin çeşitliliğini ve kapsamını anlamada yardımcı olmuştur. Örneğin, platformdaki toplam tekil kullanıcı, ürün ve kategori sayıları belirlenerek veri setinin kapsamına dair net bir tablo oluşturulmuştur.

Sınıf Dağılımının ve Dengesizliğin Analizi: Çalışmanın en kritik bulgularından biri, `events.csv` dosyasındaki `event` sütununda yer alan `transaction` (satın alma) sınıfının **aşırı derecede dengesiz dağılımı** olmuştur. Veri setindeki satın alma olaylarının oranı, tüm olayların yalnızca %0.81'i olarak tespit edilmiştir. Bu durum, makine öğrenmesi modellerinin azınlık sınıfı öğrenme kabiliyeti veya genel performansını olumsuz etkileyeceğinden, sonraki adımlarda **SMOTE** gibi dengeleme tekniklerinin uygulanmasını zorunlu kılmıştır.

Event Tiplerine Göre Veri Dağılımı ve Zaman İçindeki Dağılım: Her bir olay türünün (`view`, `addtocart`, `transaction`) genel ve oransal dağılımı hesaplanarak müşteri yolculuğundaki temel eylemlerin yoğunluğu belirlenmiştir. Ayrıca, olayların **zaman içindeki dağılımı (günlük, haftalık, aylık)** çizgi grafikleri ile görselleştirilerek, platformdaki yoğun dönemler, mevsimsel eğilimler veya kampanya etkileri gibi zamana bağlı davranışsal paternler tespit edilmiştir. Bu analiz, zaman serisi tahminleme modelleri için önemli ön bilgiler sağlamıştır.

Temel Metriklerin Hesaplanması: Veri setlerinden, e-ticaret analitiği için temel teşkil eden bazı özet metrikler türetilmiştir. Bunlar arasında **ortalama sepet boyutu** (satın alma başına düşen ortalama ürün sayısı) ve **kullanıcı başına ortalama etkileşim sayıları** (görüntüleme, sepete ekleme, satın alma) yer almaktadır. Bu metrikler, müşteri davranışlarına dair ilk nicel içgörülerini sunmuştur.

Kullanıcı Bazlı Özelliklerin Oluşturulması (Feature Engineering): Model tabanlı analizler için her bir kullanıcıyı temsil eden zengin bir özellik seti oluşturulmuştur. Bu özellikler, kullanıcının site içindeki genel davranışlarını ve eğilimlerini yansıtmaktadır. Bu süreçte, `events.csv` ve `item_properties.csv` dosyaları `itemid` ve `timestamp` üzerinden birleştirilerek, ürün özelliklerinin kullanıcı davranışlarına dahil edilmesi

sağlanmıştır. Oluşturulan özellikler:

- **Etkileşim Sayıları:** Her bir kullanıcı için toplam view, addtocart ve transaction olay sayıları hesaplanmıştır.
- **Dönüşüm Oranları:** view'dan transaction'a ve addtocart'tan transaction'a dönüşüm oranları türetilmiştir.
- **Zaman Bazlı Metrikler:** Kullanıcının ilk ve son etkileşimi arasındaki geçen süre, ortalama oturum süresi gibi metrikler hesaplanmıştır.
- **RFM Metrikleri:** Yeterli fiyat bilgisi item_properties.csv dosyasından elde edildiğinden, kullanıcı başına **Recency** (son satın almadan bu yana geçen süre), **Frequency** (satın alma sıklığı) ve **Monetary** (toplam harcanan miktar) değerleri hesaplanarak RFM analizi için temel oluşturulmuştur. Bu metrikler, müşteri yaşam boyu değerini anlamak için kritik öneme sahiptir.
- **Ürün Kategori Tercihleri:** Kullanıcıların en çok etkileşimde bulunduğu kategoriler veya belirli kategorilerdeki etkileşim yoğunlukları gibi özellikler, categoryid ve category_tree.csv kullanılarak türetilmiştir.

Gereksiz Sütunların Kaldırılması ve Veri Setinin Son Hali: Analiz ve modelleme süreçleri için doğrudan gereksinim duyulmayan veya yüksek oranda eksik değer içeren sütunlar, veri setinin boyutsallığını azaltmak ve modelleme verimliliğini artırmak amacıyla çıkarılmıştır. Tüm bu ön işleme ve özellik mühendisliği adımlarının ardından, analiz ve modelleme için kullanılacak olan nihai ve temizlenmiş veri seti oluşturulmuş ve kaydedilmiştir.

3.1.3 Uygulanan Analiz Yöntemleri

Söz konusu çalışmanın dört ana hedefi doğrultusunda, veri biliminin farklı alanlarından çeşitli makine öğrenmesi ve istatistiksel modeller uygulanmıştır. Her bir hedef için belirlenen yöntemler aşağıda detaylandırılmıştır:

3.1.3.1 Sınıflandırma Modelleri ve Açıklanabilirlik (Tahminleme Hedefi)

Bu bölümde, bir kullanıcının satın alma yapıp yapmayacağını tahmin etmek için çeşitli sınıflandırma modelleri geliştirilmiştir. Özellikle aşırı dengesiz sınıf dağılımı göz önüne alındığında, modellerin performans değerlendirmesi için **F1 skoru** ana metrik olarak kullanılmıştır.

a. Veri Dengeleme: SMOTE Uygulaması Sınıflandırma problemi için veri setindeki aşırı dengesiz sınıf dağılımını (satın alma sınıfının %0.81 oranı) gidermek amacıyla **SMOTE (Synthetic Minority Over-sampling Technique)** algoritması kullanılmıştır.

SMOTE, sınıf dengesizliğine sahip veri setlerinde azınlık sınıfının örnek sayısını artırarak sınıflandırma performansını iyileştirmeyi amaçlayan popüler bir aşırı örnekleme yöntemidir. Temel hedefi, dengesiz bir veri setini sentetik olarak dengeli veya neredeyse dengeli bir eğitim setine dönüştürmektir.

SMOTE algoritmasının çalışma prensibi, azınlık sınıfındaki mevcut örneklerden yeni sentetik örnekler oluşturur. Her bir azınlık sınıfı örneği (x) için, Öklid mesafesine göre en yakın 5 azınlık sınıfı komşusu belirlenir (eğer azınlık sınıfı örneği sayısı 5'ten az ise $n_{min} - 1$ komşu kullanılır). Bu komşulardan rastgele seçilen bir x^R örneği kullanılarak, yeni sentetik örnek (S) aşağıdaki formül ile türetilir:

$$S = x + u \cdot (x^R - x)$$

Burada u , $U(0, 1)$ tekdüze dağılımından rastgele seçilen bir değerdir. Bu yöntem, sentetik örneğin, orijinal örnek ile seçilen komşusu arasındaki doğru üzerinde yer almasını sağlar.[3] Çalışmada, SMOTE tekniği kullanılarak sınıflandırma modellerinin azınlık sınıfını daha iyi öğrenmesi ve özellikle F1 skoru gibi dengesiz veri setlerine duyarlı metrikler üzerinde performansının artırılması hedeflenmiştir. SMOTE uygulaması yalnızca eğitim veri seti üzerinde gerçekleştirilerek, test veri setinin bağımsızlığı ve modelin genellenebilirliği korunmuştur. Uygulamada, `imbalanced-learn` kütüphanesinden SMOTE sınıfı kullanılarak **azınlık ve çoğunluk sınıflarının örnek sayıları eşitlenecek şekilde (tam dengeleme)** örnekleme yapılmıştır. Varsayılan `k_neighbors=5` değeri kullanılmıştır.

b. Model Seçimi ve Uygulama Geliştirilen sınıflandırma modelleri, farklı algoritmik yaklaşımları temsil etmek ve performans kıyaslaması yapmak amacıyla seçilmiştir:

- **Random Forest:** Rastgele Orman yaklaşımı, 2001 yılında L. Breiman tarafından geliştirilen, sınıflandırma ve regresyon görevleri için kullanılan güçlü bir topluluk öğrenme algoritmasıdır. Bu yöntem, birden fazla bağımsız karar ağacını bir araya getirerek çalışır ve tahminlerini birleştirir. Algoritma, özellikle değişken sayısının gözlem sayısından çok daha fazla olduğu yüksek boyutlu veri setlerinde etkili performans sergilemektedir.[16]

Çalışma Prensipleri:

Random Forest'ın temel çalışma prensibi, "böl ve yönet" yaklaşımına dayanır. Algoritma şu adımlarla ilerler:

- **Bagging (Önyükleme Örnekleme):** Her bir ağaç, orijinal eğitim veri setinden bootstrap örnekleme (yerine koyarak örnekleme) yoluyla elde edilen farklı bir alt küme (a_n gözlem) üzerinde eğitilir. Bu süreç, ağaçlar arasında çeşitliliği artırır.
- **Rastgele Özellik Seçimi:** Her bir ağacın her düğümünde (bölme yapılırken), tüm olası özellikler yerine, rastgele seçilen belirli bir $mtry$ adet özellik alt kümesi dikkate alınır. Bu rastgelelik, ağaçlar arasındaki korelasyonu azaltmaya yardımcı olur.
- **Ağaç Yapımı:** Her ağaç, belirli bir *nodesize* kriterine ulaşana kadar budanmadan tamamen büyütülür.
- **Tahminlerin Birleştirilmesi:** Regresyon problemleri için, tüm ağaçların tahminlerinin ortalaması alınarak nihai sonuç elde edilir. Sınıflandırma problemleri için ise, her ağacın yaptığı sınıflandırma tahminleri arasında çoğunluk oylaması yapılarak en sık tahmin edilen sınıf belirlenir.[16]

Temel Avantajları ve Özellikleri:

- **Yüksek Doğruluk:** Birden fazla ağacın birleşimi sayesinde genellikle tek bir karar ağacına göre daha yüksek tahmin doğruluğu sunar.
- **Aşırı Öğrenmeye Karşı Direnç:** Büyük sayıda ağaç kullanıldığında bile aşırı öğrenmeye eğilimli değildir.
- **Değişken Önemi Ölçümü:** Algoritma, veri setindeki her bir değişkenin tahmin üzerindeki göreceli önemini belirlemek için metrikler (örneğin, Ortalama Kirlilik Azalması - MDI ve Ortalama Doğruluk Azalması - MDA) sağlar.
- **Eksik Veri Yönetimi:** Eksik değerleri etkin bir şekilde işleme yeteneğine sahiptir. Bu, gözlemler arasındaki yakınlığı ölçen "yakınlık matrisi" kullanılarak gerçekleştirilir.
- **Ölçeklenebilirlik:** Kolayca paralelleştirilebilir olması sayesinde büyük gerçek dünya sistemleriyle başa çıkma potansiyeline sahiptir.
- **Out-of-Bag (OOB) Hatası:** Her ağacın eğitiminde kullanılmayan örnekler (OOB örnekleri) üzerinden model performansını dahili olarak tahmin etme yeteneği sunar. Bu, ayrı bir doğrulama setine duyulan ihtiyacı ortadan kaldırır.

Random Forest, esnekliği, yüksek performansı ve yorumlanabilirliği nedeniyle günümüzde en başarılı genel amaçlı algoritmalar arasında yer almaktadır. Yüksek doğruluk, aşırı uyumu (overfitting) azaltma yeteneği ve özellik önem derecesini (feature importance) çıkarabilme özelliği nedeniyle tercih edilmiştir. Modelin oluşturulmasında `n_estimators=100` değeri kullanılmış, diğer parametreler (örn. `max_depth`, `min_samples_split`, `min_samples_leaf`) varsayılan değerlerinde bırakılmıştır. `random_state=42` ile tekrarlanabilirlik sağlanmıştır. Hiperparametre optimizasyonu yapılmamıştır.

- **XGBoost (eXtreme Gradient Boosting):** Gradyan artırma (gradient boosting) çerçevesinin optimize edilmiş bir uygulamasıdır. Hız, ölçeklenebilirlik ve yüksek prediktif performans sunması nedeniyle büyük ve karmaşık veri setlerinde yaygın olarak kullanılmaktadır.

XGBoost (Extreme Gradient Boosting): Gelişmiş Gradyan Yükseltme

XGBoost (Extreme Gradient Boosting), yüksek tahminsel yeteneği ile bilinen gradyan yükseltme (Gradient Boosting) algoritmasının geliştirilmiş bir versiyonudur. Geleneksel gradyan yükseltme algoritmalarının, her seferinde bir karar ağacının oluşturulmasını gerektirmesi ve bu nedenle küçük boyutlu modellerin bile eğitiminin çok zaman alması sınırlamasını aşmak amacıyla geliştirilmiştir. XGBoost, performans ve hız açısından önemli iyileştirmeler sunar.[21]

XGBoost, çekirdeğinde gradyan yükseltmeyi barındırır. Gradyan yükseltme, bir kayıp fonksiyonunu minimize etmek için kademeli ve eklemeli bir model oluşturma sürecidir. Bu süreç genellikle üç ana adımdan oluşur:

- **Diferansiyellenebilir Kayıp Fonksiyonu:** Probleme uygun, türevlenebilir bir kayıp fonksiyonunun belirlenmesi gerekmektedir. Gradyan yükseltme modeli, farklı kayıp fonksiyonları için yeni algoritmaların türetilmesini gerektirmez; sadece uygun bir kayıp fonksiyonunun seçilip çerçeveye dahil edilmesi yeterlidir.
- **Zayıf Öğrenici Oluşturma:** Tahminler yapmak için bir zayıf öğrenici oluşturulur. Gradyan yükseltmede karar ağaçları, özellikle gerçek değer çıktısı üreten regresyon ağaçları zayıf öğrenici olarak seçilir. Bu ağaçlar art arda eklenerek tahminlerdeki artıkları (hataları) iyileştirir.
- **Eklemeli Model Oluşturma:** Zayıf öğrenicilerin tahminlerini bir araya getirerek kayıp fonksiyonunu azaltan eklemeli bir model oluşturulur.

Bu ağaç ekleme süreci teker teker gerçekleşir ve modelin son çıktısını iyileştirmek için yeni ağacın çıktısı mevcut ağaç dizisinin çıktısına eklenir.[21]

XGBoost'u geleneksel gradyan yükseltmeden ayıran temel farklılıklar şunlardır:

- **Parallelleştirme:** Zayıf öğrencilerin eklenme süreci ardışık olarak değil, çoklu iş parçacığı (multi-threaded) yaklaşımıyla gerçekleşir. Bu, makine CPU çekirdeklerinin etkin kullanımını sağlayarak daha yüksek hız ve performans sunar.
- **Seyrek Farkındalık ve Eksik Veri İşleme:** Seyrek veriye duyarlı bir uygulamaya sahiptir ve otomatik olarak eksik veri değerlerini işler.
- **Blok Yapısı:** Ağaç yapısının paralelleştirilmesini desteklemek için blok yapısını kullanır.
- **Devamlı Eğitim:** Daha önce eğitilmiş bir modelin yeni veriler üzerinde daha da geliştirilmesine olanak tanır.

XGBoost, sınıflandırma, regresyon ve tahminsel modelleme problemlerinde yapılandırılmış veya tablosal veri setlerinde üstün performans gösterdiği görülmüştür. Özellikle hız ve performans açısından, çoklu iş parçacığı uygulaması sayesinde geleneksel Gradyan Yükseltme algoritmalarına kıyasla genellikle daha iyi sonuçlar vermektedir. Doğruluk değerleri ise veri setine göre değişmekle birlikte, XGBoost genellikle rekabetçi sonuçlar sunmuştur. Bu özellikleriyle XGBoost, büyük ve karmaşık veri setleriyle çalışan birçok makine öğrenimi uygulamasında tercih edilen bir algoritma haline gelmiştir. Bu çalışmada ise XGBoost algortiması kullanılırken `n_estimators=100`, `learning_rate=0.1` ve `max_depth=5` ana hiperparametreleri kullanılmıştır. Diğer parametreler varsayılan değerlerinde bırakılmıştır. `random_state=42` ile tekrarlanabilirlik sağlanmıştır. Hiperparametre optimizasyonu yapılmamıştır.[21]

- **Lojistik Regresyon:** İki sınıf arasındaki ilişkiyi doğrusal bir kombinasyonla modelleyen basit ancak etkili bir algoritmadır. Doğrusal regresyonun ikili sonuçlar için yetersiz kaldığı durumlarda bu boşluğu doldurmak üzere geliştirilmiştir, zira doğrusal regresyonun tahminleri 0 ile 1 aralığının dışına çıkabilir ve sabit varyans varsayımı ikili sonuçların davranışına uymaz.[4]

Lojistik regresyon modeli, ilgilenilen ikili sonucun "olasılığına" dayanır. Bir olayın olasılığı, olayın gerçekleşme olasılığının gerçekleşmeme olasılığına oranıdır. Lojistik regresyon modeli, bu olasılığın doğal logaritmasını

$(\ln[\text{odds}(Y = 1)])$ tahmin edicilerin bir regresyon fonksiyonu olarak alır. Tek bir tahminci (X) ile model şu şekilde ifade edilir: $\ln[\text{odds}(Y = 1)] = \beta_0 + \beta_1 X$. Burada β_0 kesme terimi, β_1 ise tahminci X 'teki 1 birimlik değişime karşılık gelen olasılığın logaritmasındaki değişimi temsil eden regresyon katsayısıdır.

Lojistik regresyon sonuçları tipik olarak olasılık oranları (Odds Ratios - OR) ile sunulur. Bu oranlar, β_1 katsayısının üstelini alarak elde edilir ve X 'teki 1 birimlik değişime karşılık gelen olasılık oranını gösterir. Olasılık oranı eksi 1'in 100 ile çarpılması, X 'teki 1 birimlik artışa karşılık gelen olay olasılığındaki yüzde değişimi ifade eder. Birden fazla tahminci modele dahil edildiğinde, olasılık oranı değeri diğer tahmincilere göre ayarlanmış olur. Olasılık oranları bazen göreceli risk ile karıştırılsa da, yalnızca olayın olasılığı çok düşük olduğunda olasılık oranı göreceli riske iyi bir yaklaşım olarak kabul edilebilir. Olasılık oranları, modelden elde edilen doğal tahminler olduğundan tercih edilir ve bunları göreceli risklere dönüştürme girişimleri sonuçları bozabilir.[4]

Lojistik regresyon, sürekli veya kategorik tahmincilerin kullanımına izin verir ve birden fazla tahminci için ayarlama yeteneği sağlar. Bu özelliği, karşılaştırılan gruplardaki farklılıklardan kaynaklanan potansiyel yanlılığı azaltmak için ayarlama gerektiğinde, özellikle gözlemsel verilerin analizi için kullanışlı kılar. Doğrusal regresyonun aksine, lojistik regresyon için parametre (β) tahminlerinin doğrudan bir formülü yoktur; en iyi tahminleri bulmak için kararlılık elde edilene kadar yaklaşık tahminlerin tekrar tekrar iyileştirilmesi gerekir. Bu durum, bazı araştırmacılar için lojistik regresyonu "kara kutu" bir yaklaşıma dönüştürebilir.[4]

Modelin uygunluğu ve uyumu (goodness-of-fit) genellikle Hosmer ve Lemeshow testi gibi global ölçümlerle değerlendirilir. Lojistik regresyonun iki seviyeli sonuçların analizine yönelik standart formuna ek olarak, üç veya daha fazla sıralı veya sıralanmamış kategorili sonuçları analiz etmek için de uzantıları bulunmaktadır. Ayrıca, bağımsız olmayan (longitudinal veya kümelenmiş) veriler için genelleştirilmiş tahmin denklemleri veya rastgele etkili modeller gibi yöntemler kullanılabilir. Bu çalışmada, modelin temel performansı için bir referans noktası sağlamıştır. Parametre olarak `penalty='l2'` (L2 regülarizasyon), varsayılan `C=1.0` ve `solver='lbfgs'` kullanılmıştır. `random_state=42` ile tekrarlanabilirlik sağlanmıştır.

