

Müşteri İlişkileri Yönetimde Veri Madenciliği: Bir Bankanın Kredi Müşterilerine

Yönelik Uygulama Denemesi

Arda Baran Baytar

LİSANS TEZİ

İstatistik Anabilim Dalı

Eskişehir, 2026

ESKİŞEHİR OSMANGAZİ ÜNİVERSİTESİ
FEN FAKÜLTESİ
İSTATİSTİK BÖLÜMÜ

**MÜŞTERİ İLİŞLERİ YÖNETİMDE VERİ MADENCİLİĞİ: BİR BANKANIN
KREDİ MÜŞTERİLERİNE YÖNELİK UYGULAMA DENEMESİ**

Arda Baran BAYTAR

Lisans Tezi

Danışman: Prof. Dr. Özlem ALPU

Eskişehir, 2026

İÇİNDEKİLER

İÇİNDEKİLER	i
ŞEKİLLER DİZİNİ	v
ÇİZELGELER DİZİNİ	viii
1. GİRİŞ VE AMAÇ	1
1.1. Müşteri İlişkileri Yönetimi	1
2. VERİ MADENCİLİĞİ.....	2
2.1. Veri Madenciliği İşlevleri	2
2.1.1. Sınıflandırma ve regresyon yöntemleri.....	2
2.1.1.1. Karar ağaçları.....	3
2.1.1.2. Random forest.....	5
2.1.1.3. K en yakın komşu	5
2.1.1.4. Naif (Naive) Bayes	6
2.1.1.5. Yapay Sinir Ağları.....	7
2.1.1.6. Multi-layer perceptron (MLP) predictor.....	7
2.1.2. Regresyon	8
2.1.2.1. Doğrusal regresyon	8
2.1.2.2. Lojistik regresyon	8
2.2. Kümeleme	9
2.2.1. Hiyerarşik olan kümeleme algoritmaları	9
2.2.1.1. En uzak komşu.....	9
2.2.1.2. Ortalama bağlantı.....	9
2.2.2. Hiyerarşik olmayan kümeleme algoritmaları	9
2.2.2.1. K-ortalama (K-means)	10
2.2.2.2. K-Medoids	10
2.2.2.3. DBSCAN	10
2.3. Birliktelik	10
2.3.1. Apriori algoritması.....	11
2.4. Algoritmalarda Kullanılan Performans Metrikleri.....	11
2.5. Veri Madenciliğinde Kullanılan Programlar	15
2.6. Veri Madenciliği Uygulayan Sektörler	16
2.7. Bankacılık Sektöründe Veri Madenciliği Uygulamaları	17

3. MÜŞTERİ İLİŞKİLERİ YÖNETİMİ ALANINDA VERİ MADENCİLİĞİ	
UYGULAMALARI.....	17
3.1. MİY’de Veri Madenciliğinin Yeri	17
3.2. MİY ve Veri Madenciliği Alanında Yapılan Bazı Çalışmalar	18
3.3. Bankacılık Sektöründe MİY ve Veri Madenciliği.....	20
4. UYGULAMA	23
4.1. İşin Anlaşılması.....	23
4.2. Verilerin Anlaşılması.....	23
4.3. Verilerin Hazırlanması	24
4.3.1. Kurulum aşaması	24
4.3.2. Verilerin açıklanması	25
4.3.3. Verilerin temizlenmesi ve düzenlenmesi aşaması.....	25
4.3.4. Verilerin Değerlendirilmesi.....	25
4.3.4.1. Kategorik verilerin değerlendirilmesi	26
4.3.4.2. Veri setindeki sayısal değerlerin değerlendirilmesi	27
4.4. Birinci Problem Modelleme.....	28
4.4.1. Birinci problemin Knime ile çözümü	35
4.4.1.1. Birinci problem için Knime ile decision tree algoritması çözümü	35
4.4.1.2. Birinci problem için Knime ile Naive Bayes algoritması Çözümü	39
4.4.1.3. Birinci problem için Knime ile K nearest neighbor algoritması çözümü	
.....	42
4.4.1.4. Knime birinci problemin algoritmalarının karşılaştırılması	44
4.4.2. Birinci problemin Orange ile çözümü	45
4.4.2.1. Birinci problem için Orange ile decision tree algoritması çözümü	46
4.4.2.2. Birinci problem için Orange ile Naive Bayes algoritması çözümü	47
4.4.2.3. Birinci problem için Orange ile K nearest neighbor algoritması çözümü	
.....	49
4.4.2.4. Orange birinci problemin algoritmalarının karşılaştırılması	51
4.4.3. Birinci problemin Python ile çözümü	52
4.4.3.1. Birinci problem için Python ile decision tree algoritması çözümü.....	57
4.4.3.2. Birinci problem için Python ile Naive Bayes algoritması çözümü.....	58
4.4.3.3. Birinci problem için Python ile K nearest neighbor algoritması çözümü	
.....	60
4.4.3.4. Python birinci problemin algoritmalarının karşılaştırılması	61

4.4.3.5. Referans çalışma ile doğruluk oranları karşılaştırma analizi.....	61
4.4.3.6. Veri setinin dengelenmesi: SMOTE uygulaması	62
4.4.3.7. İlk dengeleme aşaması	63
4.4.3.8. Veri hacminin genişletilmesi.....	63
4.4.4. Birinci problemin R ile çözümü.....	65
4.4.4.1. Birinci problem için R ile decision tree algoritması çözümü	65
4.4.4.2. Birinci problem için R ile Naive Bayes algoritması çözümü	66
4.4.4.3. Birinci problem için R ile KNN algoritması çözümü	67
4.4.4.4. Birinci problem için R ile XGBoost algoritması çözümü.....	67
4.4.5. Birinci problem için en iyi program seçimi	68
4.5. İkinci ve Üçüncü Problem Modelleme	70
4.5.1. İkinci ve üçüncü problem için Knime çözümü	71
4.5.1.1. İkinci ve üçüncü problem için Knime ile multi-layer perceptron predictor algoritması çözümü	71
4.5.2. İkinci ve üçüncü problemlerinin Orange ile çözümü.....	73
4.5.2.1. İkinci ve üçüncü problemlerinin Orange algoritmaların birlikte çözümü	73
4.5.3. İkinci ve üçüncü problemlerinin Python ile çözümü	75
4.5.3.1. İkinci ve üçüncü problemlerin Python ile multi-layer perceptron predictor algoritması çözümü	76
4.5.4. İkinci ve üçüncü problemlerinin R ile çözümü.....	77
4.5.5. İkinci ve üçüncü problemleri için en iyi program seçimi	78
4.6. Dördüncü Problem Modelleme.....	80
4.6.1. Dördüncü problem Knime ile çözümü.....	81
4.6.1.1. Borcunu ödemeyenler için Knime ile K-Means algoritması çözümü....	82
4.6.1.2. Borcunu ödeyenler için Knime ile K-Means algoritması çözümü	85
4.6.2. Dördüncü problem Orange ile çözümü.....	87
4.6.2.1. Borcunu ödemeyenler için Orange ile K-Means algoritması çözümü... 88	
4.6.2.2. Borcunu ödeyenler için Orange ile K-Means algoritması çözümü.....	91
4.6.3. Dördüncü problemin Python ile çözümü	94
4.6.3.1. Borcunu ödemeyenler için Python ile K-Means algoritması çözümü ...	94
4.6.3.2. Borcunu ödeyenler için Python ile K-Means algoritması çözümü	97
4.6.4. Dördüncü problemin R ile çözümü.....	101
4.6.4.1. Borcunu ödemeyenler için R ile K-Means algoritması çözümü.....	101

4.6.4.2. Borcunu ödeyenler için R ile K-Means algoritması çözümü.....	104
4.6.5. Dördüncü problem için en iyi program seçimi	107
4.6.5.1. Dördüncü problem borcunu ödemeyen müşteriler için programları karşılaştırma.....	108
4.6.5.2. Dördüncü problem borcunu ödeyen müşteriler için programları karşılaştırma.....	109
5. TARTIŞMA VE SONUÇLAR	111
5.1. Birinci Problem İçin Sonuçlar ve Değerlendirme.....	112
5.2. İkinci ve Üçüncü Problemler İçin Sonuçlar ve Değerlendirme	112
5.3. Dördüncü Problem İçin Sonuçlar ve Değerlendirme.....	112
6. KAYNAKÇA	114