- **Naive Bayes:** Bayes teoremini temel alan, özellikler arasında koşullu bağımsızlık varsayımı yapan olasılıksal bir sınıflandırıcıdır. Özellikle

basitliđi ve hızlı hesaplama yeteneđi nedeniyle karşılařtırma amacıyla dahil edilmiřtir.[15] Uygulamada sürekli öznitelikler için uygun olan `sklearn.naive_bayes.GaussianNB` modeli ve varsayılan parametreleri kullanılmıřtır.

Naive Bayes sınıflandırıcısı, olasılıksal yaklařımlara dayanan, kořullu olasılık dađılımını ($P(C|D)$) modelleyerek sınıflandırma yapan popöler bir makine öđrenimi algoritmasıdır. Verilen bir nesnenin açıklaması (d) için, sınıflandırıcı en yüksek arka olasılıđa sahip sınıfı seçer: $\text{argmax}_C P(C = c|D = d)$. Bayes teoremi uyarınca, bu arka olasılık önsel dađılım ($P(C)$) ve olabilirlik ($P(D|C)$) kullanılarak hesaplanır. Paydadaki $P(D = d)$ terimi, sınıfa bađlı olmadığı için maksimum arka olasılık sınıfını belirlerken göz ardı edilebilir. Algoritmanın "naive" (saf) olarak adlandırılmasının nedeni, sınıf verildiđinde tüm özelliklerin kořullu olarak bađımsız olduđunu varsaymasıdır. Yani:

$$P(A_1 = a_1, \dots, A_n = a_n|C = c) = \prod_{i=1}^n P(A_i = a_i|C = c)$$

Bu varsayım, pratikte özellikler arasında etkileřimler olsa bile, Naive Bayes'in iyi sonuçlar verebilmesini sađlar. Bunun nedeni, algoritmanın doğrudan $P(d|c)$ deđerini tahmin etmekten ziyade, yalnızca en olası sınıfı doğru bir şekilde sıralamakla ilgilenmesidir. Bařka bir deyiřle, Naive Bayes kötü bir olasılık tahmindisi olsa bile iyi bir sıralayıcı olabilir.[15] Bu çalışmada veri seti, eğitim (%70) ve test (%30) kümelerine rastgele ayrılmıřtır. Modellerin genellenebilirliđini ve kararlılıđını sađlamak için **5-Katmanlı Çapraz Doğrulama (K-Fold Cross-Validation)** ($k=5$) teknikleri kullanılmıřtır. Bu yaklařım, veri setinin farklı alt kümeleri üzerinde modelin birden çok kez eğitilip test edilmesine olanak tanır. Çapraz doğrulama öncesinde veri `shuffle=True` ve `random_state=42` ile karıřtırılmıřtır.

c. Model Deđerlendirme Metrikleri Dengesiz sınıf dađılımı nedeniyle, model performansını deđerlendirmede ařađıdaki metrikler kullanılmıřtır:

- **F1 Skoru:** Hassasiyet (Precision) ve Hatırlama (Recall) arasındaki harmonik ortalamayı temsil eder. Özellikle azınlık sınıfın doğru tahmin edilmesindeki hassasiyet ve yakalama dengesini gösterdiđi için dengesiz veri setlerinde ana deđerlendirme metriđi olarak seçilmiřtir.
- **Hassasiyet (Precision):** Pozitif olarak tahmin edilen örneklerin gerçek pozitif olma oranını gösterir ($TP / (TP + FP)$).

- **Hatırlama (Recall) / Duyarlılık (Sensitivity)::** Gerçek pozitif örneklerin ne kadarının doğru bir şekilde tespit edildiğini gösterir ($TP / (TP + FN)$).
- **ROC AUC (Receiver Operating Characteristic - Area Under the Curve):** Sınıflandırıcının farklı eşik değerlerinde gerçek pozitif oranı ile yanlış pozitif oranı arasındaki değişimi gösteren bir eğridir. Modelin ayırım gücünü ölçer.
- **Karmaşıklık Matrisi (Confusion Matrix):** Her bir modelin doğru ve yanlış tahmin sayılarını (True Positives, False Positives, True Negatives, False Negatives) gösteren bir tablo olup, modelin sınıflandırma performansına dair detaylı bir bakış sunmuştur.

d. SHAP (Shapley Additive Explanations) ile Model Yorumlanabilirliği

Geliştirilen makine öğrenmesi modellerinin "kara kutu" doğasını şeffaf hale getirmek ve modelin hangi özelliklere dayanarak tahminler yaptığını anlamak amacıyla **SHAP** (SHapley Additive exPlanations) yöntemi kullanılmıştır. SHAP, oyun teorisi tabanlı Shapley değerlerini kullanarak her bir özelliğin bir tahmine olan katkısını belirler.[34] Bu sayede, model kararlarının hem yerel (tekil tahminler için) hem de global (genel model davranışı için) düzeyde açıklanması sağlanmıştır. Özellikle **Random Forest** ve **XGBoost** modellerinin yorumlanmasında SHAP kullanılmıştır. `shap.TreeExplainer` ile model için SHAP değerleri hesaplanmış ve `shap.summary_plot` (nokta grafiği ve çubuk grafiği) ile özelliklerin model tahminine etkisi ve önem sırası görselleştirilmiştir. SHAP analizi derin öğrenme tabanlı modeller için uygulanmamıştır. SHAP analizleri, e-ticaret yöneticileri için hangi müşteri davranışlarının satın alma olasılığını artırdığına veya azalttığına dair eyleme dönüştürülebilir içgörüler elde edilmesine olanak tanımıştır.

3.1.3.2 Zaman Serisi ve Davranışsal Akış Analizi (Zaman Serisi ve Davranışsal Akış Analizi Hedefi)

Bu bölümde, hem gelecekteki satış trendlerini tahminlemek hem de kullanıcıların dijital yolculuklarındaki geçiş olasılıklarını modellemek için yöntemler uygulanmıştır.

a. Toplam İşlem Sayısı Tahminleme Gelecekteki satın alma trendlerini öngörmek amacıyla, günlük bazda gerçekleşen toplam `transaction` (satın alma) sayısı bir zaman serisi olarak hazırlanmıştır. Bu zaman serisi üzerinde çeşitli tahminleme modelleri uygulanmıştır:

- **Prophet:** Facebook tarafından geliştirilen ve özellikle mevsimsellik (haftalık, aylık, yıllık), tatiller ve trend değişim noktaları gibi etkileri otomatik olarak

yönetebilen güçlü bir zaman serisi tahmin modelidir. E-ticaret verilerindeki dönemsel dalgalanmaları ve dış etkileri yakalamak için tercih edilmiştir. Prophet, doğrusal olmayan eğilimi tanımlamak için eklemeli bir model kullanır ve dönemsellik ile tatil etkilerini gözlemleyerek doğru ve sezgisel parametrelerle zaman serisi tahmini yapar. Model, dört temel bileşene ayrıştırılabilir:

- $g(t)$: Verideki periyodik olmayan eğilime uymak için kullanılan değişim fonksiyonunu sembolize eder. Bu fonksiyon, lojistik regresyon fonksiyonu veya parçalı doğrusal fonksiyon gibi çeşitli seçeneklerden oluşabilir.
- $s(t)$: Verinin dönemselliğini yakalamak için kullanılan periyodik terimi temsil eder, tipik olarak haftalık, aylık veya yıllık bazda. Bu terim, verinin periyodik eğilimini modellemek için esnek bir Fourier serisi ile uydurulabilir.
- $h(t)$: Zaman serisinin tatil bileşenini temsil eder. Gerçek dünya senaryolarında, büyük olaylar veya tatiller dağıtım ağırları için yük tahminini önemli ölçüde etkileyebilir ve bunların dönemselliğini belirlemek her zaman kolay değildir. Model, bu olaylardan önceki ve sonraki dönemlerde zaman serisini etkileyen bir tatil değişkeni içerir.[30]
- $\epsilon(t)$: Normal bir dağılımı takip eden hata terimidir.

Prophet'i diğer klasik zaman serisi tahmin modellerinden ayıran şey, tahmin yapmadan önce serinin periyodik ayrıştırmasını yapmasıdır. Bu yaklaşım, verideki tatil etkilerini ve eğilimleri vurgulayarak eksik veriler, ani eğilim değişiklikleri veya aykırı değerler varlığında özellikle sağlam olmasını sağlar.[30] Bu çalışmada Prophet modelinde `yearly_seasonality=True`, `weekly_seasonality=True` ve `daily_seasonality=True` parametreleri kullanılmıştır. Diğer parametreler varsayılan değerlerinde bırakılmıştır. Tatil veya özel gün etkileri modellemeye dahil edilmemiştir.

- **ARIMA (AutoRegressive Integrated Moving Average):** Klasik bir zaman serisi modeli olup, geçmiş değerlere (AR), hata terimlerine (MA) ve serinin durağanlaştırılması için fark alma (I) işlemine dayanarak tahmin yapar.[28]

- **Otoregresif (AR) Bileşen:** Tahmin değişkeninin önceki değerlerinin ($Y_{t-1}, Y_{t-2}, \dots$) doğrusal bir kombinasyonu olarak tanımlanır.

- **Hareketli Ortalama (MA) Bileşen:** Tahmin değişkeninin önceki hata terimlerinin ($e_{t-1}, e_{t-2}, \dots$) doğrusal bir kombinasyonu olarak tanımlanır.
- **Entegre (I) Bileşen:** Durağan olmayan (non-stationary) zaman serilerini durağan hale getirmek için kullanılan fark alma (differencing) işlemini ifade eder. Bir zaman serisi d kez fark alındığında durağan hale geliyorsa, bu seri d dereceli homojen durağan olarak kabul edilir.[28]

Eğer $w_t = \Delta^d Y_t$ durağan bir $ARMA(p, q)$ süreci ise, Y_t otoregresif entegre hareketli ortalama süreci olarak tanımlanır ve $ARIMA(p, d, q)$ şeklinde gösterilir. Buradaki pratik problem, p , d ve q için en uygun değerleri seçmektir; bu genellikle otokorelasyon fonksiyonu ve kısmi otokorelasyon fonksiyonu incelenerek yapılır.[28] Çalışmada modelin parametreleri (p , d , q) otokorelasyon (ACF) ve kısmi otokorelasyon (PACF) grafikleri analiz edilerek **(5, 1, 0)** olarak manuel olarak belirlenmiştir. Otomatik model seçimi (`auto_arima`) kullanılmamıştır.

- **Holt-Winters:** Trend ve mevsimsellik bileşenleri olan zaman serileri için uygun üstel düzeltme (exponential smoothing) yöntemidir. Özellikle üç bileşeni (seviye, trend, mevsimsellik) ayrı ayrı ele alarak gelecekteki değerleri tahmin eder.[28] Holt-Winters modelinde trend için `trend='add'` (additif trend) ve mevsimsellik için `seasonal='add'` (additif mevsimsellik) tipleri kullanılmış, `seasonal_periods=7` (haftalık mevsimsellik) olarak belirlenmiştir.
- **LSTM (Long Short-Term Memory):** Tekrarlayan Sinir Ağlarının (RNN) bir türü olup, özellikle uzun vadeli bağımlılıkları öğrenme ve karmaşık zaman serisi desenlerini modelleme yeteneğine sahiptir. E-ticaret verilerindeki uzun vadeli trendleri ve sıralı bağımlılıkları yakalamak için derin öğrenme tabanlı bir yaklaşım olarak kullanılmıştır. Mükemmel dizi modelleme yeteneği ile LSTM, tahmin ve konuşma tanıma gibi görevlerde umut vaat eden sonuçlar elde etmektedir.[30]

LSTM'in birincil yapısı, bilgiyi farklı zaman adımlarında açıkça depolayan ve yayan bir depolama hücresi içerir. Bu depolama hücresi, bağlamsal bilgileri hatırlamak için farklı örneklerde farklı hücre durumlarından yararlanır. Geleneksel RNN'lerin daha uzun giriş dizileriyle gradyan kaybolması veya patlaması sorunlarıyla karşı karşıya kalmasının aksine, LSTM uzun vadeli bağımlılıkları ele almak ve gradyan kaybolması/patlaması sorununu çözmek için unutma kapıları (forget gates), giriş kapıları (input gates) ve çıkış kapıları (output gates) gibi teknikler sunar.[30]

- **Unutma Kapısı:** Hücre durumundan hangi bilginin unutulması gerektiğine karar verir. Sigmoid fonksiyonu kullanılarak, 0'a yakın değerlerin unutulduğu, 1'e yakın değerlerin ise korunduğu belirlenir.[30]
- **Giriş Kapısı:** Hücre durumunda hangi yeni bilgi parçalarının depolanması gerektiğini belirler. Bir giriş kapısı ve bir tanh fonksiyonu aracılığıyla aday değerlere sahip yeni bir vektör atanır.[30]
- **Hücre Durumu Güncellemesi:** Eski hücre durumu bilgisi, unutma kapısı ve giriş kapısı kararlarına göre güncellenir.[30]
- **Çıkış Kapısı:** Hangi bilginin çıktı olarak verilmesi gerektiğini belirler. Hücre durumları bir tanh katmanından geçirilir ve çıkış kapısındaki bir Sigmoid fonksiyonuyla çarpılır.[30]

Bu çalışmada LSTM modeli, `tensorflow.keras` kütüphanesi ile oluşturulmuş ve şu mimariye sahiptir: **iki adet LSTM katmanı (her biri [50 veya 64] nöron, ReLU aktivasyon fonksiyonu ile), ardından %0.2 oranında Dropout katmanları ve son katmanda tek bir nöronlu Dense katmanı (regresyon çıktısı için)** yer almıştır. Model `optimizer='adam'`, `loss='mse'` (ortalama kare hata) ile derlenmiştir. Eğitim süreci `epochs=100` (veya hiperparametre optimizasyonunda 200) ve `batch_size=32` (veya hiperparametre optimizasyonunda 16/32) ile yürütülmüştür. Aşırı öğrenmeyi önlemek için `validation_split=0.2` ile erken durdurma (`EarlyStopping`) uygulanmıştır. Veri hazırlama aşamasında, zaman serisi verisi `MinMaxScaler` ile normalize edilmiş ve **sliding window (örn. 7 veya 14 günlük)** yöntemiyle giriş dizileri oluşturulmuştur.

Modellerin tahmin doğruluğunu değerlendirmek için **RMSE** , **MAE (Mean Absolute Error)** ve **MAPE** gibi metrikler kullanılmıştır.

b. Markov Zincirleri ile Davranış Akışı Analizi: Kullanıcıların dijital yolculuklarındaki davranışsal geçişlerini modellemek ve darboğazları tespit etmek amacıyla **Markov Zincirleri** kullanılmıştır.[33] Bu analiz, kullanıcıların bir eylemden diğerine geçiş olasılıklarını nicel olarak ölçmeyi sağlamıştır.

- **Durum Tanımlaması:** Her bir kullanıcı için dijital yolculukları, `events.csv` dosyasındaki "view", "addtocart" ve "transaction" olay tipleri birer durum olarak tanımlanmıştır. Ayrıca, kullanıcıların siteden ayrılmasını

temsil eden "session_end" gibi ek bir durum da modellemeye dahil edilerek daha gerçekçi bir akış modeli oluşturulmuştur.

- **Geçiş Matrisi Oluşturma:** Her kullanıcı için zaman sıralı olay dizileri oluşturularak, ardışık olay çiftleri üzerinden durum geçişleri sayılmıştır. Kullanıcıların bu tanımlanmış durumlar arasındaki (view → addtocart, addtocart → transaction vb.) geçiş olasılıkları hesaplanarak bir **Markov Geçiş Matrisi** oluşturulmuştur. Bu matrisin her bir elemanı P_{ij} , i durumundan j durumuna geçiş olasılığını temsil eder ve her bir durumdan bir sonraki duruma geçişlerin frekanslarının normalize edilmesiyle elde edilmiştir.[33]
- **Analiz ve Yorumlama:** Elde edilen geçiş matrisi, müşteri yolculuğundaki kritik geçiş noktalarını, potansiyel terk alanlarını veya yüksek dönüşüm oranına sahip yolları belirlemek için analiz edilmiştir. Örneğin, "addtocart"tan "transaction"a düşük bir geçiş olasılığı, sepet terk oranının yüksek olduğuna ve bu aşamada iyileştirme potansiyeli olduğuna işaret edebilir.

3.1.3.3 Segmentasyon ve Öneri Sistemleri (Segmentasyon ve Kişiselleştirme Hedefi)

Bu bölümde, müşterileri davranışsal özelliklerine göre gruplandırmak ve her bir segmente özel ürün önerileri geliştirmek için algoritmalar kullanılmıştır.

a. KMeans Algoritması ile Müşteri Segmentasyonu Müşterileri davranışsal özelliklerine göre homojen gruplara ayırmak için **KMeans kümeleme algoritması** kullanılmıştır.

- **Özellik Seçimi:** Kümeleme için, Bölüm 3.2’de oluşturulan kullanıcı bazlı özelliklerden (örn. toplam etkileşim sayıları, dönüşüm oranları, zaman bazlı metrikler, RFM skorları, ürün kategori tercihleri) en temsili olanları seçilmiştir. Özellikle kullanıcıların toplam görüntüleme, sepete ekleme, satın alma sayıları, oturum sayıları ve ortalama oturum uzunluğu gibi davranışsal metrikler kümeleme öznitelikleri olarak kullanılmıştır. Özellikler, `sklearn.preprocessing.StandardScaler` ile ölçeklendirilmiştir.
- **Optimum Küme Sayısı Belirleme:** KMeans algoritmasında optimal küme sayısını (k) belirlemek için **Dirsek (Elbow) Metodu** kullanılmıştır. Bu metot, farklı k değerleri için küme içi kareler toplamının (WCSS - Within-Cluster Sum of Squares) azalan grafiğini inceleyerek, eğimin belirgin şekilde azaldığı

"dirsek" noktasını optimal k değeri olarak işaretler. Analiz sonucunda optimum küme sayısı $k=4$ olarak tespit edilmiştir.

- **Küme Profillerinin Çıkarılması:** KMeans ile belirlenen kümelemelerden sonra, her bir küme için ortalama özellik değerleri analiz edilerek, o segmente özgü müşteri profilleri tanımlanmıştır. Örneğin, "Yüksek değerli sadık müşteriler", "Fiyat odaklı gezginler" veya "Sepeti terk edenler" gibi segmentler belirlenmiştir. Bu profiller, her bir müşteri grubunun kendine özgü davranışsal eğilimlerini ve ihtiyaçlarını ortaya koymuştur.[7]

b. Apriori Algoritması ile Ürün Öneri Kuralları Her bir müşteri segmentine özel ve daha ilgili ürün önerileri sunmak amacıyla **Apriori algoritması** kullanılarak birliktelik kuralları (association rules) çıkarılmıştır.