ŞEKİLLER DİZİNİ

<u>Şekil</u>	<u>Sayfa</u>
Şekil 2.1. Karar ağacı düğümleri	4
Şekil 2.2. Birinci grafik K-en yakın komşu yöntemi	6
Şekil 2.3. İkinci grafik k=3 için K-en yakın komşu yöntemi (Aşlıyan ve Günel, 2010)..	6
Şekil 4.1. Veri setinin Python'da veri tipleri ve veri sayıları çıktısı	26
Şekil 4.2. Sayısal değerlerin korelasyon analiz değerleri ısı matrisi	27
Şekil 4.3. COLLECTION_PAIDOFF durumundaki krediler için effective_date, due_date ve paid_off_date tarihlerinin karşılaştırılması	31
Şekil 4.4. Paid_off_date frekans grafiği	32
Şekil 4.5. 28/09/2016, 11/10/2016 ve 12/10/2016 tarihlerinde eğitim seviyelerine ve cinsiyete göre ödeme dağılımının grafiği	34
Şekil 4.6. Birinci problem Knime ile tüm Algoritmaların akış şeması	35
Şekil 4.7. Birinci problem Knime ile Decision Tree algoritması akış şeması	36
Şekil 4.8. Birinci problemin Knime ile Decision Tree Confusion Matrix	38
Şekil 4.9. Birinci problem Knime Naive Bayes algoritması akış şeması	39
Şekil 4.10. Birinci problemin Knime ile Naive Bayes Confusion Matrix	41
Şekil 4.11. Birinci problem Knime ile K Nearest Neighbor algoritması akış şeması	42
Şekil 4.12. Birinci problemin Knime ile K Nearest Neighbor Confusion Matrix	44
Şekil 4.13. Birinci problem Orange Decision Tree algoritması akış şeması	46
Şekil 4.14. Birinci problemin Orange ile Decision Tree Confusion Matrix	47
Şekil 4.15. Birinci Problem Orange Naive Bayes algoritması akış şeması.	48
Şekil 4.16. Birinci problemin Orange ile Naive Bayes Confusion Matrix	49
Şekil 4.17. Birinci Problem Orange K Nearest Neighbor algoritması akış şeması	50
Şekil 4.18. Birinci problemin Orange ile K Nearest Neighbor Confusion Matrix	51
Şekil 4.19. Birinci problem için Python ile veri seti görüntüsü	53
Şekil 4.20. Birinci problem için Python ile sayısal bilgiler	54
Şekil 4.21. Birinci problem için Python ile değişken dağılımları	55
Şekil 4.22. Birinci problem için Python ile boş değer analizi	56
Şekil 4.23. Birinci problem için Python ile değişkenlerin korelasyon matrisi	56
Şekil 4.24. Birinci problem için Python ile değişkenin sayısallaştırılması	57
Şekil 4.25. Birinci problem için Python ile Decision Tree algoritması Confusion Matrix	58

Şekil 4.26. Birinci problem için Python ile Naive Bayes algoritması Confusion Matrix	59
Şekil 4.27. Birinci problem için Python ile KNN algoritması Confusion Matrix	60
Şekil 4.28. Birinci problem için Python ile SMOTE sonrası model gelişim analizi	64
Şekil 4.29. İkinci ve üçüncü problemlerin Knime akış şeması.....	71
Şekil 4.30. İkinci ve üçüncü problemlerin Knime MultiLayer Perceptron Predictor algoritması akış şeması	72
Şekil 4.31. İkinci ve üçüncü problemlerin Orange algoritmalar akış şeması	74
Şekil 4.32. Dördüncü problem borcunu ödemeyen ve ödeyen müşterilerin Knime ile K-Means algoritması akış şeması	82
Şekil 4.33. Dördüncü problem borcunu ödemeyenler için Knime ile Decision Tree algoritması ağaç yapısı.....	83
Şekil 4.34. Dördüncü problem borcunu ödemeyenler için Knime ile K-Means algoritması “Principal” ve “terms” değişkenlerine göre saçılım grafiği.....	84
Şekil 4.35. Dördüncü problem borcunu ödeyenler için Knime ile Decision Tree algoritması ağaç yapısı.....	85
Şekil 4.36. Dördüncü problem borcunu ödeyenler için Knime ile K-Means algoritması “Principal” ve “terms” değişkenlerine göre saçılım grafiği.....	86
Şekil 4.37. Dördüncü problem borcunu ödemeyenler ve ödeyenler için Orange ile K-Means algoritması akış şeması	87
Şekil 4.38. Dördüncü problem borcunu ödemeyenler için Orange ile K-Means algoritması akış şeması	88
Şekil 4.39. Dördüncü problem borcunu ödemeyenler için Orange ile Decision Tree algoritması ağaç yapısı.....	89
Şekil 4.40. Dördüncü problem borcunu ödemeyenler için Orange ile K-Means algoritması “duedate” ve “past_due_days” grafiği	90
Şekil 4.41. Dördüncü problem borcunu ödeyenler için Orange ile K-Means algoritması akış şeması	91
Şekil 4.42. Dördüncü problem borcunu ödeyenler için Orange ile Decision Tree algoritması akış şeması	92
Şekil 4.43. Dördüncü problem borcunu ödeyenler için Orange ile K-Means algoritması “paid_off_time” ve “terms” grafiği	93
Şekil 4.44. Dördüncü problem borcunu ödemeyenler için Python ile K-Means algoritması ağaç yapısı.....	95

Şekil 4.45. Dördüncü problem borcunu ödemeyenler için Python ile K-Means algoritması “terms” ve “principal grafiği.....	96
Şekil 4.46. Dördüncü problem borcunu ödeyenler için Python ile K-Means algoritması ağaç yapısı.....	98
Şekil 4.47. Dördüncü problem borcunu ödeyenler için Python ile K-Means algoritması “terms” ve “past_due_days” grafiği	99
Şekil 4.48. Dördüncü problem borcunu ödemeyenler için R ile K-Means algoritması ağaç yapısı.....	101
Şekil 4.49. Dördüncü problem borcunu ödemeyenler için R ile K-Means algoritması “terms” ve “principal” grafiği.....	102
Şekil 4.50. Dördüncü problem borcunu ödeyenler için R ile K-Means algoritması ağaç yapısı.....	105
Şekil 4.51. Dördüncü problem borcunu ödeyenler için R ile K-Means algoritması “terms” ve “past_due_days” grafiği	106

ÇİZELGELER DİZİNİ

<u>Çizelge</u>	<u>Sayfa</u>
Çizelge 2.1. Sınıflandırma performans metrikleri	12
Çizelge 2.2. Regresyon performans metrikleri	13
Çizelge 2.3. Kümeleme performans metrikleri.....	14
Çizelge 2.4. Diğer performans metrikleri	14
Çizelge 2.5. VM'nin en yaygın olarak kullanıldığı alanlar	16
Çizelge 4.1. Loan Veri seti Excel görüntüsü.....	24
Çizelge 4.2. Banka kredi müşterileri veri seti alanları ve açıklamaları	25
Çizelge 4.3. Kategorik değişkenlerin sayısal değerleri.....	26
Çizelge 4.4. Nümerik değişkenler arasındaki alanların değerleri	27
Çizelge 4.5. PAIDOFF durumundaki kredilerle ilgili özet bilgileri	29
Çizelge 4.6. COLLECTION durumundaki kredilerle ilgili özet bilgileri.....	29
Çizelge 4.7. COLLECTION PAIDOFF durumundaki kredilerle ilgili özet bilgileri	30
Çizelge 4.8. Ödeme süresi geçtikten sonra borçlarını ödeyen müşterilerin detayları.....	30
Çizelge 4.9. 28/09/2016, 11/10/2016 ve 12/10/2016 tarihlerinde yapılan ödemelerle ilgili bazı gözlemler	33
Çizelge 4.10. Birinci problem için Knime ile Decision Tree algoritması performans metrikleri.....	36
Çizelge 4.11. Birinci problem Knime ile Decision Tree algoritması diğer performans metrikleri 2.....	37
Çizelge 4.12. Birinci problem Knime ile Decision Tree algoritması diğer performans metrikleri 3.....	37
Çizelge 4.13. Birinci problem Knime ile Naive Bayes algoritması performans metrikleri	39
Çizelge 4.14. Birinci problem Knime ile Naive Bayes algoritması performans metrikleri 2	40
Çizelge 4.15. Birinci problem Knime ile Naive Bayes algoritması performans metrikleri 3	41
Çizelge 4.16. Birinci problem Knime ile K Nearest Neighbor algoritması performans metrikleri.....	42
Çizelge 4.17. Birinci problem Knime ile K Nearest Neighbor algoritması performans metrikleri 2.....	43