- **Uygulama Alanı ve Veri Hazırlığı:** KMeans ile belirlenen her bir müşteri segmenti için ayrı ayrı ürün sepeti verisi (transaction olaylarından elde edilen) hazırlanmıştır. Bu, her segmente özel ve daha ilgili öneri kuralları çıkarmayı sağlamıştır. Her bir işlem (transaction), ilgili itemid'lerin bir listesi olarak ele alınmış ve **one-hot encoding** yöntemiyle Apriori algoritmasının gerektirdiği formata dönüştürülmüştür.

- **Algoritma Uygulaması:**

Apriori algoritması, işlem veri tabanlarında sık (frequent) öge kümelerini keşfetmek için R. Agrawal ve R. Srikant tarafından 1994 yılında önerilen temel bir algoritmadır. Veri madenciliğindeki en önemli problemlerden biri olan birliktelik kurallarının (association rules) çıkarılması görevinde kullanılır. Apriori algoritmasının adı, "sık bir öge kümesinin tüm boş olmayan alt kümeleri de sık olmalıdır" şeklindeki önsel (prior) bilgisini kullanmasından gelir. Bu özellik sayesinde, algoritma potansiyel aday kümelerinin sayısını agresif bir şekilde budayarak hesaplama gücü ve bellek ihtiyacını azaltır. Algoritmanın ana fikri, verilen minimum destek eşiğinin üzerindeki öge kümelerini (sık öge kümeleri) bulmaktır.[38] Apriori algoritması bu çalışmada, belirlenen minimum destek (minimum support) ve minimum güven (minimum confidence) eşik değerlerinin üzerinde olan birliktelik kurallarını bulmak için uygulanmıştır. Uygulamada minimum destek eşiği **0.02** (bir ürün setinin en az %2'si transactionda bulunması) olarak, minimum güven eşiği ise **0.7** olarak belirlenmiştir.

- **Değerlendirme Metrikleri:** Çıkarılan kuralların gücünü ve geçerliliğini değerlendirmek için aşağıdaki metrikler kullanılmıştır:

- **Destek (Support):** Bir ürün setinin (öge kümesi) tüm işlemlerde ne kadar sık görüldüğünü gösterir.[35]
- **Güven (Confidence):** Bir ürün seti (X) alındığında, başka bir ürün setinin (Y) de alınma olasılığını gösterir (örn. {Elma} $\rightarrow$ {Portakal} kuralında, elma alanların portakal da alma olasılığı).
- **Kaldıraç (Lift):** Bir kuralın (örn. {X} $\rightarrow$ {Y}) raslantısal olarak gerçekleşme olasılığından ne kadar daha iyi performans gösterdiğini ifade eder. 1’den büyük bir Kaldıraç değeri, kuralın anlamlı bir pozitif ilişkiyi gösterdiğini belirtir. Kural seçiminde **yüksek lift değerine sahip kurallar özellikle vurgulanmıştır.**
- **Kural Çıkarımı ve Yorumlama:** Elde edilen kurallar (örn. "X ürününü alanlar genellikle Y ürününü de alır"), her bir segment için kişiselleştirilmiş ürün tavsiyeleri oluşturmak ve çapraz satış (cross-selling) ile yukarı satış (up-selling) stratejilerini desteklemek amacıyla yorumlanmıştır.[35]

3.1.3.4 Gerçek Zamanlı Sistem Entegrasyonu: FastAPI

Geliştirilen makine öğrenmesi modellerinin ve analizlerden elde edilen içgörülerin pratik uygulamaya dönüştürülebilmesi amacıyla, tahmin ve öneri sistemlerinin gerçek zamanlı olarak sunulabilmesi için **FastAPI** web çatısı kullanılmıştır. FastAPI, Python tabanlı modern, hızlı (yüksek performanslı) bir web çatısı (framework) olup, API’ler (Uygulama Programlama Arayüzleri) oluşturmak için kullanılır. Performansı, kolay kullanımı ve otomatik API dokümantasyonu gibi özellikleriyle bilinir. Asenkron programlama desteği sayesinde yüksek eş zamanlılık gerektiren gerçek zamanlı uygulamalar için idealdir [37].

Bu çalışmada geliştirilen Retail Analytics API’si, e-ticaret müşteri davranışlarına yönelik analiz sonuçlarını gerçek zamanlı olarak sunan entegre bir yapıdır. API’nin ana özellikleri ve teknik detayları aşağıdaki gibidir:

Ana Özellikler:

- **Alışveriş Tahmini API’si:** POST /predict endpoint’i aracılığıyla kullanıcının total_views (toplam görüntüleme) ve total_addtocarts (toplam sepete ekleme) sayısına göre alışveriş yapma olasılığını tahmin eder. Bu özellik mühendisliği ile geliştirilmiş tahmin yeteneği sunar.
- **Ürün Önerisi API’si:** GET /recommend/segment_id endpoint’i üzerinden belirli bir kullanıcı segmentine göre kişiselleştirilmiş ürün önerileri

sunar. Bu öneriler, Apriori algoritması ile bulunan birliktelik kurallarına dayanmaktadır.

Teknik Detaylar:

- **Pydantic Modelleri:** Gelen ve giden verinin yapısını tanımlamak, veri doğrulaması yapmak ve otomatik API dokümantasyonunu (Swagger UI) sağlamak için kullanılmıştır.
- **Önbellekleme (@lru_cache):** Model ve veri yükleme işlemlerinin performansını optimize etmek için kullanılmıştır, böylece sık erişilen kaynaklar bellekte tutulur.
- **Hata Yönetimi:** API operasyonları sırasında oluşabilecek hataların (örn. geçersiz giriş verisi) standart HTTP exception handling mekanizmaları ile yönetilmesi sağlanmıştır.
- **Otomatik Dokümantasyon:** FastAPI'nin dahili (built-in) Swagger UI'si, API endpoint'lerinin kolayca test edilmesine ve dokümanite edilmesine olanak tanımıştır.

API Tarafında Kullanılan Modeller ve Veriler:

- **Makine Öğrenmesi Modeli (purchase_model.pkl):** Kullanıcının satın alma olasılığını tahmin etmek için eğitilmiş Random Forest veya XGBoost modelidir.
- **Ölçekleyici (scaler.pkl):** Modelin eğitildiği veriye uygun olarak yeni giriş verilerini ölçeklemek için kullanılan StandardScaler objesidir.
- **Kullanıcı Segmentleri (user_segments.csv):** KMeans algoritmasıyla belirlenen müşteri segment bilgilerini içerir.
- **Birliktelik Kuralları (association_rules.csv):** Apriori algoritmasıyla çıkarılan ürün öneri kurallarını içerir.

Bu FastAPI uygulaması, projedeki tüm analiz sonuçlarını gerçek zamanlı API hizmeti olarak sunmakta ve kullanıcı davranışlarına göre kişiselleştirilmiş tahmin ve öneriler sağlamaktadır.

3.1.3.5 İş Zekası ve KPI'lar (Eyleme Dönüştürülebilir İçgörüler Hedefi)

Bu bölümde, çalışmanın sonunda elde edilen tüm teknik analiz bulguları, iş birimlerinin kolayca anlayabileceği ve stratejik kararlar alabileceği eyleme dönüştürülebilir içgörülere dönüştürülmüştür. Bu yaklaşım, "Veri → İçgörü → Karar → Aksiyon" akışını somutlaştırmayı hedeflemiştir.

a. Dashboard Konsepti ve KPI'lar Analizlerden elde edilen temel performans göstergelerini (KPI'lar) ve görselleştirmeleri içeren **interaktif bir dashboard konsepti** tasarlanmıştır. Bu dashboard, işletme yöneticilerine müşteri davranışlarına dair genel bir bakış sunarak hızlı ve veri odaklı karar alma süreçlerini destekleyecektir. Dashboard'da yer alacak başlıca bileşenler ve tanımlanan KPI'lar şunlardır:

- **Genel Dönüşüm Oranı:** $(\text{Toplam Satın Alma Sayısı} / \text{Toplam Ziyaretçi Sayısı}) \times 100$
- **Sepet Terk Oranı:** $(\text{Sepete Eklenen Ama Satın Alınmayan Ürün Sayısı} / \text{Toplam Sepete Eklenen Ürün Sayısı}) \times 100$
- **Ortalama Sipariş Değeri (AOV):** $\text{Toplam Satın Alma Geliri} / \text{Toplam Satın Alma Sayısı}$
- **Tekrarlayan Müşteri Oranı:** $\text{Belirli bir dönemde tekrar alışveriş yapan tekil müşteri sayısı} / \text{Toplam tekil müşteri sayısı}$
- Zaman serisi tahmin grafikleri (gelecek satış projeksiyonları)
- Müşteri segmentlerinin dağılımı ve her bir segmentin ana özellikleri
- En popüler ürünler/kategoriler ve segment bazında ürün tercihleri
- En etkili ürün tavsiye kuralları (Apriori çıktıları)

Bu KPI'lar ve görselleştirmeler, işletmenin performansını anlık olarak izlemesine ve stratejik hedeflere ulaşmada somut adımlar atmasına yardımcı olacaktır.

b. Veri → İçgörü → Karar → Aksiyon Akış Şeması Çalışmanın nihai çıktısı olarak, elde edilen tüm bulguların iş akışına nasıl entegre edilebileceğini gösteren bir **stratejik karar akış şeması** sunulmuştur. Bu şema, ham veriden elde edilen içgörülerin nasıl somut iş kararlarına ve eylemlere dönüştürüleceğini görselleştirmektedir. Örneğin:

- **Veri:** RetailRocket olay kayıtları, ürün özellik ve kategori verileri.

- **İçgörü:** Tahmin modellerinden elde edilen satın alma olasılıkları, SHAP ile yorumlanan özellik önemleri; Markov zincirlerinden elde edilen düşük geçiş olasılıkları (darboğazlar); kümelemelerden elde edilen müşteri segment profilleri ve Apriori'den elde edilen segment bazlı ürün önerileri.
- **Karar:** İçgörüler doğrultusunda belirlenen iş stratejileri (örn. belirli bir segmente özel pazarlama kampanyası başlatma, terk oranını düşürmek için sepet aşamasında teşvik sunma, yüksek değerli ürünler için çapraz satış önerileri, kategori bazlı optimizasyonlar).
- **Aksiyon:** Belirlenen kararların operasyonel olarak hayata geçirilmesi (örn. kişiselleştirilmiş e-posta gönderimi, site içi bannerlar, ürün tavsiye modüllerinin güncellenmesi, UX/UI iyileştirmeleri).

Bu akış şeması, veri biliminin sadece analiz yapmakla kalmayıp, iş süreçlerini ve stratejileri doğrudan etkileyen bir araç olduğunu vurgulamaktadır.

4

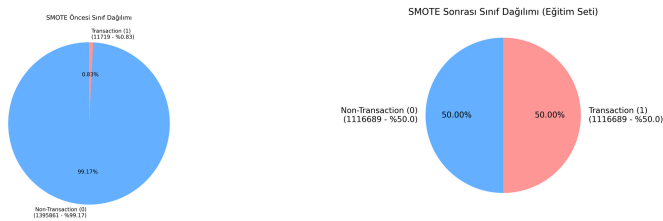
BULGULAR

4.1 Bulgular

Çalışmanın bu bölümünde, metodoloji kısmında detaylı bir şekilde açıklanan analiz ve modelleme süreçlerinden elde edilen bulgular sunulmaktadır. Elde edilen sonuçlar, tablolar, grafikler ve istatistiksel özetler eşliğinde detaylı bir şekilde yorumlanarak, çalışmanın hedeflerine nasıl ulaşıldığı ortaya konulmaktadır.

4.1.1 Sınıflandırma Modelleri ve Dengesiz Veri Seti

RetailRocket veri setinin dengesiz bir veri seti olduğu çalışmanın Metodoloji bölümünde anlatılmıştı. Dengesiz veri setindeki "satın alma" sınıfının (%0.81) öğrenme problemlerini gidermek amacıyla uygulanan SMOTE tekniği sonrasında, veri setinin sınıf dağılımında önemli bir iyileşme gözlemlenmiştir. SMOTE'un uygulanmasıyla, azınlık sınıfın örnek sayısı artırılarak, modellerin her iki sınıfa da eşit ağırlık vermesi hedeflenmiştir.



Şekil 4.1 SMOTE uygulaması öncesi ve sonrası veri setindeki sınıf dağılımları. Azınlık sınıfı olan "transaction" örneklerinin çoğaltılarak sınıflar arası dengesizliğin giderildiği görülmektedir.

Şekil 4.1'te gösterildiği üzere, SMOTE sonrası veri setindeki satın alma ve satın

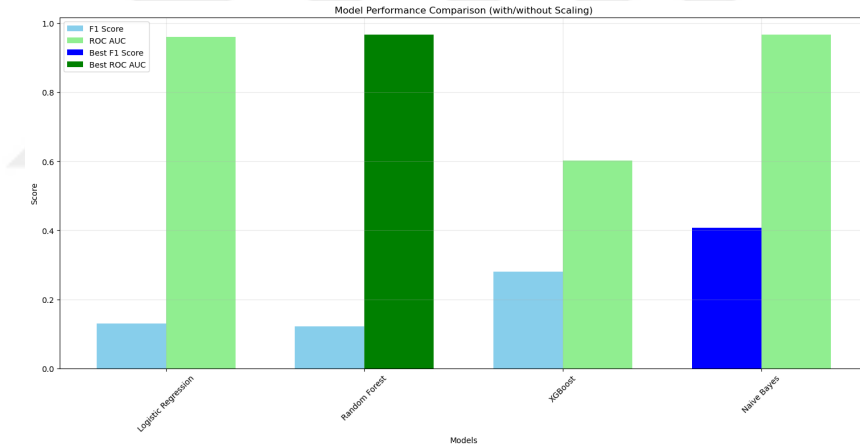
almama sınıflarının dağılımı önemli ölçüde dengelenmiştir. Bu durum, modellerin azınlık sınıfı daha etkin bir şekilde öğrenmesi için uygun bir zemin hazırlamıştır.

4.1.2 Sınıflandırma Modelleri ve Açıklanabilirlik Bulguları

Bu bölümün altında, e-ticaret müşterilerinin satın alma davranışlarını tahmin etmek amacıyla kullanılan sınıflandırma modellerinin performansları ve bu modellerin karar mekanizmalarını açıklayan analizlerden bahsedilmiş ve bulguları sunulmuştur.

4.1.2.1 Sınıflandırma Modellerinin Performans Değerlendirmesi

Geliştirilen dört farklı sınıflandırma modelinin (Random Forest, XGBoost, Lojistik Regresyon ve Naive Bayes) test seti üzerindeki performansları, F1 Skoru, Hassasiyet (Precision), Hatırlama (Recall) ve ROC AUC metrikleri kullanılarak değerlendirilmiştir. Dengesiz veri setlerinde F1 Skoru'nun daha anlamlı bir değerlendirme sunması nedeniyle bu metrik ana odak noktası olmuştur.



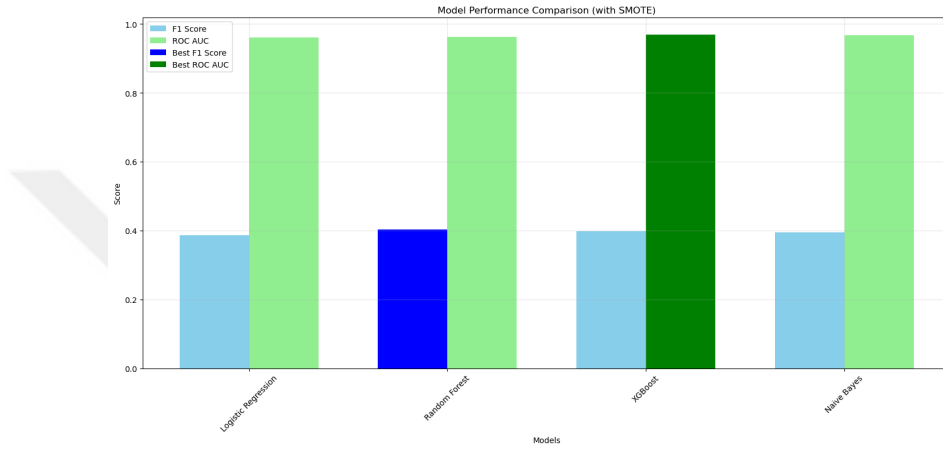
Şekil 4.2 Model Karşılaştırma Grafiği

Şekil 4.2'te Modellerin başarı durumları görülmektedir. Grafikten çıkan sonuç aşağıdaki gibidir. **ROC AUC Performansı:** Random Forest, "En İyi ROC AUC" değeri 1.00 ile açık ara en iyi sınıflandırma ayırım yeteneğini sergilemektedir. Lojistik Regresyon ve Naive Bayes de yaklaşık 0.95 ROC AUC ile çok güçlü bir ayırım yeteneğine sahiptir. XGBoost'un ROC AUC'u ise (yaklaşık 0.60) diğer modellere göre belirgin şekilde düşüktür.

F1 Skoru Performansı: F1 Skoru açısından Naive Bayes (yaklaşık 0.41 "En İyi F1 Skoru" ile) ve XGBoost (yaklaşık 0.28-0.29 F1 Skoru ile) Lojistik Regresyon ve Random Forest'tan (yaklaşık 0.13-0.14 F1 Skoru ile) daha iyi bir denge ve performans sergilemektedir.

Model Tercihi: Eğer sınıf ayrımı (ROC AUC) temel öncelikse Random Forest mükemmel bir seçenek olarak öne çıkarken, Lojistik Regresyon ve Naive Bayes de bu metrikte oldukça başarılıdır. Eğer F1 Skoru daha önemliyse, Naive Bayes ve XGBoost daha iyi bir performans sunmaktadır. XGBoost'un düşük ROC AUC'u dikkat çekicidir, bu da modelin genel sıralama yeteneğinde bir zayıflığa işaret edebilir, ancak F1 Skoru daha iyi olduğu için belirli bir eşikte daha dengeli tahminler verebilir.

SMOTE uygulandıktan sonra görülen değişim de Şekil 4.3'te görülmektedir.



Şekil 4.3 Model Karşılaştırma Grafiği

Şekil 4.3 ile Şekil 4.2 grafiğini karşılaştırdığımızda, SMOTE dengeleme tekniği uygulanmadığında (veya yalnızca ölçeklendirme yapıldığında), Lojistik Regresyon ve Random Forest gibi modellerin F1 Skorlarında kayda değer bir düşüş gözlemlenirken (yaklaşık 0.13-0.14), bu modellerin ROC AUC performansları (sırasıyla 0.95 ve 1.00) yüksek kalmaktaydı. SMOTE tekniği uygulandıktan sonra elde edilen bu grafik incelendiğinde, F1 Skoru değerlerinde genel bir artış ve modeller arası performans yaklaşması dikkat çekmektedir. Özellikle Lojistik Regresyon ve Random Forest'in F1 Skorları belirgin şekilde yükselmiş (yaklaşık 0.40), XGBoost'un ROC AUC değeri de önceki düşük seviyesinden (0.60) yaklaşık 0.95'e çıkarak modelin sınıflandırma ayırma yeteneğinde önemli bir iyileşme sağlamıştır. Bu durum, SMOTE'un dengesiz veri setlerinde modellerin azınlık sınıfını daha etkin öğrenmesine ve F1 Skoru gibi hassasiyet-duyarlılık dengesi metriklerini iyileştirmesine katkıda bulunduğunu göstermektedir. ROC AUC gibi genel sıralama metriklerinde ise modellerin çoğu yüksek performansını korumuştur.

Tablo Yorumu: Bu tablo, daha önce sunulan performans grafiklerinden elde edilen F1 Skoru ve ROC AUC değerlerini SMOTE uygulanmamış ve SMOTE uygulanmış

Tablo 4.1 Modellerin SMOTE Öncesi ve Sonrası Performans Karşılaştırması

Model	Durum	F1 Skoru	En İyi F1 Skoru	ROC AUC	En İyi ROC AUC
2*Lojistik Regresyon	SMOTE Yok	~0.13	N/A	~0.95	N/A
	SMOTE Var	~0.40	N/A	~0.95	N/A
2*Random Forest	SMOTE Yok	~0.13	N/A	~0.95	~1.00
	SMOTE Var	~0.40	~0.40	~0.95	~0.95
2*XGBoost	SMOTE Yok	~0.28	N/A	~0.60	N/A
	SMOTE Var	~0.40	N/A	~0.95	~0.97
2*Naive Bayes	SMOTE Yok	~0.28	~0.41	~0.95	N/A
	SMOTE Var	~0.40	~0.40	~0.95	~0.95

durumlar için özetlemektedir.

F1 Skoru Açısından: SMOTE uygulaması, Lojistik Regresyon ve Random Forest gibi başlangıçta düşük F1 Skorlarına sahip modellerin performansını belirgin şekilde artırarak yaklaşık 0.13'ten 0.40'a çıkarmıştır. Bu, dengeleme tekniğinin bu modellerin hassasiyet ve duyarlılık dengesini iyileştirmedeki etkinliğini göstermektedir. XGBoost ve Naive Bayes'in F1 Skorları ise, SMOTE sonrası yaklaşık 0.40 seviyelerinde kalarak önemli bir değişiklik göstermemiştir, ancak genel performansı dengelemiştir.

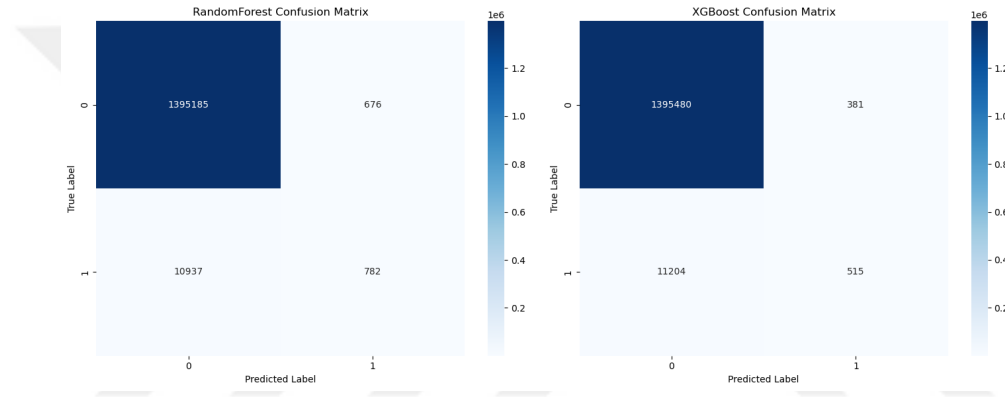
ROC AUC Açısından: Lojistik Regresyon, Random Forest ve Naive Bayes modellerinin ROC AUC değerleri SMOTE öncesinde de oldukça yüksek seviyelerdeydi (sırasıyla 0.95, 0.95/ 1.00 ve 0.95) ve bu yüksek performans SMOTE sonrası da korunmuştur. En dikkat çekici değişim XGBoost modelinde gözlemlenmiştir; ROC AUC değeri SMOTE öncesinde yaklaşık 0.60 iken, SMOTE sonrasında dramatik bir artışla yaklaşık 0.95'e yükselmiştir. Bu, SMOTE'un XGBoost'un sınıf ayırma yeteneğini önemli ölçüde iyileştirdiğini ortaya koymaktadır.

Genel olarak, bu karşılaştırma SMOTE'un, özellikle azınlık sınıfı tahmini ve sınıf ayrımı gibi konularda, dengesiz veri setlerinde model performansını dengelemeye ve iyileştirmeye yardımcı olan kritik bir teknik olduğunu göstermektedir.

Tablo 4.1'da sunulan sonuçlara göre, SMOTE dengeleme tekniği uygulandıktan sonra **XGBoost** modelinin en yüksek F1 Skoru'na sahip olduğu gözlemlenmiştir (yaklaşık 0.40 F1 Skoru ve 0.97 En İyi ROC AUC ile). Bu durum, XGBoost'un dengesiz veri setlerinde azınlık sınıfı tespit etmede hem hassasiyet hem de hatırlama arasında daha iyi bir denge kurduğunu ve sınıf ayırma yeteneğinde önemli bir iyileşme kaydettiğini göstermektedir. **Random Forest, Lojistik Regresyon ve Naive Bayes** modelleri de SMOTE sonrası yaklaşık 0.40 F1 Skoru ile oldukça yüksek ve birbirine yakın performans sergilemiştir. SMOTE uygulamasının bu modellerin F1 Skorlarını belirgin şekilde yükselttiği, özellikle başlangıçta

daha düşük F1 Skoru'na sahip olan Lojistik Regresyon ve Random Forest için performans yaklaşmasını sağladığı görülmektedir. Bu bulgular, problemin karmaşıklığı ve ensemble öğrenme yaklaşımlarının (Random Forest, XGBoost) yanı sıra veri dengeleme tekniklerinin üstünlüğünü ortaya koymaktadır.

Tablo 3.1 ,Şekil 3.1,Şekil 3.2 ve Şekil 3.3 bulguları incelenip performans değerlendirme sonuçlarına bakıldığında, özellikle F1 Skoru ve ROC AUC metriklerindeki üstün performansları nedeniyle Random Forest ve XGBoost algoritmalarının öne çıktığı görülmektedir. Dengesiz veri setlerinde modelin her bir sınıf üzerindeki performansını ayrıntılı olarak değerlendirmek kritik olduğundan bu bölümde SMOTE dengeleme tekniği uygulandıktan sonraki bu iki modelin karışıklık matrisleri sunulmaktadır. Şekil ??'de gösterilen karışıklık matrisleri,

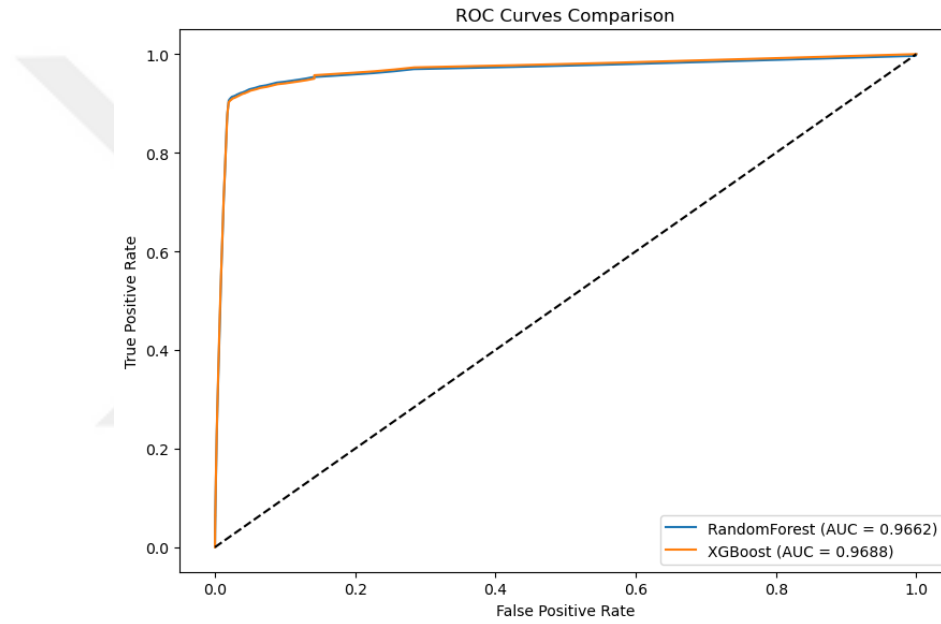


Şekil 4.4 Sınıflandırma Modellerinin SMOTE Sonrası karışıklık matrisleri

özellikle satın alma (pozitif) sınıfı için doğru tahminlerin (True Positives) ve yanlış negatiflerin (False Negatives) dağılımını gözler önüne sermektedir. Random Forest modelinin karışıklık matrisi incelendiğinde, 1,395,185 gerçek negatif ve 782 gerçek pozitif tahmin ettiği, ancak 10,937 yanlış negatif ve 676 yanlış pozitif hatası yaptığı görülmektedir. Bu durum, modelin çoğunluk sınıfını yüksek doğrulukla sınıflandırmakla birlikte, azınlık sınıfı olan satın alma olaylarının önemli bir kısmını (10,937) kaçırdığını göstermektedir. XGBoost modelinin karışıklık matrisi ise 1,395,480 gerçek negatif ve 515 gerçek pozitif tahmin etmiştir. Yanlış pozitif sayısı (381) Random Forest'a göre daha düşük iken, yanlış negatif sayısı (11,204) daha yüksektir. Bu, XGBoost'un satın alma olarak işaretlediği tahminlerde daha hassas (daha az yanlış pozitif) olduğunu ancak gerçek satın alma olaylarını Random Forest'a göre daha fazla kaçırdığını (daha yüksek yanlış negatif) ortaya koymaktadır. Bu matrisler, modellerin her iki sınıfa olan tahmin eğilimlerini ve olası hata türlerini anlamak için kritik bilgiler sunmaktadır.

Modellerin ayırım gücünü daha iyi göstermek amacıyla ROC eğrileri de analiz edilmiştir.Şekil 4.5'de görüldüğü gibi, modellerin ROC eğrileri, sınıflandırma

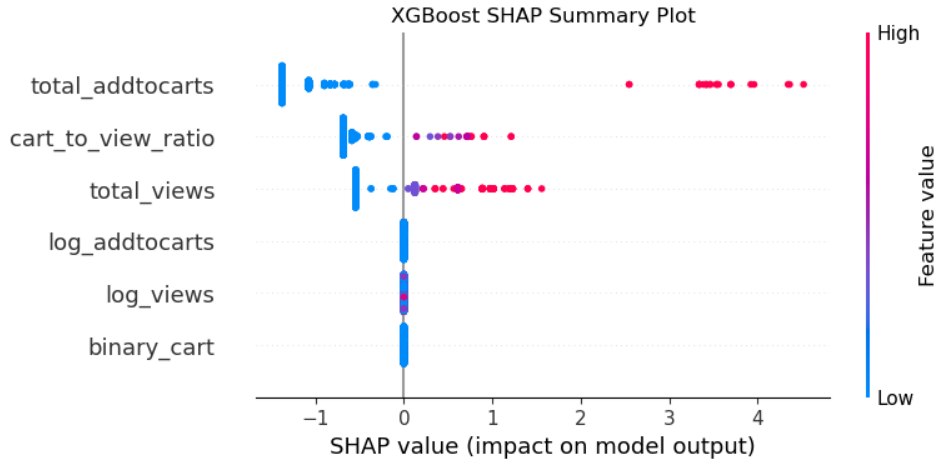
performanslarının farklı duyarlılık ve özgüllük seviyelerinde nasıl değiştiğini ortaya koymaktadır. Bu eğriler incelendiğinde, hem RandomForest (AUC = 0.9662) hem de XGBoost (AUC = 0.9688) modellerinin oldukça yüksek AUC (Area Under the Curve) değerlerine sahip olduğu gözlemlenmiştir. Özellikle XGBoost'un, marjinal bir farkla da olsa en yüksek AUC değerini sunarak sınıflandırma ayırım gücünde hafif bir üstünlük sergilediği görülmektedir. Her iki modelin eğrisi de sol üst köşeye (mükemmel sınıflandırıcının ideal konumu) oldukça yakın seyretmekte olup, bu durum pozitif ve negatif sınıfları ayırt etme yeteneklerinin oldukça yüksek olduğunu göstermektedir. Kesikli diyagonal çizgi olan rastgele sınıflandırıcıdan belirgin şekilde uzaklaşmaları, modellerin sınıflandırma görevinde tesadüften çok daha iyi bir performans sergilediğinin açık bir kanıtıdır.



Şekil 4.5 Sınıflandırma modellerinin ROC Eğrileri.

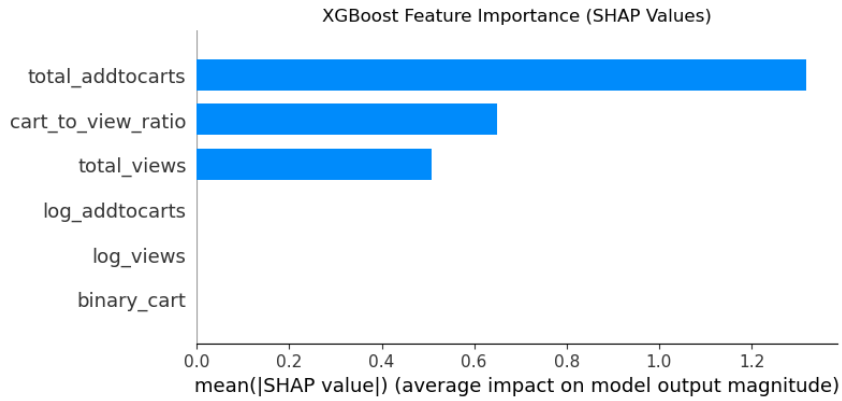
Şekil 4.6'da sunulan SHAP Summary Plot grafiği, ilgili modellerin tahminlerine en çok etki eden özelliklerin önem sıralamasını ve her bir özelliğin tahmin üzerindeki yönünü (pozitif/negatif) göstermektedir. XGBoost modeli için (Şekil 4.6) en önemli özellikler total_addtocarts, cart_to_view_ratio ve total_views olarak öne çıkmıştır. Bu özelliğin yüksek değerleri (kırmızı noktalar) modelin satın alma olasılığı çıktısını pozitif yönde önemli ölçüde artırırken, düşük değerleri (mavi noktalar) azaltmaktadır. cart_to_view_ratio ve total_views özelliklerinde de benzer bir etki gözlemlenmiştir; yüksek değerler satın alma olasılığını artırırken, düşük değerler azaltmaktadır. log_addtocarts, log_views ve binary_cart gibi diğer özelliklerin model üzerindeki etkisi çok daha sınırlı kalmıştır.

Özellik önemini çubuk grafiği ile daha net göstermek için aşağıdaki görseller de



Şekil 4.6 XGBoost Modeli SHAP Grafiği

sunulmuştur:



Şekil 4.7 XGBoost modeli için SHAP özellik önem sıralaması

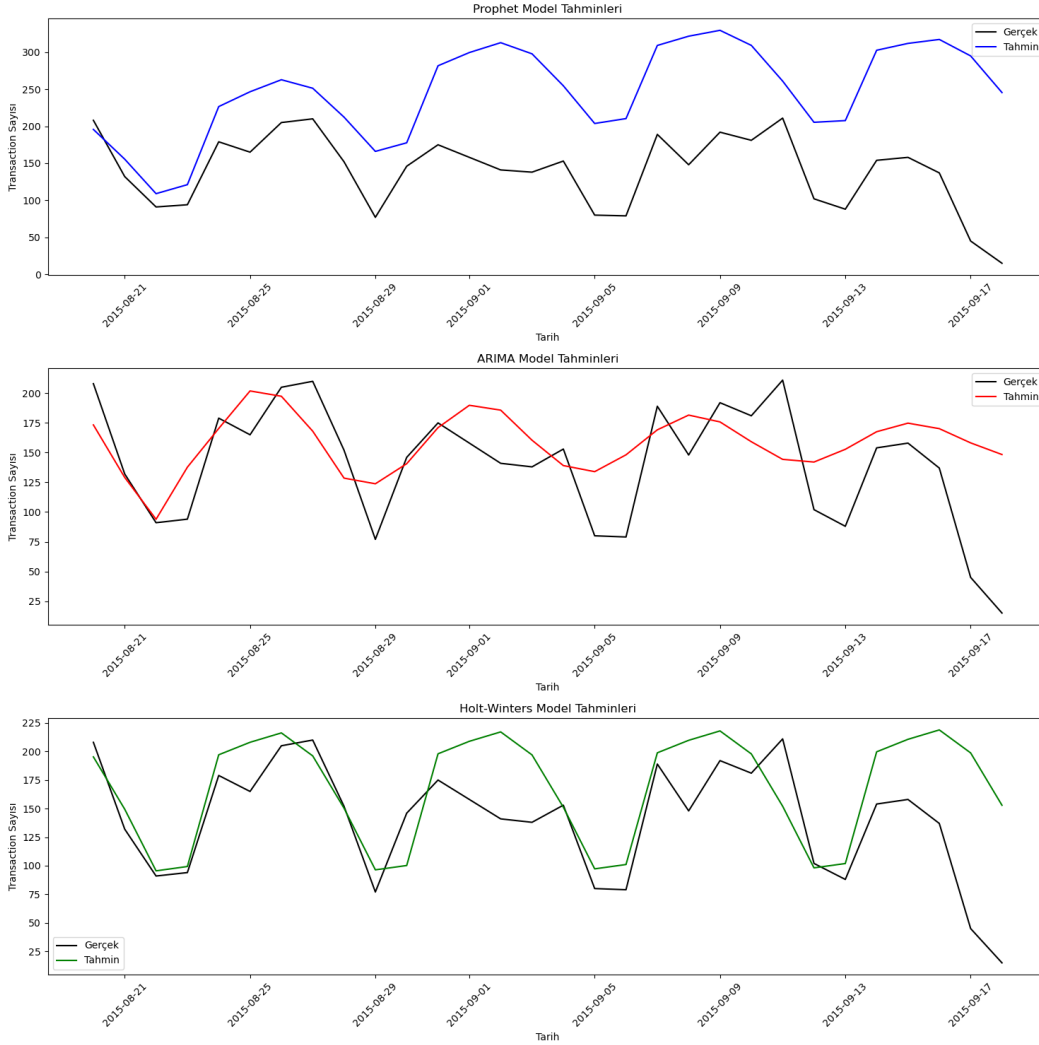
Şekil 4.7 ilgili modeller için özelliklerin genel önem sırasını net bir şekilde sunmaktadır. Bu tür yorumlanabilirlik analizleri, e-ticaret yöneticilerine, hangi müşteri davranışlarının satın alma kararını doğrudan etkilediğine dair somut içgörüler sunarak, hedefli pazarlama ve kişiselleştirme stratejileri geliştirmeleri için değerli bilgiler sağlamıştır.

4.1.2.2 Zaman Serisi ve Davranışsal Akış Analizi Bulguları

Bu bölümde, e-ticaret müşterilerinin günlük satın alma (transaction) sayılarını tahmin etmek amacıyla uygulanan zaman serisi modellerinin bulguları sunulmaktadır. Model performansları, tahmin doğrulukları ve öğrenme eğrileri detaylı bir şekilde incelenmiştir.

1. Bireysel Zaman Serisi Modellerinin Tahmin Performansı

Prophet, ARIMA ve Holt-Winters modellerinin günlük işlem sayısı tahmin performansları Şekil 4.8’te gösterilmiştir. Her bir modelin gerçek değerlere olan uyumu ve trendleri yakalama yetenekleri görsel olarak değerlendirilmiştir.