Çizelge 4.18. Birinci Problem Knime ile K Nearest Neighbor algoritması performans metrikleri 3.....	43
Çizelge 4.19. Birinci problem Knime ile tüm algoritmaların performans metrikleri	45
Çizelge 4.20. Birinci problem Orange ile Decision Tree için genel performans metrikleri	46
Çizelge 4.21. Birinci problem Orange ile Naive Bayes için genel performans metrikleri	48
Çizelge 4.22. Birinci problem Orange ile K Nearest Neighbor için genel performans metrikleri.....	50
Çizelge 4.23. Birinci problem Orange ile tüm algoritmaların performans metrikleri	52
Çizelge 4.24. Birinci problem için Python ile Decision Tree algoritması performans metrikleri.....	57
Çizelge 4.25. Birinci problem için Python ile Naive Bayes algoritması performans metrikleri.....	58
Çizelge 4.26. Birinci problem için Python ile K Nearest Neighbor algoritması performans metrikleri	60
Çizelge 4.27. Birinci problem Python ile tüm algoritmaların performans metrikleri.....	61
Çizelge 4.28. Birinci problem Python ile referans çalışma karşılaştırma performans metrikleri.....	62
Çizelge 4.29. Birinci problem için Python ile çapraz doğrulama performans metrikleri	63
Çizelge 4.30. İkinci aşama SMOTE uygulaması ile genişletilmiş veri setine ait karşılaştırmalı performans metrikleri.....	64
Çizelge 4.31. Birinci problem için R ile Decision Tree algoritması performans metrikleri	65
Çizelge 4.32. Birinci problem için R ile Naive Bayes algoritması performans metrikleri	66
Çizelge 4.33. Birinci problem için R ile KNN algoritması performans metrikleri	67
Çizelge 4.34. Birinci problem için R ile XGBoost algoritması performans metrikleri ..	68
Çizelge 4.35. Birinci problem için tüm programların performans metrikleri.....	69
Çizelge 4.36. İkinci ve üçüncü problemlerin modüler çözüm adımları.....	70
Çizelge 4.37. İkinci ve üçüncü problemlerin Knime MultiLayer Perceptron Predictor algoritması performans metrikleri	72
Çizelge 4.38. İkinci ve üçüncü problemler için Orange algoritmalarının performans metrikleri.....	74

Çizelge 4.39. İkinci problemin Python ile MultiLayer Perceptron Predictor algoritması performans metrikleri	76
Çizelge 4.40. Üçüncü problemin Python ile MultiLayer Perceptron Predictor algoritması performans metrikleri	76
Çizelge 4.41. İkinci problemin R ile MultiLayer Perceptron Predictor algoritması performans metrikleri	77
Çizelge 4.42. Üçüncü problemin R ile MultiLayer Perceptron Predictor algoritması performans metrikleri	78
Çizelge 4.43. İkinci ve üçüncü problemler için tüm programların performans metrikleri	79
Çizelge 4.44. Dördüncü problem borcunu ödemeyenler için Python ile K-Means algoritması performans metrikleri	97
Çizelge 4.45. Dördüncü problem borcunu ödeyenler için Python ile K-Means algoritması performans metrikleri	100
Çizelge 4.46. Dördüncü problem borcunu ödemeyenler için R ile K-Means algoritması performans metrikleri	103
Çizelge 4.47. Dördüncü problem borcunu ödeyenler için R ile K-Means algoritması performans metrikleri	107
Çizelge 4.48. Dördüncü problem borcunu ödemeyenler için programların performans metrikleri.....	108
Çizelge 4.49. Dördüncü problem borcunu ödeyenler için programların performans metrikleri.....	109
Çizelge 4.50. Dördüncü problem borcunu ödeyenler için programların performans metrikleri 2.....	110
Çizelge 5.1. Problemlerde kullanılan program ve algoritmalar.....	111

1. GİRİŞ VE AMAÇ

Bu çalışmada, YÖKTez platformunda 899661 kayıt numarasıyla yayımlanmış olan "Müşteri İlişkileri Yönetiminde Veri Madenciliği: Bir Bankanın Kredi Müşterilerine Yönelik Uygulama" başlıklı tez esas alınmış; söz konusu çalışmada uygulanan yöntemler yeniden denenerek benzer sonuçların elde edilmesi ve bu sonuçların geliştirilmesi amaçlanmıştır. Referans tezde kullanılan veri seti Kaggle platformundan temin edilmiş, gerekli ön işleme aşamalarından geçirilerek modellemeye uygun hâle getirilmiştir. Araştırma kapsamında veri madenciliği tekniklerinden kümeleme ve sınıflandırma algoritmaları uygulanmış; dört temel hedef doğrultusunda analizler gerçekleştirilmiştir. Bu hedefler sırasıyla şunlardır: müşterilerin borç ödeme durumunu tahmin eden bir sistemin geliştirilmesi, ödeme zamanının öngörülmesine yönelik bir modelin oluşturulması, müşterilere sağlanabilecek kredi miktarının tahmin edilmesi ve müşteri segmentasyonunun gerçekleştirilmesi.

Referans araştırmada analizler KNIME, R, Orange ve Python gibi veri analitiği araçları kullanılarak yürütülmüş ve her problem için en uygun algoritmalar belirlenmiştir. Mevcut çalışmada ise daha sonradan söz konusu araçlara ek olarak R programlama dili ve XGBoost algoritması da analizlere dahil edilecektir.

1.1. Müşteri İlişkileri Yönetimi

Teknolojinin hızlı gelişimiyle birlikte hem bireyler arası ilişkiler hem de şirketlerin müşterileriyle olan etkileşimleri kayda değer biçimde değişim geçirmiştir. Günümüzde, şirketler arasındaki rekabet giderek yoğunlaşmakta, coğrafi sınırlar anlamını yitirmekte ve müşterilerin kişisel tercihleri daha fazla önem kazanmaktadır. Bu nedenle, özelleşmiş birebir pazarlama stratejileri öne çıkmaktadır. Şirketler yeni girişim sayısının artmasıyla birlikte ise, müşterilerinin sadakatini sağlamada zorluk çekmekte ve onlara daha yakın olma gerekliliği hissetmektedir (Odabaşı, 2015). Tüm bu değişen koşullara çare olarak ortaya çıkan Müşteri İlişkileri Yönetimi (MİY) ise, işletmenin tüm birimlerinin uyum içinde çalışmasını sağlayan ve entegre bir şekilde uygulanan kapsamlı bir pazarlama, satış ve hizmet stratejisidir (Kotler ve Keller , 2015).

2. VERİ MADENCİLİĞİ

2.1. Veri Madenciliği İşlevleri

VM modelleri işlevlerine göre üç ana başlık altında toplanabilir (Chorianopoulos ve Tsiptsis, 2009) Bunlar:

- Sınıflama (Classification) ve Regresyon (Regression)
- Kümeleme (Clustering)
- Birliktelik Kuralları (Association Rules)

Sınıflama ve regresyon modelleri tahmin edici model yöntemleridir. Kümeleme ve birliktelik kuralları tanımlayıcı yöntem modelleridir (Akpınar, 2000). Birçok algoritma VM’de sınıflandırma, kümeleme ve birliktelik kuralları analizi gibi modeller uygulanmak üzere geliştirilmiştir. Bu model yöntemlerinin hemen hemen hepsi istatistiksel tabanlı algoritmalarından oluşmaktadır. Bu algoritmaların bir kısmı sadece sınıflandırma yöntemleri ya da kümeleme yöntemleri gibi konular üzerinde yoğunlaşırken bir takımı ise farklı varyasyonlarla birçok konuda uygulanabilmektedir. Örnek verecek olunursa, genetik algoritmalar, yapay sinir ağları gibi algoritmalar sınıflandırma ve kümeleme modellerinde uygulanabilirler. Diğer yandan Apriori algoritması sadece birliktelik kurallarının belirlenmesinde kullanılan bir algoritmadır denilebilir. (Agrawal ve Srikant, 1994).

2.1.1. Sınıflandırma ve regresyon yöntemleri

Sınıflama ve regresyon yöntemleri, önemli veri sınıflarını belirleyerek veya gelecekteki veri eğilimlerini tahmin ederek modeller oluşturur (Han vd., 2011). Sınıflandırma yöntemleri, verileri önceden belirlenmiş sonuçlar ışığında bölümlendiren bir tekniktir. Bu süreç, sonuçların bilindiği ve veri setinin denetimli bir şekilde öğrenildiği anlamına gelir (Giudici, 2000).

Sınıflama yöntemleri kategorik verileri tahmin ederken, regresyon ise sürekli verilerin tahmin edilmesinde uygulanan bir yöntemlerdir (Han vd., 2011). Sınıflandırma işlemi, veri tabanı kayıtlarını inceleyerek önceden belirlenmiş sınıflara göre verileri ayırıştırır ve bu şekilde gelecekteki müşteri davranışlarını tahmin etmek için modeller oluşturur. Yapay

sinir ağırları ve karar ağaçları gibi araçlar sınıflandırma modellerinde sıkça kullanılır. MİY için sınıflandırma modeli, gelecekteki müşteri davranışlarını tahmin etmekte en çok kullanılan tekniktir (Ngai, Xiu, ve Chau, 2009).

Regresyon istatistiksel modelleri, sürekli değişkenlerin tahmini için kullanılan bir tekniktir. Karşılaşılan birçok problem ve sorunlar, lineer regresyon modeli uygulanarak çözüme kavuşturulabilir. Regresyon modellerinde daha karmaşık değişkenler üzerinde dönüşümler uygulanır. Bu sayede, doğrusal olmayan problemlerin doğrusal problemler olarak dönüşümü sağlanabilir (Han vd., 2011).

Regresyon analizi, bağımlı bir değişkenin bir veya daha fazla bağımsız değişkenle olan ilişkisinin matematiksel bir fonksiyon olarak ifade edilmesidir. Oluşturulan bu matematiksel fonksiyon sayesinde bağımlı değişkenin gelecekteki değerinin tahmin edilmesine olanak sağlamaktadır (Sykes, 1993). Sınıflandırma modelleri olarak uygulamada kullanılan; karar ağaçları (Decision Tree), Random Forset, K En Yakın Komşu (K-Nearest Neighbors), Naif (Naive) Bayes, Neural Network (Yapay Sinir Ağı), Multi-Layer Perceptron (MLP) Predictor ve Regresyon modellerine bu tezde yer verilmiştir.

2.1.1.1. Karar ağaçları

Karar ağaçları, sınıflandırma ve regresyon görevleri için kullanılan parametrik olmayan, denetimli bir öğrenme algoritmasıdır. Bu algoritma, kök düğümden başlayarak dallar, iç düğümler ve yaprak düğümlerden oluşan hiyerarşik bir ağaç yapısına sahiptir. Her bir düğüm, veri setindeki özelliklere göre veriyi bölerek, nihai bir sınıf etiketi veya tahmin değerine ulaşır (Han vd., 2011). Şekil 2.1.'de ise Karar Ağacı düğümleri verilmiştir.



Şekil 2.1. Karar ağacı düğümleri

Yukarıdaki Şekil 2.1’de görülebileceği gibi, bir karar ağacı, kendisine doğru uzanan dalları olmayan bir kök düğümle başlar. Kök düğümden başlayan dallar daha sonra karar düğümleri olarak da adlandırılan iç düğümlere akar. Mevcut özelliklere dayalı olarak, yaprak düğümler veya terminal düğümler olarak adlandırılan homojen alt kümeler oluşturmak için her iki düğüm türü üzerinde değerlendirmeler gerçekleştirilir. Yaprak düğümler veri seti içindeki tüm olası sonuçları temsil eder (Han vd., 2011).

Karar ağaçları, yorumlanabilirlikleri, veri tabanı sistemlerine kolayca entegre edilebilmeleri ve güvenilirliklerinin yüksek olması nedeniyle sınıflandırma modelleri arasında yaygın olarak kullanılır (Linoff ve Berry , 1999). Bu teknik, karar vericilerin olasılıklar ve maksimum fayda temelinde en uygun karara ulaşmalarına yardımcı olur. (Han vd., 2011).

Karar ağaçları, kök düğümden başlayarak yukarıdan aşağıya doğru ilerleyerek veriyi ayrıştırır ve sınıflandırır. Modelin doğruluğu, test verisi ile doğrulandıktan sonra, yeni veriler üzerinde sınıflama işlemi gerçekleştirilir. Bu işlem sonucunda yeni verinin sınıfı belirlenebilir (Özekes, 2003).

Karar ağacı analizi, genellikle seçenekler üzerinde uygulanan bu analiz türlerinin VM’de kullanılma sebepleri olarak şunlar sayılabilir (Chu ve Lin, 2005):

- Maliyetleri azaltmaktadır.
- Yöntemin yorumlanmasını ve anlaşılması oldukça basite indirger.
- Veri tabanlarına kolay bir şekilde uygulanabilir hale getirilebilir.
- Güvenilirliği oldukça yüksektir.

Karar ağaçları oluşturmak için birçok algoritma geliştirilmiştir. Bunlar arasında CHAID (Chi-Squared Automatic Interaction Detector), Exhaustive CHAID, MARS

(Multivariate Adaptive Regression Splines), ID3, CART, C4.5, QUEST (Quick, Unbiased, Efficient Statistical Tree), SLIQ (Supervised Learning in Quest), SPRINT (Scalable Parallelizable Induction of Decision Trees) ve CRT (Classification and Regression Trees) bulunmaktadır. Bu algoritmalar, yukarıdan aşağıya doğru ilerleyen "böl ve yönet" yaklaşımını kullanır (Han vd., 2011).

2.1.1.2. Random forest

Random Forest, makine öğreniminde hem sınıflandırma hem de regresyon problemlerini çözmek için kullanılan bir yöntemidir. Bu algoritma, çok sayıda karar ağacının bir araya getirilmesiyle oluşturulur. Her ağaç, veri setinin rastgele seçilen bir alt kümesi ve özellikleri kullanılarak eğitilir, bu da modelin genelleme yeteneğini artırır ve tek bir ağacın yaratabileceği aşırı öğrenme riskini azaltır. Sınıflandırma durumlarında, model her bir ağacın oylarını alarak en yaygın sınıfı seçerken, regresyonda bu ağaçlardan elde edilen tahminlerin ortalaması alınır.

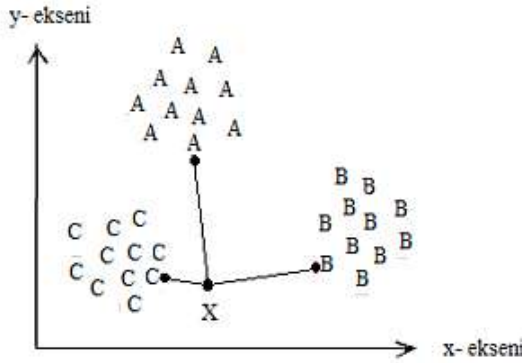
Random Forest algoritması, özellikle yüksek boyutlu ve karmaşık veri setleri üzerinde etkili çalışabilmesi, gürültüye karşı dayanıklı olması ve aşırı uyum sorununu hafifletmesi gibi avantajları nedeniyle yaygın olarak tercih edilir. Bu özellikleri, onu birçok veri bilimi uygulamasında güvenilir ve güçlü bir araç haline getirmektedir (Breiman, 2001).

2.1.1.3. K en yakın komşu

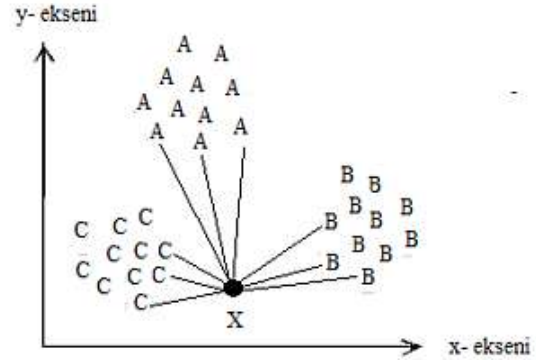
K-Nearest Neighbors (KNN) yöntemi, sınıflandırma problemlerini çözen denetimli öğrenme tekniklerinden biridir. Bu yöntemde, sınıflandırılacak verilerin öğrenme kümesindeki normal davranış verilerine benzerlikleri hesaplanır ve en yakın olan k verinin ortalaması alınarak sınıflandırma yapılır. Her sınıfın özelliklerinin önceden belirlenmiş olması önemlidir. KNN yönteminin performansı, k-en yakın komşu sayısı, eşik değeri, benzerlik ölçümü ve öğrenme kümesindeki normal davranışların yeterli sayıda olmasına bağlıdır (Çalışkan ve Soğukpınar, 2008).

KNN yöntemi, sınıflandırılacak verinin en yakın komşularına dayanarak sınıflandırma yapar. Şekil 2.2'de, X noktası sınıflandırılacak veriyi, A, B ve C ise sınıfları temsil etmektedir. Bu örnekte X, en yakın komşusu olan C sınıfına ait olarak sınıflandırılır. Şekil 2.3'te ise k=3 için KNN yöntemi gösterilmektedir. X noktasının A, B ve C sınıflarına en yakın üç komşusunun uzaklıkları hesaplanır ve bu uzaklıklar toplamı ile en yakın sınıf

belirlenir. Bu durumda, X noktası en yakın olduğu C sınıfına atanır (Aşlıyan ve Günel, 2010).



Şekil 2.2. Birinci grafik K-en yakın komşu yöntemi



Şekil 2.3. İkinci grafik k=3 için K-en yakın komşu yöntemi (Aşlıyan ve Günel, 2010)

2.1.1.4. Naif (Naive) Bayes

Naive Bayes sınıflandırıcıları, olasılık temelli modeller olup, makine öğrenmesi ve VM alanlarında sıkça kullanılmaktadır. Bu modeller, her özelliğin birbirinden bağımsız olduğunu varsayarak Bayes teoremini temel alır. Bu varsayım gerçek dünyadaki veriler için tam olarak geçerli olmasa da Naive Bayes algoritmaları pratikte oldukça etkili sonuçlar verebilir (Russell ve Norvig , 2009).