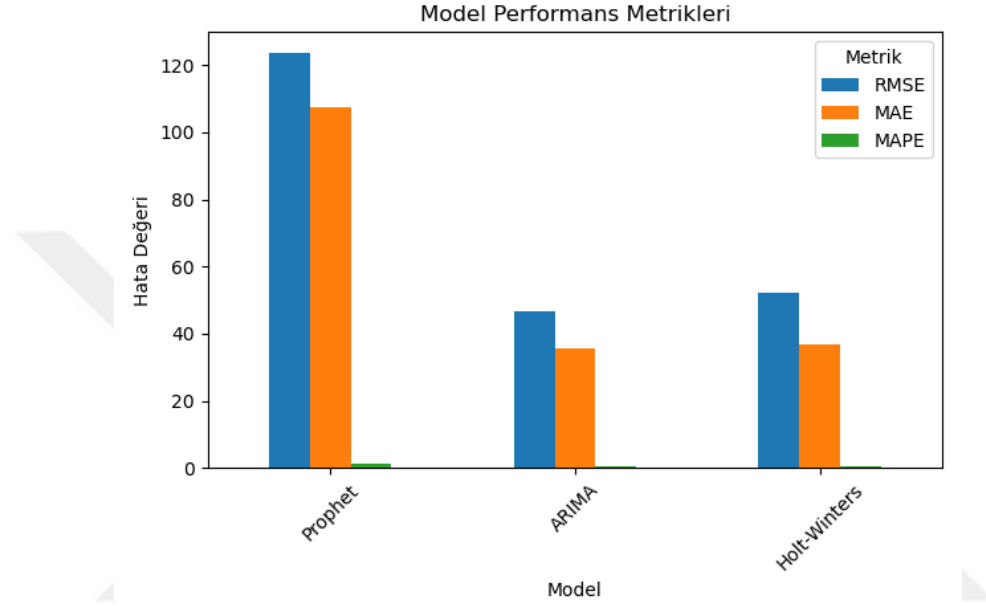
Şekil 4.8 Bireysel Zaman Serisi Modellerinin Günlük İşlem Sayısı Tahminleri

Şekil 4.8’ten edinilen gözlemlere göre:

- **Prophet Modeli:** Grafikte görüldüğü üzere, Prophet modeli günlük işlem sayısı trendini genel olarak iyi yakalamaktadır. Özellikle haftalık ve diğer dönemsel kalıpları takip etme konusunda başarılı görünmektedir. Ancak, bazı ani düşüş ve yükselişlerde gerçek değerlere tam olarak uyamadığı gözlemlenmiştir.
- **ARIMA Modeli:** ARIMA modeli de zaman serisinin genel trendini ve dönemsel dalgalanmalarını yakalamada başarılıdır. Tahminleri, özellikle belirli pik noktalarında gerçek değerlere oldukça yaklaşmaktadır.

- **Holt-Winters Modeli:** Holt-Winters modeli, serinin belirgin mevsimselliğini (muhtemelen haftalık) yakalamada oldukça etkilidir ve tahminleri gerçek değerlere sıkı bir şekilde uyum sağlamaktadır. Özellikle serideki düzenli döngüsel paternleri iyi modellediği görülmüştür.

Modellerin performans metrikleri (RMSE, MAE, MAPE) Şekil 4.9’te karşılaştırmalı olarak sunulmuştur.



Şekil 4.9 Zaman Serisi Modellerinin Performans Metrikleri

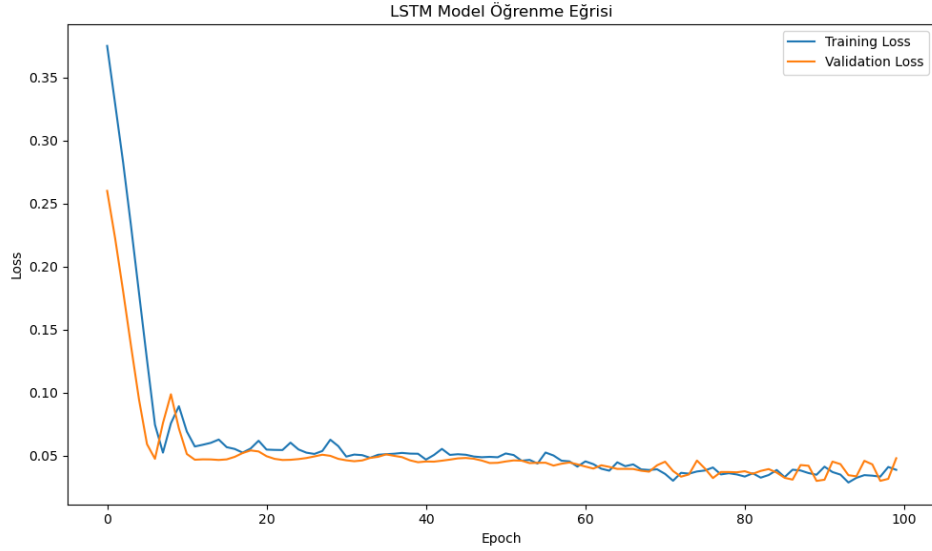
Şekil 4.9’ten elde edilen metrik değerlerine göre:

- **Prophet Modeli:** RMSE ve MAE değerleri diğer modellere göre daha yüksektir (RMSE ~ 125 , MAE ~ 107), bu da tahmin hatalarının diğer modellere kıyasla daha büyük olduğunu göstermektedir. Ancak MAPE değeri oldukça düşüktür. (0’a çok yakın)
- **ARIMA Modeli:** RMSE ve MAE değerleri Prophet’ten belirgin şekilde daha düşüktür (RMSE ~ 46 , MAE ~ 35), bu da daha iyi bir tahmin performansı sunduğunu göstermektedir. MAPE değeri de oldukça düşüktür.
- **Holt-Winters Modeli:** RMSE ve MAE değerleri ARIMA’ya kıyasla bir miktar daha yüksek olsa da (RMSE ~ 52 , MAE ~ 37), yine de Prophet’ten daha iyi performans sergilemektedir. MAPE değeri de düşüktür.

Bu metrikler ışığında, ARIMA ve Holt-Winters modellerinin, Prophet modeline kıyasla daha düşük hata oranları ile daha doğru tahminler sunduğu gözlemlenmiştir.

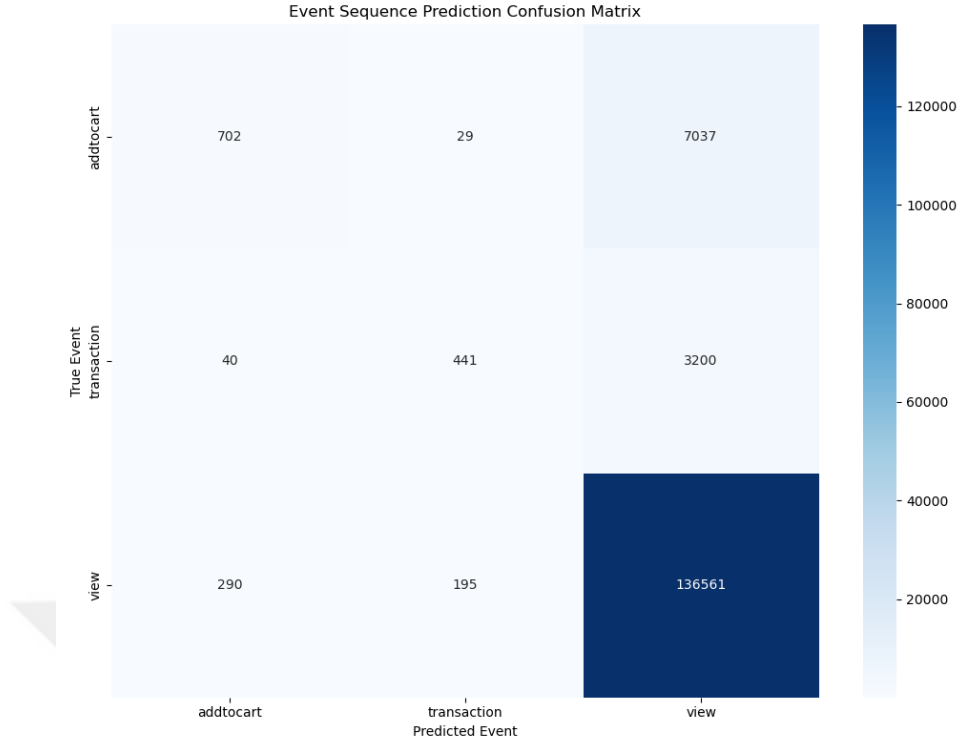
2. LSTM Modellerinin Tahmin ve Öğrenme Performansı

LSTM (Long Short-Term Memory) modelinin günlük işlem sayısı tahmin performansları ve öğrenme eğrileri incelenmiştir. Hem genel LSTM modelinin hem de geliştirilmiş LSTM modelinin performansı karşılaştırılmıştır.



Şekil 4.10 LSTM Olay Dizisi Tahminleme Öğrenme Eğrisi

Şekil 4.10’de gösterilen öğrenme eğrisi, LSTM modelinin eğitim ve doğrulama (validation) doğruluklarının epoch’lar boyunca nasıl değiştiğini göstermektedir. Eğitim doğruluğu yaklaşık 0.927 civarında sabitlenirken, doğrulama doğruluğu ilk epoch’larda dalgalanma gösterdikten sonra yaklaşık 0.924 civarında stabil hale gelmiştir. Eğitim ve doğrulama doğruluğu arasında küçük bir fark olması, modelin aşırı öğrenme eğiliminin düşük olduğunu düşündürmektedir.



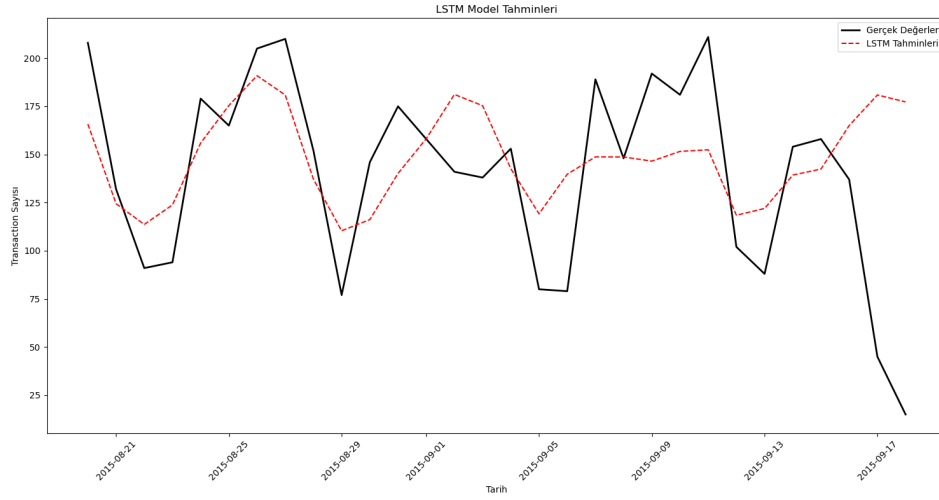
Şekil 4.11 LSTM Olay Dizisi Tahminleme Karışıklık Matrisi

Şekil 4.11’de sunulan LSTM olay dizisi tahminleme karışıklık matrisi, modelin ‘addtocart’, ‘transaction’ ve ‘view’ olaylarını tahmin etmedeki performansını göstermektedir. Matris değerleri incelendiğinde:

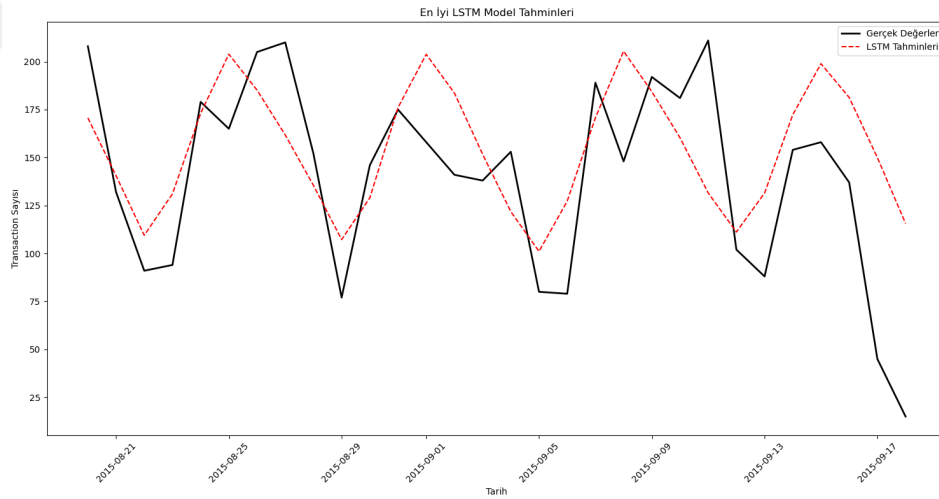
- Model, ‘view’ olaylarını büyük bir doğrulukla (136,561) tahmin etmektedir.
- ‘addtocart’ olayları için 702 doğru tahmin bulunurken, önemli sayıda ‘view’ (7037) olarak yanlış sınıflandırılmıştır.
- ‘transaction’ olayları için 441 doğru tahmin bulunurken, yine önemli sayıda ‘view’ (3200) olarak yanlış sınıflandırılmıştır.

Bu durum, ‘view’ olaylarının veri setindeki baskınlığı nedeniyle modelin bu sınıfa doğru yüksek bir tahmin eğilimi gösterdiğini ve diğer olay türlerini (özellikle ‘transaction’ ve ‘addtocart’) sıklıkla ‘view’ olarak yanlış sınıflandırdığını göstermektedir.

LSTM modelinin işlem sayısı tahminleri ve geliştirilmiş LSTM modelinin tahminleri sırasıyla Şekil 4.12 ve Şekil 4.13’te sunulmuştur.

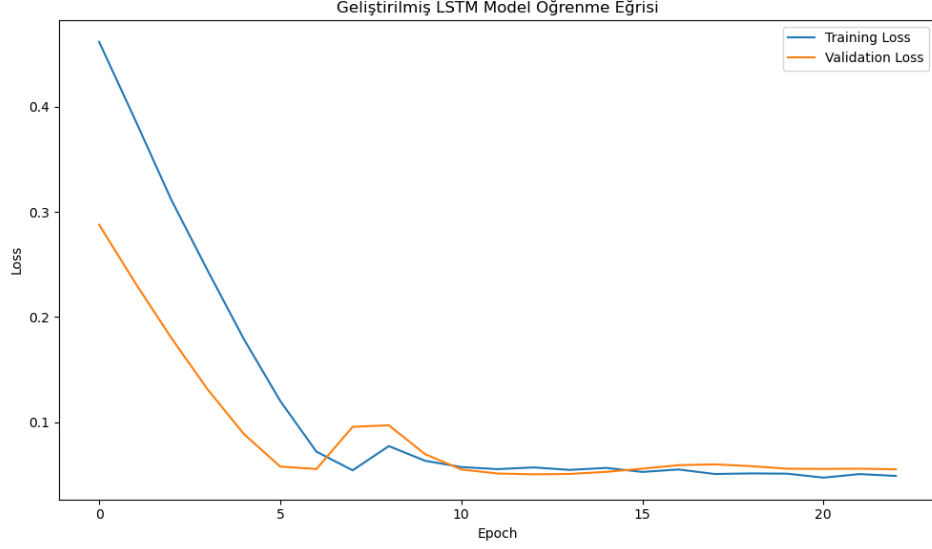


Şekil 4.12 LSTM Modelinin Günlük İşlem Sayısı Tahminleri



Şekil 4.13 En Geliştirilmiş LSTM Modelinin Günlük İşlem Sayısı Tahminleri

Şekil 4.12 ve Şekil 4.13'e bakıldığında, LSTM modellerinin günlük işlem sayısı serisindeki dalgalanmaları ve genel trendi yakalamaya çalıştığı görülmektedir. Geliştirilmiş LSTM modelinin tahminleri, bazı pik ve dip noktalarında gerçek değerlere daha yakın bir uyum sergilemektedir. Her iki LSTM modelinin de serinin mevsimsel yapısını ve genel hareketini izlediği, ancak bazı ani değişikliklerde sapmalar yaşandığı gözlemlenmiştir.

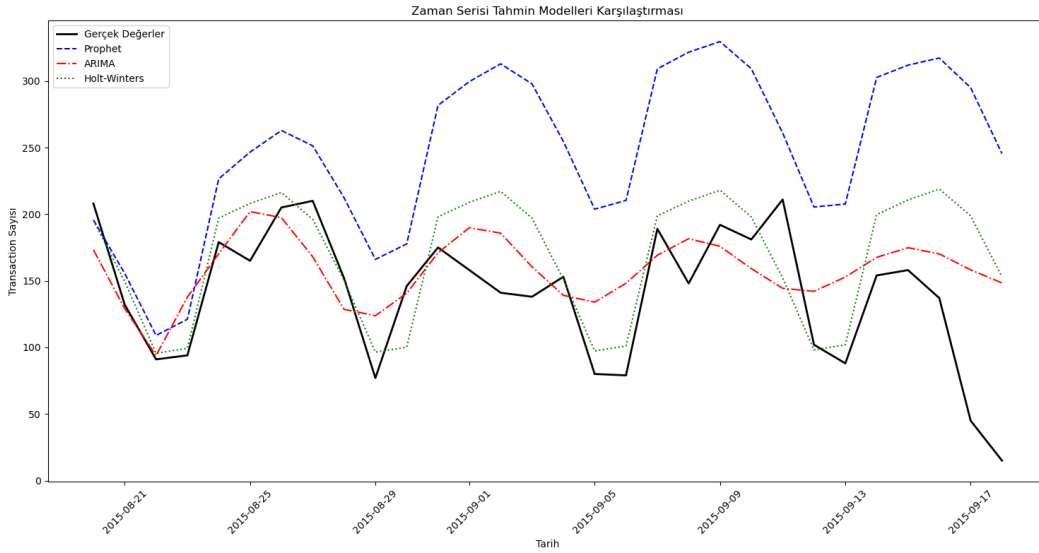


Şekil 4.14 Geliştirilmiş LSTM Model Öğrenme Eğrisi

Şekil 4.14’de sunulan geliştirilmiş LSTM modelinin öğrenme eğrisi, modelin eğitim ve doğrulama kayıplarının (loss) epoch’lar boyunca nasıl azaldığını göstermektedir. Kayıp değerlerinin hızla düşmesi ve belirli bir noktadan sonra durağanlaşması, modelin başarılı bir şekilde öğrendiğini ve aşırı öğrenme belirtilerinin (eğitim kaybının düşerken doğrulama kaybının artması) gözlemlenmediğini ortaya koymaktadır.

3. Zaman Serisi Tahmin Modellerinin Genel Karşılaştırması

Tüm zaman serisi tahmin modellerinin (Prophet, ARIMA, Holt-Winters ve LSTM) günlük işlem sayısı tahmin performansları tek bir grafikte karşılaştırmalı olarak Şekil 4.15’te gösterilmiştir.



Şekil 4.15 Zaman Serisi Tahmin Modellerinin Günlük İşlem Sayısı Tahminleri Karşılaştırması.