Bu sınıflandırıcıların en büyük avantajlarından biri, hesaplama açısından çok verimli olmalarıdır. Özellikle büyük veri kümelerinde hızlı bir şekilde çalışabilirler ve yüksek boyutlu verilerde bile etkili sonuçlar üretebilirler (Rish, 2001). Modelin eğitilmesi ve tahmin yapması, daha karmaşık algoritmalara göre oldukça hızlıdır ve bu özellikleri, Naive Bayes'i gerçek zamanlı uygulamalar için ideal bir seçenek haline getirir (Zhang, 2004).

Naive Bayes algoritmaları aynı zamanda az veriyle bile iyi performans gösterebilir. Bu durum, etiketli verilerin sınırlı olduğu senaryolarda büyük bir avantaj sağlar. Örneğin, e-posta spam filtrelemede yaygın olarak kullanılırlar çünkü bu algoritma, sınırlı sayıda örnekle bile spam ve spam olmayan e-postaları etkili bir şekilde ayırt edebilir (Metsis, Androutsopoulos, ve Paliouras, 2006).

Bununla birlikte, bağımsızlık varsayımının geçerli olmadığı durumlarda Naive Bayes sınıflandırıcıları sınırlı performans gösterebilir. Özellikle özellikler arasında güçlü korelasyonlar olduğunda, modelin doğruluğu azalabilir. Ancak, bu sınırlamalara rağmen,

Naive Bayes algoritmaları genellikle birçok uygulamada tatmin edici sonuçlar verir ve özellikle başlangıç seviyesinde ve hızlı prototipleme için uygundur (Hand ve Yu, 2007).

Sonuç olarak, Naive Bayes sınıflandırıcıları, basitlikleri ve hesaplama verimlilikleri nedeniyle makine öğrenmesi uygulamalarında önemli bir yere sahiptir. Modelin temel prensiplerini ve avantajlarını anlamak, veri bilimciler ve araştırmacılar için büyük fayda sağlar (Russell ve Norvig , 2009).

2.1.1.5. Yapay sinir ağı

Neural Network (Yapay Sinir Ağı) algoritmaları, insan beyninin sinir hücrelerinin çalışma prensiplerini taklit ederek veri işleme ve modelleme yapan güçlü bir makine öğrenme yöntemidir. Bu algoritmalar, bir giriş katmanı, bir veya daha fazla gizli katman ve bir çıkış katmanından oluşan çok katmanlı bir yapıya sahiptir. Her bir katmandaki yapay nöronlar, önceki katmandan gelen bilgiyi işler ve bu bilgiyi belirli ağırlıklarla çarparak bir aktivasyon fonksiyonu yardımıyla bir sonraki katmana aktarır.

Neural Network'ler, özellikle karmaşık ve doğrusal olmayan ilişkilerin olduğu veri kümelerinde, yüksek başarı oranlarıyla dikkat çeker. Bu algoritmalar, görüntü tanıma, doğal dil işleme ve ses tanıma gibi alanlarda yaygın olarak kullanılmaktadır (Goodfellow, Bengio, ve Courville, 2016).

2.1.1.6. Multi-layer perceptron (MLP) predictor

Multi-Layer Perceptron (MLP) Predictor, yapay sinir ağlarının bir türü olarak, veri setlerindeki karmaşık ilişkileri modellemek için kullanılan güçlü bir algoritmadır. MLP, bir giriş katmanı, bir veya daha fazla gizli katman ve bir çıkış katmanından oluşur. Her bir katmandaki nöronlar, bir önceki katmandan gelen bilgiyi alır, belirli ağırlıklarla çarpar ve bir aktivasyon fonksiyonu aracılığıyla işlenmiş veriyi bir sonraki katmana iletir. Bu katmanlı yapı, MLP'nin doğrusal olmayan ilişkileri öğrenmesine ve sınıflandırma ya da regresyon problemlerinde etkili sonuçlar elde etmesine olanak tanır.

MLP, özellikle yüksek boyutlu ve karmaşık veri kümelerinde başarılı bir performans gösterir; ancak, doğru parametre seçimi ve eğitim süreci, modelin başarısını büyük ölçüde etkileyebilir. Bu nedenle, MLP'nin uygulanması genellikle dikkatli bir optimizasyon süreci gerektirir ve bu süreç, hesaplama kaynakları açısından maliyetli olabilir (Goodfellow vd., 2016).

2.1.2. Regresyon

Diğer veri seti değişkenlerine bağlı olarak sürekli değerleri tahmin etmek için regresyon analiz gerçekleştirilir. Örneğin regresyon analizi bir tahminde bulunmak için kullanılabilir. Talep arz ve diğer değişkenlere bağlı olarak ürünün gelecekteki fiyatı. Doğrusal regresyon ve lojistik regresyon en popüler iki regresyon yöntemidir. (Akpınar, 2000)

2.1.2.1. Doğrusal regresyon

Doğrusal (linear) regresyon, bağımlı bir değişken ile bir veya daha fazla bağımsız değişken arasındaki ilişkiyi modelleyen istatistiksel bir tekniktir. Bu teknik, bağımsız değişkenlerin bir doğrusal kombinasyonu olarak bağımlı değişkenin tahmin edilmesini sağlar. Doğrusal regresyonun temel varsayımlarından biri, bağımsız değişkenler ile bağımlı değişken arasındaki ilişkinin doğrusal olmasıdır (Montgomery, Peck , ve Vining, 2012).

Doğrusal regresyon, en küçük kareler yöntemi kullanılarak katsayıların tahmin edilmesiyle gerçekleştirilir. Bu yöntem, hata terimlerinin karelerinin toplamını minimize etmeyi amaçlar (Draper ve Smith, 1998).

2.1.2.2. Lojistik regresyon

Lojistik regresyon, bağımlı değişkenin iki veya daha fazla kategoriye sahip olduğu durumlarda kullanılan bir sınıflandırma yöntemidir. Bu yöntem, bağımlı değişkenin olasılıklarını modellemek için lojistik fonksiyonu kullanır (Hosmer Jr., Lemeshow, ve Sturdivant , 2013).

Lojistik regresyon modeli, maksimum olasılık yöntemi kullanılarak tahmin edilir. Bu yöntem, gözlemlenen veri altında model parametrelerinin en olası değerlerini belirlemeyi amaçlar (Long, 1997).

2.2. Kümeleme

Kümeleme, VM ve makine öğrenimi alanlarında yaygın olarak kullanılan bir tekniktir. Kümeleme, veri setindeki nesneleri veya gözlemleri, aralarındaki benzerliklere göre gruplara ayırmayı amaçlar. Bu bölümde, hiyerarşik ve hiyerarşik olmayan kümeleme yöntemleri ayrıntılı olarak ele alınacaktır. (Han vd., 2011)

2.2.1. Hiyerarşik olan kümeleme algoritmaları

Hiyerarşik kümeleme yöntemleri, veri setinde bulunan nesneleri hiyerarşik bir yapıda kümeler. Bu yöntemler, genellikle alt kümeleme veya üst kümeleme yaklaşımlarını kullanır. (Tan, Steinbach, ve Vipin , 2018)

2.2.1.1. En uzak komşu

En uzak komşu algoritması, en yakın komşu algoritmasının tersine çalışır. Bu yöntemde, kümeler arasındaki mesafe, en uzak iki nokta arasındaki mesafe olarak tanımlanır. Bu, kümelerin daha homojen ve kompakt olmasını sağlar, ancak hesaplama maliyeti yüksektir. (Jain ve Dubes , 1988)

2.2.1.2. Ortalama bağlantı

Ortalama bağlantı algoritması, kümeler arasındaki mesafeyi, tüm noktalar arasındaki ortalama mesafe olarak tanımlar. Bu yöntem hem en yakın komşu hem de en uzak komşu algoritmalarının bir kombinasyonu olarak düşünülebilir (Sokal ve Michener, 1958).

2.2.2. Hiyerarşik olmayan kümeleme algoritmaları

Hiyerarşik olmayan kümeleme yöntemleri, veri setindeki nesneleri önceden belirlenmiş sayıda küme içerisine dağıtarak çalışır. Bu yöntemlerin en yaygın olanı K-ortalama algoritmasıdır. (Jain, Murty, ve Flynn, 1999).

2.2.2.1. K-ortalama (K-Means)

K-ortalama algoritması, veri setini K adet küme ile temsil etmeyi amaçlayan hiyerarşik olmayan bir kümeleme yöntemidir. Algoritma, başlangıçta rastgele seçilen K adet merkez etrafında kümeler oluşturur ve ardından bu merkezleri güncelleyerek her bir veri noktasını en yakın merkeze atar. Bu işlem, merkezlerin konumları stabil hale gelene kadar tekrarlanır. (MacQueen, 1967) K-ortalama algoritmasının avantajları arasında, büyük veri setlerinde hızlı çalışması ve kolay uygulanabilirliği yer alır. Ancak, başlangıçta merkezlerin rastgele seçilmesi, sonuçların farklı olmasına neden olabilir. (Jain, 2010).

2.2.2.2. K-Medoids

K-Medoids algoritması, K-ortalama algoritmasına benzer, ancak merkez noktalar olarak gerçek veri noktalarını kullanır. Bu, K-Medoids algoritmasını, K-ortalama algoritmasına göre daha dayanıklı ve veri noktalarındaki uç değerlerden daha az etkilenir hale getirir (Kaufman ve Rousseeuw, 1990). K-Medoids algoritmasının avantajı, veri noktalarındaki uç değerlerden daha az etkilenmesidir. Ancak, K-ortalama algoritmasına göre daha yavaş çalışabilir. (Park ve Jun, 2009).

2.2.2.3. DBSCAN

DBSCAN (Density-Based Spatial Clustering of Applications with Noise), yoğunluk tabanlı bir kümeleme algoritmasıdır. Bu yöntem, veri noktalarının yoğun bölgelerde kümeler oluşturduğunu varsayar ve seyrek bölgelerdeki noktaları gürültü olarak kabul eder. (Ester, Kriegel, Sander, ve Xu, 1996)

2.3. Birliktelik

Birliktelik kuralları ilk kez 1993 yılında Agrawal, Imielinski ve Swami tarafından tanımlanmıştır. Bu kurallar, bazı yönleriyle sınıflandırma kurallarına benzer, ancak birliktelik analizinde sınıf değişkeni bulunmaz (Agrawal, Imieli'nski, ve Swami,1993).

Birliktelik kuralları, veri tabanlarındaki bilgiler arasındaki ilişkileri keşfeder ve bu bilgilerin diğer kayıtlarla olan bağlantılarını açıklar. VM'nin en önemli tekniklerinden biri olan birliktelik analizi, veri kümesindeki kayıtlar arasındaki bağlantıları arayan denetimsiz bir yöntemdir. Bu analiz, veri setinde eş zamanlı oluşumları, olayları ve durumları tespit etmek için kullanılır ve VM sürecinde en olası modeli ortaya çıkarmada önemli bir rol oynar. Birliktelik analizi, müşteri davranışlarını tanımlamak için kullanılan kurallar oluşturur ve bu sayede işletmelere değerli bilgiler sağlar (Kantardzic, 2011).

2.3.1. Apriori algoritması

Apriori algoritması, Agrawal ve diğerleri tarafından 1994 yılında geliştirilmiştir ve birliktelik kurallarının çıkarılmasında büyük başarı sağlamıştır. Birliktelik kuralları çıkarımında en bilinen algoritma haline gelmiştir. Apriori, boolean ilişki kuralları için geçerli bir VM algoritmasıdır ve tekrarlayan bir yapıya sahiptir. Algoritma, sık tekrar eden nesnelerin önsel bilgilerini kullanarak çalışır ve bu nedenle "apriori" adını alır (Agrawal ve Srikant, 1994).

2.4. Algoritmalarda Kullanılan Performans Metrikleri

VM algoritmalarında kullanılan tüm performans metrikleri Çizelge 2.1-2.4 arasında toplu olarak verilmiştir.

Çizelge 2.1. Sınıflandırma performans metrikleri

Performans Rate	Türkçe Karşılığı	Açıklama
Accuracy	Doğruluk Oranı	Modelin doğru sınıflandırma yapma oranı
Classification Error	Sınıflandırma Hatası	Modelin yanlış sınıflandırma yapma oranı
Kappa	Kappa	Sınıflandırma doğruluğunun rastgele tahminlere göre ayarlanmış hali
Correctly Classified Instances	Doğru Sınıflandırılan Örnekler	Doğru sınıflandırılmış örnek sayısı
Wrong Classified Instances	Yanlış Sınıflandırılan Örnekler	Yanlış sınıflandırılmış örnek sayısı
Cohen's kappa	Cohen's kappa	İki gözlemcinin sınıflandırmaları arasındaki uyum ölçüsü
Recall	Hatırlama Oranı	Doğru pozitif oranı, gerçek pozitiflerin toplamına oranı
Precision	Kesinlik	Doğru pozitif oranı, tahmin edilen pozitiflerin toplamına oranı
Sensitivity	Duyarlılık	Gerçek pozitif oranı, hastalık gibi durumların doğru tespiti
Specificity	Özgüllük	Gerçek negatif oranı, sağlıklı gibi durumların doğru tespiti
F-measure	F-ölçütü	Kesinlik ve hatırlama oranının harmonik ortalaması
TP Rate	Doğru Pozitif Oranı	Gerçek pozitif oranı, hastalık gibi durumların doğru tespiti
FP Rate	Yanlış Pozitif Oranı	Gerçek negatif oranı, sağlıklı gibi durumların yanlış tespiti
AUC	ROC Eğrisi Altındaki Alan	Modelin pozitif ve negatif sınıfları ayırt etme yeteneği
CA	Sınıflandırma Doğruluğu	Sınıflandırma doğruluk oranı
F1 Score	F1 Skoru	Kesinlik ve hatırlamanın harmonik ortalaması
Spec	Özgüllük	Gerçek negatif oranı, hastalık olmayanların doğru tespiti
LogLoss	Logaritmik Kayıp	Modelin logaritmik hata ölçüsü

Çizelge 2.2. Regresyon performans metrikleri

Performans Metrikleri	Türkçe Karşılığı	Açıklama
Absolute Error	Mutlak Hata	Tahmin edilen değer ile gerçek değer arasındaki mutlak fark
Relative Error	Göreceli Hata	Mutlak Hatanın gerçek değere oranı
Root Mean Squared Error (RMSE)	Kök Ortalama Kare Hatası	Hata karelerinin ortalamasının karekökü
Root Relative Squared Error	Kök Göreceli Kare Hatası	Hata karelerinin ortalamasının karekökünün, veri varyansına oranı
Mean Absolute Error (MAE)	Ortalama Mutlak Hata	Tahmin edilen değerler ile gerçek değerler arasındaki ortalama mutlak fark
Normalized Absolute Error	Normalize Mutlak Hata	Mutlak hatanın normalize edilmiş hali
Squared Error	Kare Hata	Hata karelerinin toplamı
Correlation	Korelasyon	Tahmin edilen değerler ile gerçek değerler arasındaki doğrusal ilişki
Correlation (R^2 Score)	Korelasyon (R^2 Score)	Modelin veri varyansını açıklama oranı
Explained Variance Score	Açıklanan Varyans Skoru	Modelin toplam varyansı ne kadar iyi açıkladığını gösterir
Margin	Marj	Tahmin edilen ve gerçek sınıflar arasındaki marjinal fark
Soft Margin Loss	Yumuşak Marj Kaybı	Yumuşak marj sınıflandırma hatası
Logistic Loss	Lojistik Kayıp	Lojistik regresyon modelindeki hata ölçüsü
Cross-Entropy	Çapraz-Entropi	Tahmin edilen ve gerçek sınıflar arasındaki farklılığı ölçer

Çizelge 2.3. Kümeleme performans metrikleri

Performans Metrikleri	Türkçe Karşılığı	Açıklama
Rand Index	Rand İndeksi	İki veri kümesi arasındaki benzerlik ölçüsü
Purity	Safılık	Kümeleme sonuçlarının doğruluk ölçüsü
Adjusted Rand Index (ARI)	Düzeltilmiş Rand İndeksi	Rand indeksinin rastgele tahminlere göre düzeltilmiş hali
Adjusted Mutual Information (AMI)	Düzeltilmiş Karşılıklı Bilgi	Karşılıklı bilginin rastgele tahminlere göre düzeltilmiş hali
Normalized Mutual Information (NMI)	Normalize Karşılıklı Bilgi	Karşılıklı bilginin normalize edilmiş hali
Homogeneity	Homojenlik	Tüm kümelerin aynı sınıfa ait olma derecesi
Completeness	Tamlık	Aynı sınıfa ait tüm öğelerin aynı kümeye atanma derecesi
V-Measure	V-Ölçütü	Homojenlik ve tamlığın harmonik ortalaması
Silhouette Score	Silüet Skoru	Kümeleme kalitesinin bir ölçüsü
Davies-Bouldin Index	Davies-Bouldin İndeksi	Kümeleme performansını değerlendirir
Fowlkes-Mallows Index (FMI)	Fowlkes-Mallows İndeksi	Kümeleme sonuçlarının doğruluk ölçüsü

Çizelge 2.4. Diğer performans metrikleri

Performans Metrikleri	Türkçe Karşılığı	Açıklama
Spearman Rho	Spearman Rho	Sıralı veri arasındaki doğrusal olmayan ilişki ölçüsü
Kendall Tau	Kendall Tau	Sıralı veri arasındaki doğrusal olmayan ilişki ölçüsü

2.5. Veri Madenciliğinde Kullanılan Programlar

VM’de en çok kullanılan araçlar: KNIME, WEKA, RapidMiner, Orange, Python, R programlama, SAS (Statistical Analysis System), MATLAB, Apache Hadoop, Apache Spark tır. Bu çalışmada; KNIME, WEKA, RapidMiner, Orange ve Python araçlarından yararlanılarak VM uygulaması gerçekleştirilmiştir.