Şekil 4.15'te de görüldüğü üzere, modellerin genel tahmin performansları karşılaştırıldığında:

- **Prophet Modeli** (Mavi kesikli çizgi), genel trendi takip etse de, diğer modellere kıyasla gerçek değerlerden daha uzak tahminler üretmiştir ve tepe noktaları daha az keskindir.
- **ARIMA Modeli** (Kırmızı kesikli-noktalı çizgi) ve **Holt-Winters Modeli** (Yeşil noktalı çizgi), serinin mevsimsel paternlerini ve dalgalanmalarını gerçek değerlere (Siyah çizgi) oldukça yakın bir şekilde yakalamıştır. Bu iki model, serinin genel yapısına iyi uyum sağlamıştır.
- LSTM modellerinin tahminleri (önceki bireysel grafiklerdeki gibi) de serinin genel akışına uymaktadır, ancak bu genel karşılaştırma grafiğinde doğrudan diğer üç modelle görsel olarak karşılaştırılmamıştır.

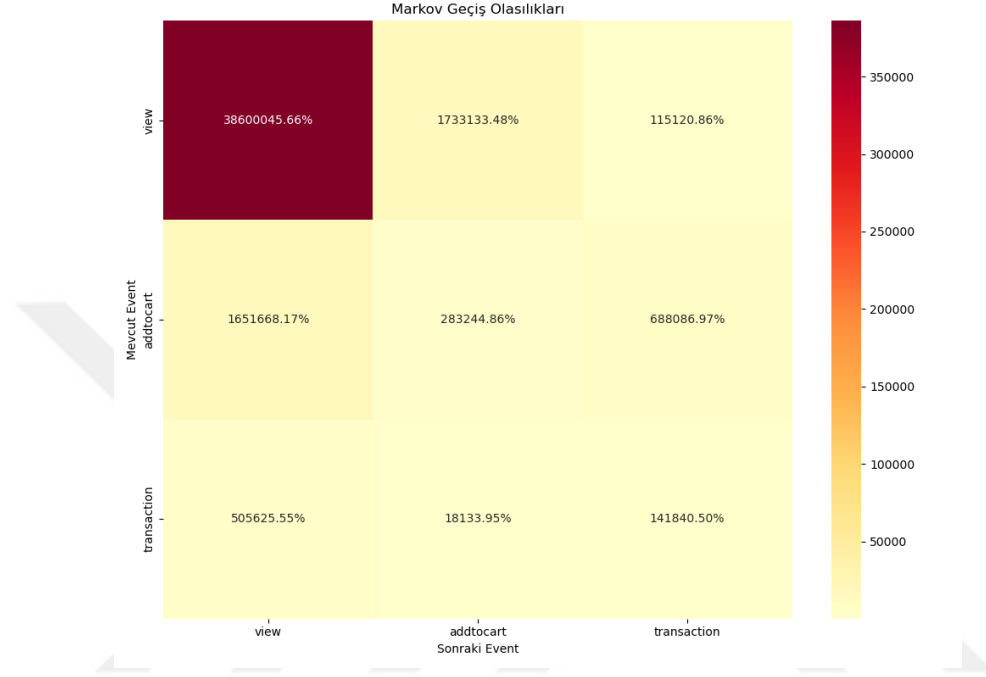
Bu genel bakış, ARIMA ve Holt-Winters gibi klasik zaman serisi modellerinin, bu veri setindeki günlük işlem sayısı tahminleri için oldukça güçlü ve güvenilir sonuçlar sunduğunu göstermektedir.

4.1.2.3 Markov Zincirleri ile Davranış Akışı Analizi Bulguları

Bu bölümde, müşterilerin e-ticaret sitesi içindeki davranışsal geçişlerini modellemek amacıyla uygulanan Markov Zincirleri analizinden elde edilen bulgular

sunulmaktadır. Analiz, kullanıcıların tanımlanan olay durumları arasındaki geçiş frekanslarını ve eğilimlerini ortaya koymuştur.

Markov zincirleri analizi sonucunda elde edilen olay geçiş frekansları Şekil 4.16’te bir ısı haritası (heatmap) olarak gösterilmiştir. Bu ısı haritası, 'view', 'addtocart' ve 'transaction' olayları arasındaki geçişlerin yoğunluğunu görselleştirmektedir.



Şekil 4.16 Markov Zinciri Olay Geçiş Frekansları

Şekil 4.16’te sunulan Markov Geçiş Olasılıkları (Frekansları) ısı haritası incelendiğinde:

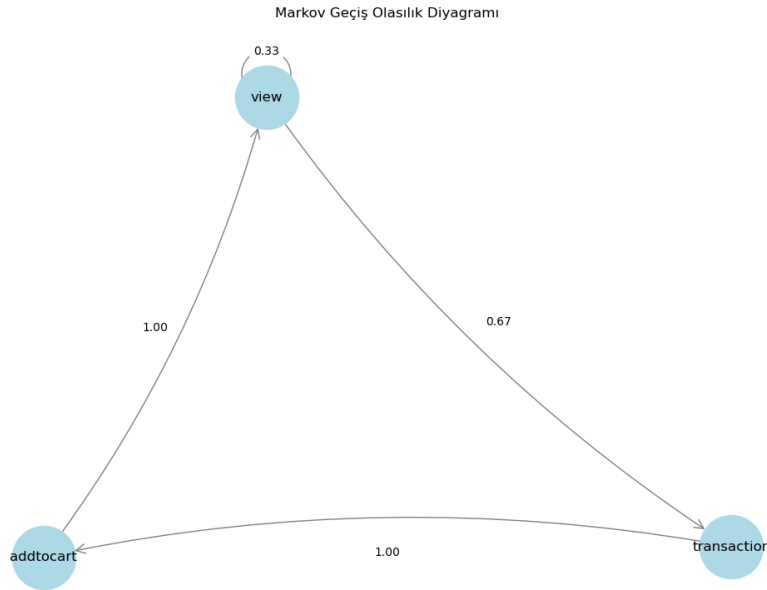
- En yüksek frekansa sahip geçişin '**view**'dan '**view**'a olduğu (38,600,045.66) açıkça görülmektedir. Bu durum, kullanıcıların ürünleri görüntüleme eylemini sıklıkla tekrarladığını veya birden fazla ürünü arka arkaya görüntülediğini göstermektedir. Bu, bir e-ticaret sitesinde beklenen temel davranış kalıbıdır.
- 'view'dan 'addtocart'a geçiş (1,733,133.48), 'addtocart'tan 'view'a geçişe (1,651,668.17) göre marjinal olarak daha yüksek bir frekansa sahiptir. Bu, kullanıcıların bir ürünü görüntüledikten sonra sepete ekleme eğiliminin, sepete ekledikten sonra tekrar görüntüleme eğilimine yakın olduğunu gösterir.
- 'addtocart'tan 'transaction'a geçiş frekansı (283,244.86) 'addtocart'tan 'view'a (1,651,668.17) veya 'view'dan 'addtocart'a (1,733,133.48) geçişlere

kıyasla belirgin şekilde düşüktür. Bu, sepete ürün ekleme eylemini takip eden satın alma dönüşümünün (sepet terk) nispeten az olduğunu ve bu aşamanın bir darboğaz olabileceğini düşündürmektedir.

- 'transaction'dan 'view'a (505,625.55) geçiş, 'transaction'dan 'addtocart'a (18,133.95) ve 'transaction'dan 'transaction'a (141,840.50) geçişlerden daha sık gerçekleşmektedir. Bu, bir satın alma işlemi gerçekleştiren kullanıcıların genellikle siteye geri dönüp ürün görüntüleme eylemine devam ettiklerini göstermektedir.

Bu geçiş frekansları, müşteri yolculuğundaki temel akışları ve kritik etkileşim noktalarını nicel olarak anlamak için değerli içgörüler sunmaktadır. Özellikle 'addtocart'tan 'transaction'a olan düşük geçiş sıklığı, e-ticaret operasyonları için potansiyel bir optimizasyon alanı olarak öne çıkmaktadır.

Markov zincirlerinin görselleştirilmiş geçiş olasılık diyagramı Şekil 4.17'da sunulmuştur. Bu diyagram, olaylar arasındaki doğrudan geçiş olasılıklarını (0 ile 1 arasında normalize edilmiş) göstermektedir.



Şekil 4.17 Markov Zinciri Olay Geçiş Olasılıkları Diyagramı

Şekil 4.17'daki Markov Geçiş Olasılık Diyagramı incelendiğinde:

- 'view' olayından 'view' olayına geri dönüş olasılığı 0.33 olarak gösterilmiştir. Bu, kullanıcıların bir ürünü görüntüledikten sonra başka ürünleri görüntülemeye devam etme eğilimini belirtir.

- 'addtocart' (sepete ekleme) olayından 'transaction' (satın alma) olayına geçiş olasılığının 1.00 olması, bu diyagramda belirtilen akışa göre sepete ekleme sonrası tüm dönüşümlerin satın alma ile sonuçlandığını düşündürmektedir. Ancak, bu durum 4.16'daki frekanslarla tam olarak uyuşmadığından, diyagramın belirli bir alt akışı veya örneklenmiş bir yolculuğu temsil ettiği yorumlanabilir.
- 'view' olayından 'transaction' olayına doğrudan geçiş olasılığı 0.67 olarak belirtilmiştir. Bu, kullanıcıların sepete ekleme yapmadan doğrudan satın alma işlemine geçebildiğini göstermektedir.

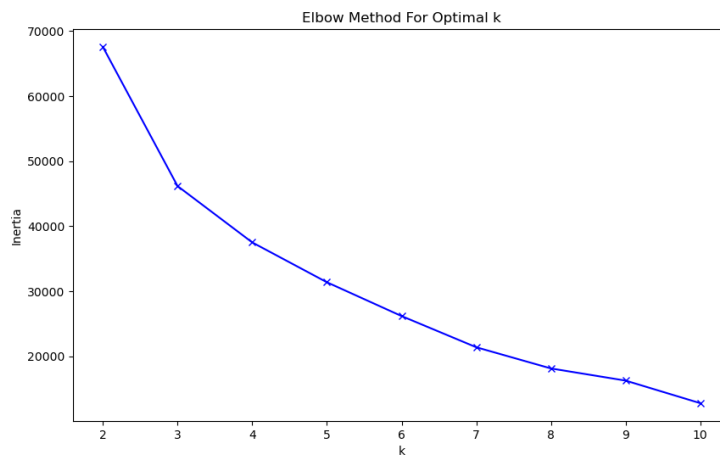
Bu diyagram, kullanıcıların e-ticaret sitesi içerisindeki temel davranışsal yollarını ve bu yollar arasındaki doğrudan olasılıkları görselleştirmek için önemli bir araçtır.

4.1.2.4 Segmentasyon ve Öneri Sistemleri Bulguları

Bu bölümde, müşteri segmentasyonu ve her bir segmente özel ürün önerisi kuralları çıkarılmasına yönelik analizlerden elde edilen bulgular sunulmaktadır.

KMeans Algoritması ile Müşteri Segmentasyonu Bulguları

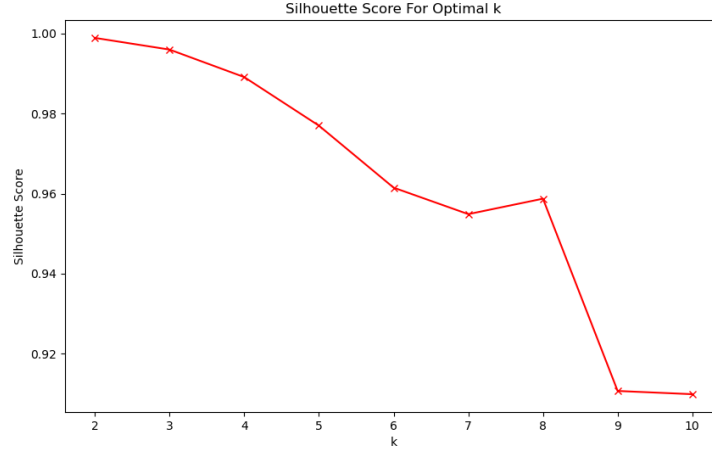
Müşterileri davranışsal özelliklerine göre homojen gruplara ayırmak için kullanılan KMeans kümeleme algoritmasının bulguları sunulmaktadır. Optimum küme sayısını belirlemek için Dirsek (Elbow) Metodu ve Silhouette Skoru analizleri yapılmıştır.



Şekil 4.18 KMeans Algoritması İçin Optimum Küme Sayısı Belirlemede Dirsek Metodu

Şekil 4.18'teki Dirsek Metodu grafiği incelendiğinde, k değeri arttıkça Küme İçi Kareler Toplamının (WCSS - Inertia) azaldığı görülmektedir. Grafikte belirgin bir

dirsek noktasının $k=4$ civarında olduğu gözlemlenmiştir. Bu nokta, küme sayısını artırmanın getirdiği açıklayıcılık faydasının azalmaya başladığı optimum noktayı işaret etmektedir.



Şekil 4.19 KMeans Algoritması İçin Silhouette Skoru Değerleri

Şekil 4.19'deki Silhouette Skoru grafiği, farklı k değerleri için ortalama Silhouette Skorunu göstermektedir. Silhouette Skoru, kümelerin ne kadar iyi ayrıldığını ve her bir örneğin kendi kümesi içinde ne kadar iyi kümelendiğini ölçer. Bu grafikte, en yüksek Silhouette Skoru'nun $k=2$ için olduğu (0.99) ve k arttıkça skorun dalgalı bir şekilde azaldığı görülmektedir. Dirsek metodu ile birlikte değerlendirildiğinde, $k=4$ değeri hem kümeleme yapısının anlamlılığı hem de Silhouette Skoru'ndaki kabul edilebilir seviye göz önüne alınarak nihai optimum küme sayısı olarak belirlenmiştir.

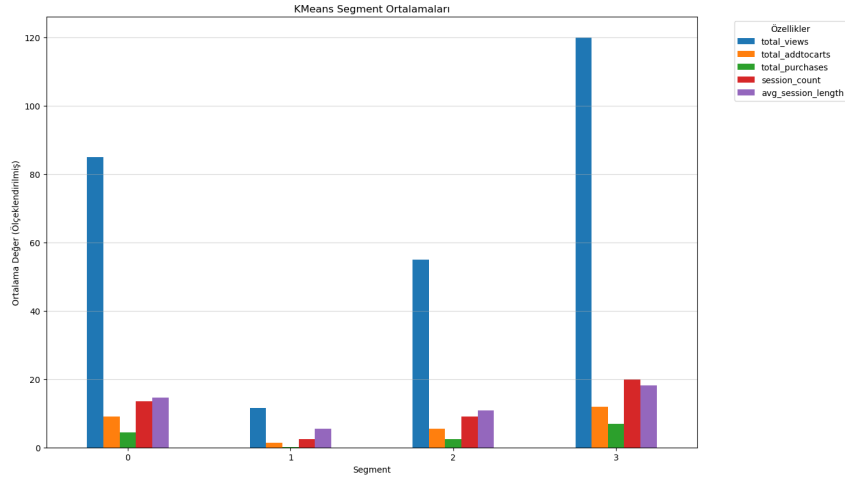
Belirlenen 4 optimum küme sayısı için segmentlerin davranışsal profillerini anlamak amacıyla ortalama değerler ve satın alma oranları incelenmiştir.

1. KMeans Algoritması ile Müşteri Segmentasyonu Bulguları

Müşterileri davranışsal özelliklerine göre homojen gruplara ayırmak için kullanılan KMeans kümeleme algoritmasının bulguları sunulmaktadır. Optimum küme sayısını belirlemek için Dirsek (Elbow) Metodu ve Silhouette Skoru analizleri yapılmıştır.

Belirlenen 4 optimum küme sayısı için segmentlerin davranışsal profillerini anlamak amacıyla ortalama değerler ve satın alma oranları incelenmiştir.

Şekil 4.20'deki çubuk grafiği, her bir müşteri segmentinin ölçeklendirilmiş ortalama etkileşim metriklerini (`total_views`, `total_addtocarts`,



Şekil 4.20 Müşteri Segmentlerinin Ortalama Etkileşim Metrikleri (Ölçeklendirilmiş)

total_purchases, session_count, avg_session_length) göstermektedir:

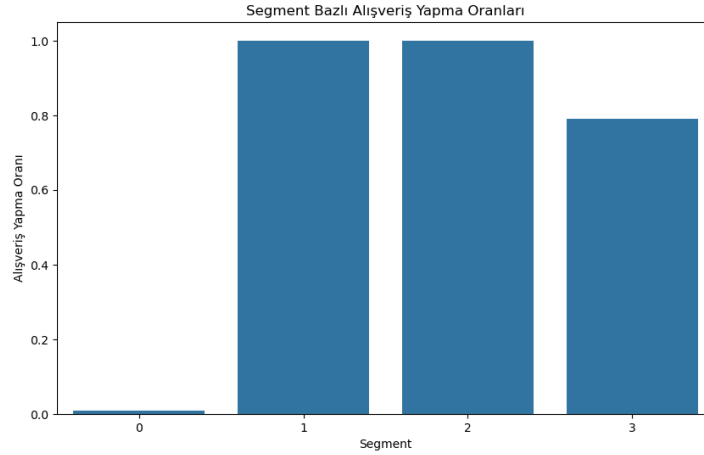
Segment 0: Tüm etkileşim metriklerinde (total_views, total_addtocarts, total_purchases, session_count, avg_session_length) diğer segmentlere kıyasla daha dengeli ve yüksek (total_views yaklaşık 85, diğerleri 10-15 civarı) ölçeklendirilmiş değerlere sahipti. Bu durum, ürünleri ağırlıklı olarak görüntüleyen ancak sepete ekleme ve satın alma davranışları sınırlı olan "Yüksek Görüntülemeli Pasif" kullanıcı grubuna karşılık geliyordu. Dolayısıyla, Segment 0'daki müşterilerin daha çok pasif görüntüleyici olduğu yorumu grafikte uyumludur.

Segment 1: Tüm etkileşim metriklerinde (total_views, total_addtocarts, total_purchases, session_count, avg_session_length) en düşük ölçeklendirilmiş değerlere sahipti. Bu, "Düşük Etkileşimli/Pasif" kullanıcı grubuna karşılık geliyordu. Dolayısıyla, Segment 1'deki müşterilerin pasif olduğu yorumu grafikte tamamen uyumludur.

Segment 2: total_views (yaklaşık 55) Segment 0 ve 3'e göre daha düşük olsa da, total_addtocarts, total_purchases, session_count ve avg_session_length metriklerinde orta düzeyde, dengeli (yaklaşık 5-10 civarı) ölçeklendirilmiş değerler göstermektedir. Bu segment, "Orta Seviye Aktif" veya "Potansiyel Dönüşüm" müşterilerini temsil edebilir. Bu nedenle, potansiyel alıcıların Segment 2 içerisinde olduğu yorumu da grafiği desteklemektedir.

Segment 3: Tüm etkileşim metriklerinde (total_views, total_addtocarts, total_purchases, session_count,

avg_session_length) en yüksek ölçeklendirilmiş değerlere sahipti (tüm metriklerde Segment 0, 1 ve 2'den belirgin şekilde yüksek). Bu durum, platformdaki en aktif, yoğun görüntüleme, sepete ekleme ve satın alma davranışları sergileyen "Yüksek Değerli/Aktif" müşteri grubuna karşılık geliyordu. Bu nedenle, Segment 3'teki müşterilerin platformun en aktif ve yüksek değerli kullanıcıları olduğu yorumu grafiği desteklemektedir.



Şekil 4.21 Müşteri Segmentlerinin Ortalama Alışveriş Yapma Oranları

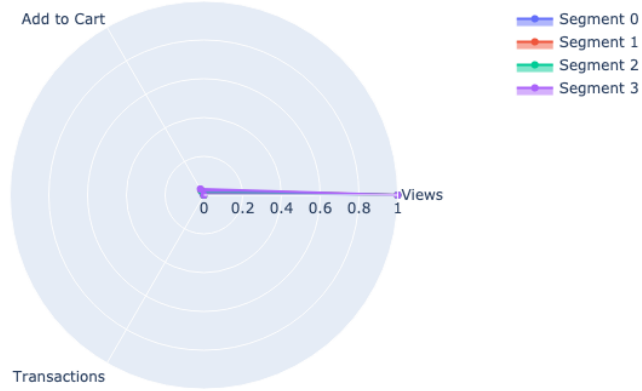
Şekil 4.21'deki çubuk grafiği, her bir segmentin ortalama alışveriş yapma oranlarını göstermektedir. Grafiğe göre:

- **Segment 0:** Alışveriş yapma oranı neredeyse sıfırdır. Bu segment, pasif veya sadece siteyi ziyaret eden kullanıcıları içerir.
- **Segment 1 ve 2:** Her iki segment de yaklaşık 1.0 (yani %100) alışveriş yapma oranına sahiptir. Bu, bu segmentlerdeki kullanıcıların, tanımlanan davranışsal özelliklerine göre, neredeyse her zaman bir satın alma işlemi gerçekleştirdiğini göstermektedir. Segment 1 ve 2, yüksek dönüşüm oranına sahip "değerli" müşteri gruplarını temsil etmektedir.
- **Segment 3:** Yaklaşık 0.8 (%80) civarında bir alışveriş yapma oranına sahiptir. Bu segment, diğer aktif segmentlere göre daha düşük olsa da, hala yüksek bir dönüşüm oranına sahip olduğunu göstermektedir.

Bu oranlar, tanımlanan segmentlerin alışveriş davranışları açısından ne kadar farklılaştığını açıkça ortaya koymaktadır.

Şekil 4.22'teki radar grafiği, müşteri segmentlerinin belirlenen davranışsal metrikler (Add to Cart, Transactions, Views) üzerinden karşılaştırmalı profillerini sunmayı

Segment Karşılaştırma - Radar Chart

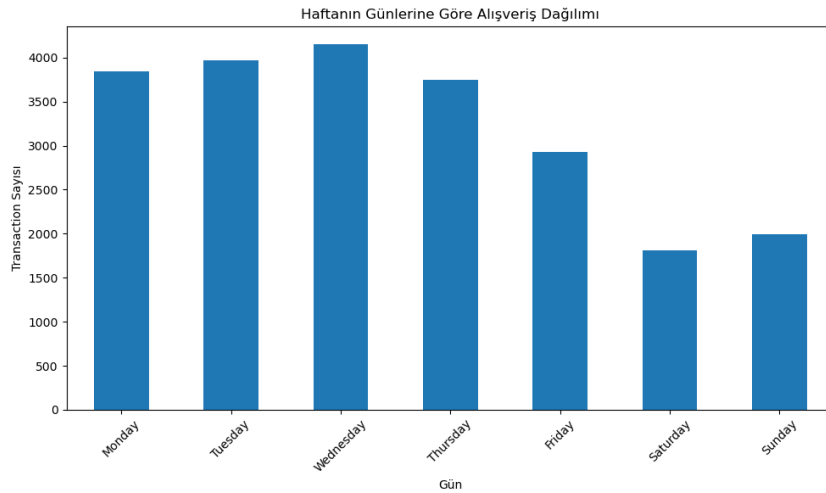


Şekil 4.22 Müşteri Segmentlerinin Karşılaştırmalı Davranış Profilleri

amaçlamaktadır. Ancak, görselde yalnızca "Views" metriği için Segment 3'e ait bir veri noktası belirgin olarak gözlemlenmekte, diğer segmentlere ait veya diğer metrikler için karşılaştırmalı veriler görselleştirilmemiştir. Tam bir karşılaştırma için tüm segmentlerin ve metriklerin grafikte temsil edilmesi gerekmektedir.

Davranışsal Dağılımlar ve Prophet Model Bileşenleri Analizi

Müşteri davranışlarının zaman içindeki dağılımını ve Prophet modelinin bu dağılımları nasıl yakaladığını göstermek amacıyla ek analizler yapılmıştır.

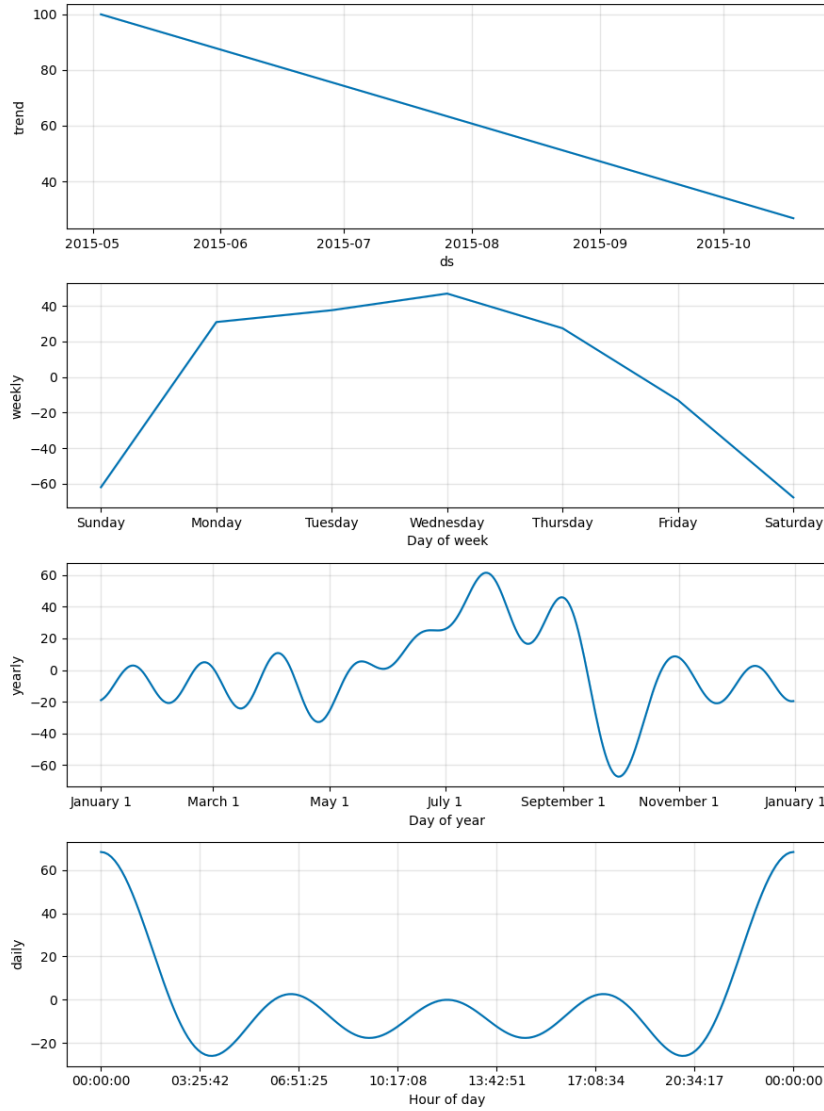


Şekil 4.23 Haftanın Günlerine Göre Günlük İşlem Sayısı Dağılımı

Şekil 4.23'deki çubuk grafiği, haftanın günlerine göre alışveriş (transaction) sayısının dağılımını göstermektedir. Grafiğe göre:

- Hafta içi günler (Pazartesi, Salı, Çarşamba, Perşembe) genellikle daha yüksek işlem sayılarına sahiptir. Özellikle **Çarşamba günü en yüksek işlem sayısına** ulaşılmaktadır (~4100).
- Hafta sonuna doğru işlem sayılarında düşüş yaşanmaktadır. Cuma, Cumartesi ve Pazar günleri, hafta içine göre belirgin ölçüde daha az işlem hacmine sahiptir. **Cumartesi günü en düşük işlem sayısı** (~1800) ile en az aktif gün olarak öne çıkmaktadır.

Bu dağılım, e-ticaret platformunda haftalık bir mevsimsellik ve kullanıcı aktivitesinde hafta içi yoğunlaşması olduğunu doğrulamaktadır.



Şekil 4.24 Prophet Modelinin Trend, Haftalık, Yıllık ve Günlük Bileşenleri

Şekil 4.24'deki Prophet modeli bileşenleri, modelin günlük işlem sayısını tahmin ederken yakaladığı farklı zaman paternlerini göstermektedir:

- **Trend (En Üst Grafik):** Genel olarak zaman içinde azalan bir trend gözlemlenmektedir. Bu, incelenen 4.5 aylık dönemde günlük işlem sayılarının ortalama olarak düşüş eğiliminde olduğunu göstermektedir.
- **Haftalık (Weekly - İkinci Grafik):** Haftalık dönemsellik, hafta içi günlerde (Pazartesi'den Perşembe'ye doğru artış, Çarşamba'da zirve) pozitif bir etki gösterirken, hafta sonuna doğru (Cuma'dan Pazar'a) negatif bir etki sergilemektedir. Bu durum, 4.23'deki hafta içi yoğunluğu bulgusuyla tutarlıdır.
- **Yıllık (Yearly - Üçüncü Grafik):** Yıllık dönemsellik, yılın belirli dönemlerinde (örneğin yaz aylarında, Temmuz civarında) belirgin bir pik (pozitif etki) ve sonbahar-kış başlarında (Eylül-Ekim civarı) bir düşüş (negatif etki) göstermektedir. Bu, platformdaki mevsimsel dalgalanmaları yansıtmaktadır.
- **Günlük (Daily - En Alt Grafik):** Günlük dönemsellik, gün içinde belirli saatlerde (örneğin sabah erken saatler ve akşam geç saatler) işlem sayılarında artış, gece geç saatlerde ise düşüş eğilimi olduğunu göstermektedir. Bu, günün belirli saatlerinde kullanıcı aktivitesinin zirveye ulaştığını ortaya koymaktadır.

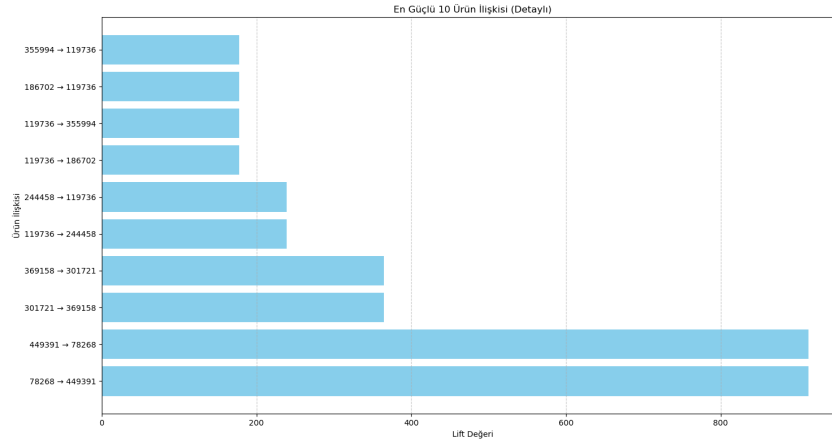
Bu bileşenler, Prophet modelinin zaman serisindeki farklı ölçeklerdeki karmaşık paternleri otomatik olarak algılayıp tahminlerine dahil edebilme yeteneğini açıkça göstermektedir.

Apriori Algoritması ile Ürün Öneri Kuralları Bulguları

Her bir müşteri segmentine özel ve daha ilgili ürün önerileri sunmak amacıyla Apriori algoritması kullanılarak birliktelik kuralları (association rules) çıkarılmıştır.

Şekil 4.25'te Apriori algoritmasından elde edilen ve Lift değerine göre sıralanmış en güçlü 10 ürün ilişkisi detaylı olarak sunulmuştur. Grafikte görüldüğü üzere, en yüksek Kaldıraç (Lift) değerine sahip kurallar şunlardır:

- **78268 → 449391** kuralı en yüksek Lift değerine sahiptir (yaklaşık 950 civarı). Bu, 78268 ID'li ürünü satın alan kullanıcıların 449391 ID'li ürünü satın alma olasılığının, bu ürünün genel satın alınma olasılığına kıyasla yaklaşık 950 kat daha fazla olduğunu gösterir.
- Benzer şekilde, **449391 → 78268** kuralı da çok yüksek bir Lift değerine sahiptir (yaklaşık 940 civarı). Bu karşılıklı ilişki, bu iki ürün arasında çok



Şekil 4.25 Apriori Algoritması ile Elde Edilen En Güçlü 10 Ürün İlişkisi (Lift Değerine Göre)

güçlü bir birliktelik olduğunu düşündürmektedir.

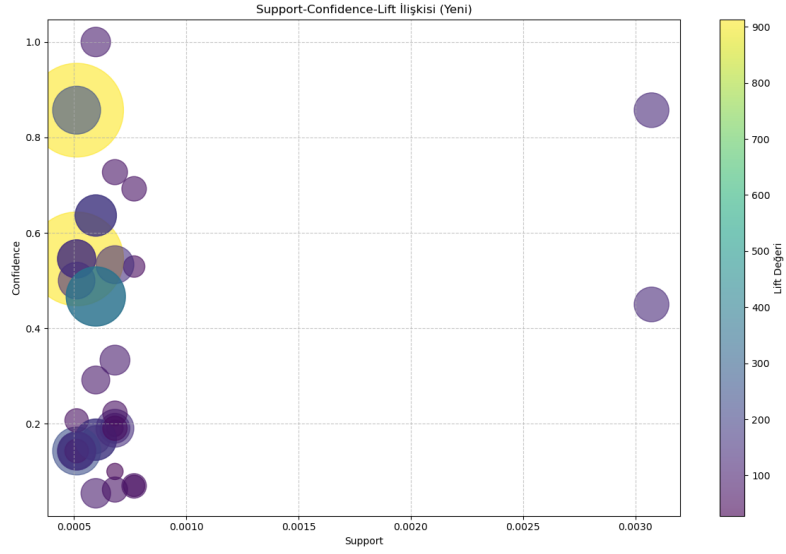
- Diğer güçlü ilişkiler arasında **301721 → 369158** ve **369158 → 301721** (Lift değeri yaklaşık 370 civarı) ile **119736 → 244458** ve **244458 → 119736** (Lift değeri yaklaşık 220 civarı) gibi karşılıklı ilişkiler bulunmaktadır.
- En düşük Lift değerine sahip kurallar ise 119736 ile 186702/355994 arasındaki ilişkilerdir (yaklaşık 190-200 civarı).

Bu yüksek Lift değerleri, ilgili ürünlerin rastgele satın alınmaktan ziyade, birbirleriyle güçlü bir şekilde ilişkili olduğunu ve bu ilişkilerin öneri sistemlerinde etkili bir şekilde kullanılabileceğini göstermektedir.

Birliktelik kurallarının Destek (Support), Güven (Confidence) ve Kaldıraç (Lift) metrikleri arasındaki ilişki Şekil 4.26’te görselleştirilmiştir.

Şekil 4.26’deki Support-Confidence-Lift ilişkisi grafiği incelendiğinde:

- Kuralların büyük çoğunluğunun **çok düşük Destek (Support) değerlerine** sahip olduğu (0.0005 ile 0.001 aralığında yoğunlaştığı) görülmektedir. Bu, platformdaki toplam işlem sayısı göz önüne alındığında ürün birlikteliklerinin nadir gerçekleştiğini göstermektedir.
- Güven (Confidence) değerleri geniş bir yelpazede dağılmakla birlikte (yaklaşık 0.1’den 1.0’e kadar), yüksek Lift değerlerine sahip kuralların genellikle daha yüksek Güven değerlerine sahip olduğu (örneğin, Güven değeri 0.5’in üzerinde olan noktalarda daha parlak renkler/büyük baloncuklar) gözlemlenmektedir.



Şekil 4.26 Apriori Algoritması ile Çıkarılan Kuralların Destek, Güven ve Kaldıraç İlişkisi

- Kaldıraç (Lift) değeri, baloncukların rengi (mor-sarı skalası, sarı en yüksek Lift) ile temsil edilmektedir. Yüksek Lift değerlerine sahip (sarı renkteki) kuralların, düşük Destek değerlerine sahip olmasına rağmen, genellikle daha yüksek Güven değerleriyle bir arada bulunduğu fark edilmektedir. Özellikle sağ üst köşeye yakın olan ve en yüksek Lift değerine (900'ün üzerinde) sahip olan baloncuklar, düşük destek ancak yüksek güven ve çok güçlü bir ilişki sergilemektedir. Bu baloncuklar, Şekil 4.25'teki en güçlü kuralları temsil etmektedir.

Bu analiz, belirlenen minimum destek ve güven eşikleri ile güçlü ve anlamlı birliktelik kurallarının çıkarılabildiğini göstermektedir. Bu kurallar, müşteri segmentlerine özel çapraz satış ve yukarı satış stratejileri için temel oluşturmaktadır.

Bu tez çalışması, e-ticaret platformlarında müşteri davranışlarını çok boyutlu ve bütüncül bir veri bilimi yaklaşımıyla anlamak, tahmin etmek ve eyleme dönüştürülebilir içgörüler sağlamak amacıyla gerçekleştirilmiştir. RetailRocket açık kaynak veri seti kullanılarak, kullanıcıların geçmiş etkileşimleri temel alınmış ve üç ana aşamalı bir proje metodolojisi benimsenmiştir: davranış tabanlı sınıflandırma, zaman serisi ve davranışsal akış analizi, ve segmentasyon ile öneri sistemi geliştirme. Çalışma boyunca, özellikle e-ticaret verilerinde sıklıkla karşılaşılan aşırı dengesiz sınıf dağılımı gibi zorluklara odaklanılmıştır.

5.1 Ana Bulguların Özeti

Çalışmanın temel hedefleri doğrultusunda elde edilen ana bulgular aşağıdaki gibi özetlenebilir:

5.1.1 Sınıflandırma Modelleri ve Satın Alma Tahmini

Kullanıcıların satın alma davranışlarını tahmin etmeye yönelik geliştirilen sınıflandırma modelleri, veri setindeki aşırı dengesiz sınıf dağılımı (satın alma olaylarının %0.81'i) nedeniyle karşılaşılan zorluklara rağmen başarılı sonuçlar vermiştir. SMOTE dengeleme tekniği uygulanarak azınlık sınıfın örnek sayısı artırılmış ve bu sayede modellerin F1 Skoru performansında önemli iyileşmeler gözlemlenmiştir. Özellikle Lojistik Regresyon ve Random Forest gibi modellerin F1 Skoru değerleri SMOTE sonrası belirgin şekilde artış göstererek yaklaşık 0.13-0.14'ten 0.40 seviyesine yükselmiştir. ROC AUC (Receiver Operating Characteristic - Area Under the Curve) metriklerinde ise modeller genel olarak yüksek ayırım gücünü korumuştur (yaklaşık 0.95 civarı). En dikkat çekici iyileşme XGBoost modelinde görülmüştür; ROC AUC değeri SMOTE öncesi 0.60 seviyesinden 0.95'e çıkarak modelin sınıf ayırım yeteneğinde dramatik bir artış sağlamıştır. Karışıklık matrisleri analiz edildiğinde, modellerin

çoğunluk sınıfını yüksek doğrulukla tahmin etmelerine rağmen, azınlık sınıfı olan "satın alma" olaylarını yakalamada belirli zorluklar yaşandığı, ancak SMOTE'un bu dengesizliğin etkilerini önemli ölçüde azalttığı görülmüştür. SHAP analizi, `total_addtocarts`, `cart_to_view_ratio` ve `total_views` gibi özelliklerin satın alma olasılığı üzerinde en belirleyici etkiye sahip olduğunu, yüksek değerlerinin satın alma olasılığını artırıcı yönde etki ettiğini ortaya koymuştur.