1. KNIME

KNIME (Konstanz Information Miner), veri analizi ve VM süreçlerini görsel bir programlama dili kullanarak kolaylaştıran açık kaynaklı bir platformdur. Kullanıcı dostu arayüzü sayesinde, veri hazırlama, temizleme, modelleme ve görselleştirme gibi adımları sürükle ve bırak yöntemiyle gerçekleştirmenize olanak tanır. KNIME, genişletilebilir yapısı ve çeşitli veri kaynaklarıyla entegrasyonu sayesinde esnek bir araçtır (Berthold vd., 2009).

2. Orange

Orange, veri analizi ve görselleştirme işlemlerini basitleştiren açık kaynaklı bir yazılımdır. Görsel programlama dili kullanarak kullanıcıların veri işleme, analiz ve makine öğrenmesi modellerini oluşturma süreçlerini destekler. Orange, özellikle veri bilimi eğitiminde ve araştırmalarda yaygın olarak kullanılır ve kullanıcıların veri kümelerini hızlı ve etkili bir şekilde analiz etmelerine olanak tanır (Demsar vd., 2013).

3. Python

Python, esnekliği ve geniş kütüphane desteği ile veri bilimi ve VM’de en yaygın kullanılan programlama dillerinden biridir. Pandas, NumPy, Scikit-learn, TensorFlow ve Keras gibi kütüphaneler, veri işleme, analiz ve makine öğrenmesi modellerinin oluşturulmasını sağlar. Python, veri bilimciler ve analistler için güçlü ve çok yönlü bir araç olarak kabul edilir ve büyük veri kümeleri üzerinde karmaşık analizler yapmayı mümkün kılar (McKinney, 2010)

4. R

R, istatistiksel hesaplama ve veri analizi için geliştirilmiş açık kaynaklı bir programlama dilidir. Veri madenciliği, biyoinformatik ve veri bilimi alanlarında yaygın biçimde kullanılmaktadır. ggplot2, dplyr, tidyverse, caret ve randomForest gibi binlerce paket desteğiyle makine öğrenmesi modellerinin oluşturulması, veri görselleştirme ve istatistiksel analizlerin gerçekleştirilmesi mümkün kılınmaktadır. R, araştırmacılar ve veri bilimciler için güçlü ve esnek bir araç olarak kabul edilmekte; karmaşık analizlerin tekrarlanabilir biçimde yürütülmesine olanak tanımaktadır (Giorgi vd., 2022).

2.6. Veri Madenciliği Uygulayan Sektörler

VM'nin günümüzde kullanım yeri ile alakalı herhangi bir kısıtlama/sınırlama yoktur. Verinin biriktiği, depolandığı her yerde/alanda VM uygulaması rahatlıkla yapılabilir. İster kamu kuruluşu olsun isterse özel sektör veya ister küçük işletme olsun isterse büyük işletme olsun fark etmeksizin günümüzde birçok sektörde VM uygulamaları kolaylıkla yapılmaktadır. (Savaş , Topaloğlu , ve Yılmaz , 2012). Günümüzde VM'nin en yaygın olarak kullanıldığı alanlar aşağıda Çizelge 2.5'de verilmiştir;

Çizelge 2.5. VM'nin en yaygın olarak kullanıldığı alanlar

VM'nin en yaygın olarak kullanıldığı alanlar
Bankacılık ve sigortacılık
Pazarlama
Müşteri İlişkileri Yönetimi (MİY)
Kimya
Tıp,Biyoloji ve Genetik
Görüntü Tanıma ve Robot Görüş Sistemleri
Uzay Bilimleri ve Teknolojisi
Coğrafi Bilimler ve Yüzey Analizi Uzay Bilimleri
Meteoroloji ve Atmosfer Bilimleri
Metin Madenciliği
Sosyal Bilimler ve Davranış Bilimleri
Web Verileri Analizi
İş ve Elektronik Ticaret Verileri
Bilimsel, Mühendislik ve Sağlık Bakım Verileri

2.7. Bankacılık Sektöründe Veri Madenciliği Uygulamaları

Bankacılık sektöründe VM ve MİY, kredi risk analizi, dolandırıcılık tespiti ve pazarlama stratejileri gibi çeşitli alanlarda önemli katkılar sağlamaktadır. Bu teknikler, bankaların büyük veri setlerinden anlamlı bilgiler çıkararak stratejik kararlar almalarına yardımcı olur. Örneğin, müşteri segmentasyonu sayesinde müşteriler davranışlarına göre gruplandırılabilir ve bu da kişiselleştirilmiş pazarlama kampanyalarının oluşturulmasını sağlar (Linoff ve Berry , 2011).

Kredi risk analizinde, müşterilerin kredi ödemelerini zamanında yapma olasılıkları değerlendirilerek, bankaların daha bilinçli kredi verme kararları alması sağlanır (Edelman, Crook, ve Thomas, 2017). Dolandırıcılık tespiti için kullanılan VM teknikleri, anormal işlem desenlerini belirleyerek potansiyel dolandırıcılık vakalarını erken tespit eder (Bolton ve Hand, 2002).

3. MÜŞTERİ İLİŞKİLERİ YÖNETİMİ ALANINDA VERİ MADENCİLİĞİ UYGULAMALARI

3.1. MİY’de Veri Madenciliğinin Yeri

Günümüzde MİY, yaygın bir şekilde kullanılan bir kavram olarak karşımıza çıkmaktadır. İyi bir yöntem ile oluşturulmuş MİY sistemi, müşterileri en iyi şekilde tanıma, anlama ve müşterileri analiz ederek onların ilgi alanlarını bilmeyi gerektirir (Buttle & Maklan, 2019). MİY, müşterilerin talep ve beklentilerini ortaya koyduktan sonra analiz edip uygun yöntemi oluşturup geliştirmeyi de içine almaktadır.

VM uygulama yöntemleri, MİY tekniklerini oluşturmada önemli bir role sahiptir. Ancak, VM uygulamalarıyla şirketlerin oluşturulan büyük veri tabanlarında toplanan kayıtlar anlamlı bilgilere dönüştürülebildiklerinde daha faydalı bir yol izlenecektir (Swift R. , 2001). VM yöntemleri sayesinde, toplanan bu veriler içerisinde saklanan bilgiler keşfedilerek kurumsal rekabet, başarı, verim maliyet gibi etkenlerin iyileştirilmesi açısından kritik bir önem taşımaktadır (Rust, Lemon, ve Zeithaml, 2004). Aşağıda metinde vurgulanan bazı önemli noktalar MİY'in VM'deki yerinin vurgulayan maddeler bulunmaktadır. Bunlar;

Müşteri Profili Oluşturma: Firma bünyesinde geçmiş dönemlerde müşterilere ait verilerin toplanmasıyla müşteri profilleri oluşturulabilmektedir. Toplanan bu veriler müşterilerin yaş, cinsiyet, gelir düzeyi, eğitim seviyesi, meslek, coğrafi konum gibi faktörleri içeren demografik özelliklerin yanı sıra yaşam tarzı, alışveriş sıklığı gibi özelliklerini içermektedir (Gartner, 2022). Müşteri gruplarına özel oluşturulan bu segmentasyonlar, pazarlama ve satış stratejileri için önemli bir temel oluşturabilir (Swift R. , 2001).

Kampanya Yönetimi: VM, müşteri davranışlarını ve geçmiş alışveriş alışkanlıklarını analiz ederek müşteriye özel kampanyalar oluşturmak için kullanılabilir. Bu durum, müşteriye daha kişiselleştirilmiş teklifler sunmanın yolunu göstererek; müşterilerin kendilerini biricik hissetmelerini sağlayabilir (Buttle ve Maklan, 2019).

Müşteri Sadakati Belirleme: Müşteri sadakati, müşteri verilerinden yola çıkarak geçmiş davranışlara dayalı müşteri profili oluşturma ile adımı atılarak belirlenebilir (Rust vd., 2004). VM, müşteri sadakatini etkileyen faktörleri bulmak, anlamak ve müşteri sadakatini artırmak için yöntem ve stratejiler geliştirme aşamalarını kullanarak müşteriye şirkete bağlayabilir (Peppers ve Rogers, 2011).

Sepet Analizi: Bu tür analizler, promosyonlarda hedef müşteri profilinin belirlenmesi ve ürünlerin en uygun düzenle yerleşimi gibi karar destek süreçlerine önemli katkı sağladığı tespit edilmiştir (Swift R. , 2001).

Müşteri Bölümlendirme: VM toplanan veriler ışığında, müşterileri belirli özelliklere göre homojen gruplara ayırarak müşteri bölümlendirme sürecine katkı sağlayabilir. Bu durum, müşterilerle olan iletişim konusunda şirket çalışanlarına yardımcı olarak daha etkili pazarlama stratejileri uygulamayı sağlar (Gartner, 2022).