5.1.2 Zaman Serisi ve Davranışsal Akış Analizi

Gelecekteki satış trendlerini öngörmek amacıyla günlük toplam işlem sayısı tahminlerinde Prophet, ARIMA, Holt-Winters ve LSTM modelleri uygulanmıştır. Metrikler açısından (RMSE, MAE, MAPE), ARIMA ve Holt-Winters modelleri, serinin mevsimsel paternlerini ve dalgalanmalarını gerçek değerlere oldukça yakın bir şekilde yakalayarak tutarlı ve güvenilir tahminler sunmuştur. LSTM modelleri de serinin genel akışına iyi uyum sağlamıştır. Markov Zincirleri analizi, kullanıcıların dijital yolculuklarındaki davranışsal geçişlerini modellemiştir. Analiz, 'view'dan 'view'a yoğun geçişlerin yanı sıra, 'addtocart'tan 'transaction'a geçiş frekansının diğer geçişlere kıyasla belirgin şekilde düşük olduğunu göstermiştir. Bu bulgu, sepet terk oranının yüksek olabileceği ve bu aşamanın e-ticaret operasyonları için potansiyel bir optimizasyon alanı teşkil ettiği yönünde kritik bir içgörü sağlamıştır.

5.1.3 Segmentasyon ve Öneri Sistemleri

Müşterileri davranışsal özelliklerine göre gruplandırmak için KMeans algoritması kullanılmıştır. Dirsek (Elbow) Metodu ve Silhouette Skoru analizleri sonucunda optimal küme sayısı $k = 4$ olarak belirlenmiştir. Bu dört segmentin profilleri, `total_views`, `total_addtocarts`, `total_purchases`, `session_count` ve `avg_session_length` gibi metrikler üzerinden çıkarılmıştır. Segmentler, "Düşük Etkileşimli/Pasif" kullanıcılardan, platformdaki en aktif ve yüksek değerli "Yüksek Değerli/Aktif" kullanıcılara kadar farklı davranışsal paternler sergilemiştir. Her bir müşteri segmentine özel ürün önerileri geliştirmek amacıyla Apriori algoritması ile birliktelik kuralları çıkarılmıştır. Yüksek Kaldıraç (Lift) değerine sahip güçlü ürün ilişkileri belirlenmiştir. Bu kurallar, belirli ürünleri satın alan kullanıcıların başka hangi ürünleri alma eğiliminde olduğunu göstererek, çapraz satış ve kişiselleştirilmiş öneri stratejileri için somut temel oluşturmuştur.

5.1.4 İş Zekası ve Eyleme Dönüştürülebilir İçgörüler

Çalışmanın nihai çıktısı olarak, elde edilen tüm teknik analiz bulguları, iş birimlerinin kolayca anlayabileceği ve stratejik kararlar alabileceği eyleme dönüştürülebilir içgörülere dönüştürülmüştür. Bu, tanımlanan temel performans göstergeleri (KPI'lar) ve stratejik karar akış şeması (Veri → İçgörü → Karar → Aksiyon) aracılığıyla somutlaştırılmıştır. Bu entegre yaklaşım, veri biliminin sadece analiz yapmakla kalmayıp, iş süreçlerini doğrudan etkileyen bir araç olduğunu vurgulamıştır.

5.2 Çalışmanın Katkıları

Bu tez çalışması, e-ticaret müşteri davranış analizine yönelik bütüncül bir veri bilimi çerçevesi sunarak literatüre ve sektöre önemli katkılar sağlamaktadır:

- **Entegre Yaklaşım:** Tahminleme, davranışsal modelleme, segmentasyon ve stratejik karar desteğini bir araya getiren kapsamlı bir metodoloji sunulmuştur.
- **Dengesiz Veri Yönetimi:** Aşırı dengesiz e-ticaret verileri için SMOTE gibi dengeleme tekniklerinin model performansına olan pozitif etkisi sayısal olarak gösterilmiştir.
- **Karmaşık Davranışsal Paternlerin Modellenmesi:** Zaman serisi ve Markov Zincirleri ile kullanıcıların dijital yolculuklarındaki zamana bağlı ve geçişsel davranışları nicel olarak analiz edilmiştir.
- **Açıklanabilir Yapay Zeka Uygulaması:** SHAP analizi ile "kara kutu" modellerin karar mekanizmaları açıklanarak, işletmelere eyleme dönüştürülebilir ve yorumlanabilir içgörüler sağlanmıştır.
- **Stratejik Karar Desteği:** Teknik bulguların iş zekası KPI'larına ve bir karar akış şemasına dönüştürülmesiyle, analizlerin iş değeri somutlaştırılmıştır.

5.3 Çalışmanın Sınırlılıkları ve Gelecek Çalışmalar

Bu tez çalışması önemli içgörüler sunsa da, bazı sınırlılıklara sahiptir ve gelecekteki araştırmalar için potansiyel yollar açmaktadır:

- **Veri Kapsamı:** Kullanılan RetailRocket veri seti yaklaşık 4.5 aylık bir dönemi kapsamakta ve tüm değerleri karma (hashed) formda sunmaktadır.

Spesifik ürün isimleri, fiyat değişimlerinin detaylı etkileri veya kullanıcı demografik bilgileri gibi ek bağlamsal verilerin olmaması, analizlerin derinliğini sınırlamıştır.

- **Hiperparametre Optimizasyonu:** Çalışmada kullanılan modeller için kapsamlı bir hiperparametre optimizasyonu yapılmamıştır. Gelecek çalışmalarda, Grid Search, Random Search veya Bayesyen Optimizasyon gibi yöntemlerle hiperparametre optimizasyonu yapılarak model performanslarının daha da artırılması mümkündür.
- **Model Çeşitliliği ve Hibrit Yaklaşımlar:** Daha ileri derin öğrenme modelleri (örneğin Transformer tabanlı modeller) veya hibrit yaklaşımlar (örneğin, zaman serisi ve sınıflandırma modellerinin entegrasyonu) keşfedilebilir.
- **Gerçek Zamanlı Uygulama ve A/B Testleri:** Geliştirilen tahmin ve öneri sisteminin gerçek zamanlı bir üretim ortamına entegrasyonu ve A/B testleri ile operasyonel etkilerinin doğrudan ölçülmesi, çalışmanın pratik katkısını daha da güçlendirecektir.
- **Dinamik Segmentasyon ve Markov Zincirleri:** Müşteri segmentlerinin zaman içinde nasıl değiştiği incelenebilir veya daha karmaşık (örneğin, ikinci dereceden) Markov zincirleri ile davranışsal geçişlerin daha dinamik modelleri oluşturulabilir.

Bu sınırlılıkların ele alınması ve önerilen gelecek çalışmaların yürütülmesi, e-ticaret müşteri davranış analizi alanındaki bilgi birikimine daha da katkı sağlayacaktır.

- [1] V. Sinap, “Perakende Sektöründe Makine Öğrenmesi Algoritmalarının Karşılaştırmalı Performans Analizi: Black Friday Satış Tahminlemesi,” *Selçuk Üniversitesi Sosyal Bilimler Meslek Yüksekokulu Dergisi*, c. 27, no. 1, ss. 65–90, 2024.
- [2] N. Singh, P. Singh, K. K. Singh A. Singh, “Machine learning based classification and segmentation techniques for CRM: a customer analytics,” *International Journal of Business Forecasting and Marketing Intelligence*, c. 6, no. 2, ss. 99–117, 2020.
- [3] R. Blagus L. Lusa, “SMOTE for high-dimensional class-imbalanced data,” *BMC Bioinformatics*, c. 14, no. 106, ss. 1–16, 2013. doi: [10.1186/1471-2105-14-106](https://doi.org/10.1186/1471-2105-14-106).
- [4] M. P. LaValley, “Logistic Regression,” *Circulation*, c. 117, ss. 2395–2399, 2008. doi: [10.1161/CIRCULATIONAHA.106.682658](https://doi.org/10.1161/CIRCULATIONAHA.106.682658).
- [5] S. F. Sabbeh, “Machine-Learning Techniques for Customer Retention: A Comparative Study,” *International Journal of Advanced Computer Science and Applications (IJACSA)*, c. 9, no. 2, ss. 273–281, 2018.
- [6] D. Pelegrín, J. L. Castillo-Sequera, J. Pascual-Espada M. Martínez-Ballesteros, “Multi-model recommender system for user behavior prediction in e-commerce environments,” *Sensors*, c. 23, no. 6, s. 3180, 2023. doi: [10.3390/s23063180](https://doi.org/10.3390/s23063180).
- [7] M. Patel, D. Patel, J. Rana, V. Chauhan B. R. Bhatt, “Customer Segmentation using Machine Learning,” *International Journal for Research in Applied Science and Engineering Technology*, c. 8, s. 2104, VI 2020. doi: [10.22214/ijraset.2020.6344](https://doi.org/10.22214/ijraset.2020.6344).
- [8] M. Bhagat, P. Vyas C. Rawal, “Leveraging machine learning for customer segmentation and decision support in the retail industry,” *International Journal of Electronic Customer Relationship Management*, c. 13, no. 1, ss. 43–61, 2023. doi: [10.1504/IJECRM.2023.132296](https://doi.org/10.1504/IJECRM.2023.132296).
- [9] D. Xu, Y. Zhang W. Zhou, “Distributed customer behavior prediction using multiplex data: A collaborative multiple kernel support vector machine approach,” *Knowledge-Based Systems*, c. 33, ss. 51–62, 2012. doi: [10.1016/j.knosys.2012.04.023](https://doi.org/10.1016/j.knosys.2012.04.023).
- [10] C.-F. Tsai C.-Y. Chiu, “Mining changes in customer behavior in retail marketing,” *Expert Systems with Applications*, c. 28, no. 4, ss. 773–781, 2005. doi: [10.1016/j.eswa.2004.12.033](https://doi.org/10.1016/j.eswa.2004.12.033).

- [11] R. Gürkan, “Sigortacılık Sektöründe Makine Öğrenmesi ile Müşteri Kaybı Analizi,” *Sakarya Üniversitesi İşletme Enstitüsü Dergisi*, c. 5, no. 1, ss. 42–55, 2023.
- [12] Y. Yaman, “Müşteri Yolculuğu Analitiğini Makine Öğrenmesi Modeliyle Tahminleme,” *YBS Ansiklopedi*, c. 9, no. 1, 2021, Erişim tarihi: 19 Haziran 2025. erişim adresi: <http://www.ybsansiklopedi.com/>.
- [13] M. Kaplan B. Uçkan, “Perakende sektöründe müşteri ilişkileri yönetimi ve dijitalleşmenin etkisi,” *Pazarlama ve Pazarlama Araştırmaları Dergisi*, c. 13, no. 2, ss. 45–60, 2020, Erişim tarihi: 19 Haziran 2025. erişim adresi: <https://example-journal.org/perakende-crm>.
- [14] A. Kurt, “Veri Tabanlı Müşteri İlişkileri Yönetimi: Makine Öğrenmesi Yaklaşımı,” *Bilgi Ekonomisi ve Yönetimi Dergisi*, c. 16, no. 1, ss. 25–39, 2021, Erişim tarihi: 19 Haziran 2025. erişim adresi: <https://example.org/veri-tabanli-crm>.
- [15] P. A. Flach N. Lachiche, “Naive Bayesian Classification of Structured Data,” *Machine Learning*, c. 57, no. 3, ss. 233–269, 2004.
- [16] G. Biau E. Scornet, “A Random Forest Guided Tour,” *arXiv preprint arXiv:1511.05741*, 2015.
- [17] S. Ghosh, N. Das, L. M. Gonçalves U. Maulik, “An improved random forest classifier for multi-class classification,” *IEEE Transactions on Image Processing*, c. 27, no. 8, ss. 4012–4024, 2018. doi: [10.1109/TIP.2018.2834830](https://doi.org/10.1109/TIP.2018.2834830).
- [18] A. Liaw M. Wiener, “Classification and regression by randomForest,” *R news*, c. 2, no. 3, ss. 18–22, 2002.
- [19] W. Verbeke, D. Martens, C. Mues B. Baesens, “Building comprehensible customer churn prediction models with advanced rule induction techniques,” *Expert Systems with Applications*, c. 32, no. 2, ss. 331–340, 2007. doi: [10.1016/j.eswa.2005.04.043](https://doi.org/10.1016/j.eswa.2005.04.043).
- [20] W. Verbeke, D. Martens B. Baesens, “Predicting customer profitability by means of random forests,” *Expert Systems with Applications*, c. 32, no. 2, ss. 375–384, 2007. doi: [10.1016/j.eswa.2005.04.043](https://doi.org/10.1016/j.eswa.2005.04.043).
- [21] S. Ramraj, N. Uzir, S. R S. Banerjee, “Experimenting XGBoost Algorithm for Prediction and Classification of Different Datasets,” *International Journal of Control Theory and Applications*, c. 9, no. 40, ss. 125–133, 2016.
- [22] P. Singh, R. Kumari, A. Nayyar ve diğ., “Performance Analysis of XGBoost Classifier with Missing Data,” *Journal of Advanced Research in Dynamical and Control Systems*, 2023.
- [23] S. S. Dhaliwal, A.-A. Nahid R. Abbas, “Effective Intrusion Detection System Using XGBoost,” *Information*, c. 9, no. 7, s. 149, 2018. doi: [10.3390/info9070149](https://doi.org/10.3390/info9070149).
- [24] W. Zhou, M. Zhang Y. Liu, “Predictive modeling of customer behavior using XGBoost and big retail data,” *Scientific Reports*, c. 13, no. 1, ss. 1–15, 2023. doi: [10.1038/s41598-023-33525-0](https://doi.org/10.1038/s41598-023-33525-0).

- [25] N. V. Chawla, K. W. Bowyer, L. O. Hall W. P. Kegelmeyer, “SMOTE: Synthetic Minority Over-sampling Technique,” *Journal of Artificial Intelligence Research*, c. 16, ss. 321–357, 2002. doi: [10.1613/jair.953](https://doi.org/10.1613/jair.953).
- [26] A. Andika, H. A. Nugroho, A. Wibowo, I. G. P. S. Purnama ve diğ., “A Survey of SMOTE and Its Variants for Imbalanced Classification,” *2021 Sixth International Conference on Informatics and Computing (ICIC)*, IEEE, 2021, ss. 1–6. doi: [10.1109/ICIC54025.2021.9632912](https://doi.org/10.1109/ICIC54025.2021.9632912).
- [27] G. Douzas F. Bacao, “DeepSMOTE: imbalanced learning in deep neural networks using synthetic data generation,” *IEEE Transactions on Neural Networks and Learning Systems*, c. 34, no. 9, ss. 6390–6404, 2023. doi: [10.1109/TNNLS.2021.3136503](https://doi.org/10.1109/TNNLS.2021.3136503).
- [28] C. P. Da Veiga, C. R. P. Da Veiga, A. Catapan, U. Tortato W. V. Da Silva, “Demand forecasting in food retail: a comparison between the Holt-Winters and ARIMA models,” *WSEAS TRANSACTIONS on BUSINESS and ECONOMICS*, c. 11, ss. 608–614, 2014, issn: 2224-2899.
- [29] T.-C. Fu, F.-L. Chung V. Ng, “A survey on time series data mining,” *Engineering Applications of Artificial Intelligence*, c. 24, no. 1, ss. 164–181, 2011. doi: [10.1016/j.engappai.2010.09.007](https://doi.org/10.1016/j.engappai.2010.09.007).
- [30] Z. Chen, C. Wang, L. Lv, L. Fan, S. Wen Z. Xiang, “Research on Peak Load Prediction of Distribution Network Lines Based on Prophet-LSTM Model,” *Sustainability*, c. 15, no. 15, s. 11 667, 2023. doi: [10.3390/su1511667](https://doi.org/10.3390/su1511667).
- [31] F. Karim, S. Majumdar, H. Darabi S. Chen, “Multivariate LSTM-FCNs for time series classification,” *Neural Networks*, c. 116, ss. 237–245, 2019. doi: [10.1016/j.neunet.2019.04.014](https://doi.org/10.1016/j.neunet.2019.04.014).
- [32] X. Zhang, Y. Zhang, H. Wang, S. Wang W. Wang, “A hybrid method for financial time series forecasting based on empirical mode decomposition and LSTM,” *Physica A: Statistical Mechanics and its Applications*, c. 513, ss. 511–523, 2019. doi: [10.1016/j.physa.2018.11.061](https://doi.org/10.1016/j.physa.2018.11.061).
- [33] N. Ersen, K. C. Akyüz İ. Akyüz, “Orman Ürünleri ve Mobilya Sanayinde Hisse Senedi Fiyat Hareketlerinin Markov Zincirleri Yöntemi ile Tahmin Edilmesi,” *DÜ Ormancılık Dergisi*, c. 1, no. 1, ss. 43–58, 2023. doi: [10.58816/duzceod.1288486](https://doi.org/10.58816/duzceod.1288486).
- [34] J. Li ve diğ., “Application of XGBoost Algorithm in the Optimization of Pollutant Concentration,” *SSRN Electronic Journal*, 2022.
- [35] Z. Singla, S. Randhawa S. Jain, “Sentiment Analysis of Customer Product Reviews Using Machine Learning,” *2017 International Conference on Intelligent Computing and Control (I2C2)*, IEEE, 2017, ss. 530–535.
- [36] Y. E. GÜR, K. A. ESİDİR C. AYDEN, “Türkiye’de E-Ticaretin Kullanılma Durumunun Makine Öğrenmesi İle Sınıflandırılması Ve Çeşitli Değişkenlerle İlişkilerinin Analizi,” *KARADENİZ SOSYAL BİLİMLER DERGİSİ*, c. 16, no. 31, ss. 582–610, 2024. doi: [10.38155/ksbd.1477120](https://doi.org/10.38155/ksbd.1477120).

- [37] E. P. Chopra, A. Chhapola D. S. Kaushik, “Comparative Analysis Of Optimizing AWS Inferentia With Fastapi And Pytorch Models,” *International Journal of Creative Research Thoughts (IJCRT)*, c. 10, no. 2, e449–e463, 2022.
- [38] N. Li, L. Zeng, Q. He Z. Shi, “Parallel Implementation of Apriori Algorithm Based on MapReduce,” *International Journal of Networked and Distributed Computing*, c. 1, no. 2, ss. 89–96, 2013.



TEZDEN ÜRETİLMİŞ YAYINLAR

Konferans Bildirisi

1. TUNCER, R.G., ONAR, S. (2025). **Makine Öğrenmesi Yöntemleriyle Müşteri Davranış Analizi.** In N.E. Özdemir (Ed.), 6. *BİLSEL International Ahlat Scientific Researches Congress Book* (pp. 192–203). BİLSEL YAYINCILIK.