Müşteri Davranış Modelleri: Bu tür modeller, müşterinin kültürel yapısı, alışveriş alışkanlıkları, yaşam tarzı gibi faktörleri içerek müşteri hakkında bilgiye ulaşmayı hedefleyen ve pazarlama ile satış uygulamalarında önemli bir yol gösterici olan rehberdir (Buttle ve Maklan, 2019).

Müşteri Memnuniyeti Analizi: VM, müşteri geri bildirimlerini, şikayetlerini, taleplerini ve etkileşimlerini analiz ederek müşteri memnuniyetini değerlendirebilir. Bu durum, müşteri hizmetlerini geliştirmek ve müşteri memnuniyetini artırmak için stratejiler ve yöntemler oluşturmaya yardımcı olarak verimi optimize edebilir (Rust vd., 2004).

Sonuç olarak, VM, MİY uygulamalarında müşteri ilişkilerini yönetmek ve iş stratejilerini optimize etmek için kullanılan çok yönlü etkili ve güçlü bir araçtır. Müşteri verilerinin analiz edilmesi, müşteri beklentilerini ve taleplerini daha iyi anlamak ve

müşterilere daha etkili bir yöntem ve stratejilerle hizmet verebilmek için önemli bir adımdır (Gartner, 2022).

3.2. MİY ve Veri Madenciliği Alanında Yapılan Bazı Çalışmalar

Nguyen ve Mutum, MİY’de VM uygulamalarının önemini vurgulayan çalışmalarında, VM tekniklerinin müşteri segmentasyonu ve sadakat programları gibi alanlarda nasıl kullanıldığını ele almışlardır. Bu çalışma, MİY stratejilerinin iyileştirilmesinde VM’nin rolünü detaylandırmıştır (Nguyen ve Mutum, 2024).

Wang ve Zhou, yapayzeka kullanarak müşteri sadakat programlarının optimizasyonunu incelemişlerdir. Çalışmaları, müşteri davranışlarını analiz etmek için makine öğrenimi algoritmalarının kullanımına odaklanmıştır. Bu, modern MİY stratejileri için önemli bir adımdır (Wang ve Zhou, 2023).

Rodpysh ve arkadaşları, VM’nin MİY üzerindeki etkilerini araştırmışlardır. Eski çalışmaları baz alarak, büyük veri tabanlarındaki değerli müşterileri tespit etme ve geliştirme stratejileri geliştirmişlerdir. Bu çalışma, MİY ve VM arasındaki etkileşimi anlamada önemli bir katkı sağlamıştır (Rodpysh, 2022).

Babu ve Bhuvaneswari, VM tekniklerini kullanarak MİY sistemlerinin verimliliğini artırmayı hedeflemişlerdir. Müşterilerin geçmiş verilerini analiz ederek en karlı müşterileri belirlemiş ve bu müşterileri elde tutma stratejileri geliştirmişlerdir (Babu ve Bhuvaneswari, 2021).

Wang ve Pang, bir alışveriş merkezinin işlem verilerini kullanarak Apriori ve Fuzzy algoritmalarını uygulamışlardır. Bu çalışma, müşteri davranışlarını analiz ederek yeni müşteriler çekme ve mevcut müşterileri elde tutma stratejilerini geliştirmiştir (Wang ve Pang, 2021).

Peyman ve Safanaz, bilgi yönetimi (BY), MİY ve VM arasındaki ilişkiyi incelemişlerdir. Demografik verileri kullanarak istatistiksel analizler yapmış ve MİY ve VM arasındaki bağlantıları ortaya koymuşlardır (Peyman ve Safanaz, 2020).

Khajvand ve Tarokh, değerli müşteri segmentlerini analiz etmek için kümeleme algoritmalarını kullanmışlardır. Bu çalışma, müşteri demografik özelliklerini kullanarak MİY stratejilerinin iyileştirilmesine katkıda bulunmuştur (Khajvand ve Tarokh, 2019).

Wang ve Wu, banka sektöründe müşteri ilişkilerini incelemek için VM tekniklerini kullanmışlardır. Demografik verileri analiz ederek müşteri davranışlarını ve finansal ürün alım sıklıklarını tahmin etmişlerdir (Wang ve Wu, 2018).

Boginski ve Butensko, apraz korelasyon ve kmeleme tekniklerini kullanarak hisse senedi piyasalarının yapısal zelliklerini analiz etmiřlerdir ve finansal ıkarımlarda bulunmuřlardır (Boginski ve Butensko, 2017).

Liu, Shue ve Lee, internet servis saėlayıcı (IPS) endstrisinde kullanıcıların genel aė kullanma rntlerini analiz etmek iin nitelik odaklı indirgeme yntemini kullanmıřlardır. Bu, mřteri iliřkilerini glendirecek stratejiler geliřtirmeye yardımcı olmuřtur (Liu, Shue, ve Lee, 2016).

Liu ve Shih, AHP ve birliktelik kurallarını birleřtirerek mřteri segmentasyon modelleri geliřtirmiřlerdir. Mřteri yařam mr deėerlerine gre blmlere ayırarak, mřteri elde tutma stratejilerini iyileřtirmiřlerdir (Liu ve Shih, 2015).

Chen ve arkadařları, perakende endstrisinde etkin pazarlama kampanyaları dzenlemek amacıyla davranıřsal VM yapmıřlardır. Bu alıřma, mřteri davranıřlarını hedef alarak pazarlama stratejilerini geliřtirmiřtir (Chen J. , 2015).

Wang ve Hung, VM teknikleri ile bir telekomnikasyon řirketi iin mřteri karlılık ynetim sistemi geliřtirmiřlerdir. Bu, karlı mřteri segmentlerine odaklanarak MİY stratejilerini iyileřtirmiřtir (Hung ve Wang, 2014).

Last ve Kandel (2013), karar aėaları algoritmasını kullanarak yarı iletken endstride retim planlama uygulamaları gerekleřtirmiřlerdir (Last ve Kandel, 2013).

Hsieh (2012), sinir aėları ve birliktelik kurallarını kullanarak bir bankanın kredi kartı mřterileri iin davranıřsal skorlama modeli oluřturmuřtur (Hsieh, 2012).

Hwanng, Jung ve Suh (2011), lojistik regresyon, sinir aėları ve karar aėaları kullanarak mřteri yařam mr analiz modeli geliřtirmiřlerdir (Hwang, Jung, ve Suh, 2011).

3.3. Bankacılık Sektrnde MİY ve Veri Madenciliėi

Bankacılık sektrnde VM ve MİY entegrasyonu, son yıllarda byk nem kazanmıřtır. VM, bankaların mřteri verilerini analiz ederek daha bilinli kararlar almasına olanak tanırken, MİY stratejileri mřteri memnuniyetini ve sadakatini artırmayı hedeflemektedir. Bu entegrasyon, bankaların mřteri ihtiyalarını daha iyi anlamalarına ve hizmetlerini kiřiselleřtirmelerine yardımcı olmaktadır.

Wang ve Zhou (2023), yapay zeka ve VM kullanarak mřteri sadakat programlarının optimizasyonunu incelemiř ve bu teknolojilerin modern MİY stratejilerinde nasıl kullanıldıėını gstermiřlerdir (Wang ve Zhou, 2023). Aynı řekilde, Nguyen ve Mutum

(2024), VM tekniklerinin müşteri segmentasyonu ve sadakat programları üzerindeki etkilerini vurgulamışlardır. Bu çalışmalar, bankacılık sektöründe VM ve MİY entegrasyonunun müşteri memnuniyetini artırmadaki önemini ortaya koymaktadır (Nguyen ve Mutum, 2024).

Rodpysh ve arkadaşları (2022), büyük veri tabanlarındaki değerli müşterileri tespit etmek ve geliştirmek için VM'nin nasıl kullanılabileceğini araştırmışlardır. Bu çalışma, VM ve MİY arasındaki etkileşimi anlamada bankacılık sektörüne önemli bir katkı sağlamıştır (Rodpysh, 2022). Ayrıca, Babu ve Bhuvaneswari (2021), banka sektöründe VM tekniklerini kullanarak MİY sistemlerinin verimliliğini artırmayı başarmışlardır (Babu ve Bhuvaneswari, 2021).

Khajvand ve Tarokh (2019), müşteri segmentasyonunu kümeleme algoritmaları kullanarak analiz etmişler ve bu yöntemin müşteri demografik özelliklerine dayalı MİY stratejilerinin iyileştirilmesinde nasıl fayda sağladığını göstermişlerdir. Bu, bankaların müşteri verilerini daha etkili bir şekilde analiz ederek, müşteri memnuniyetini artırmak ve daha etkili pazarlama stratejileri geliştirmek için VM tekniklerinden nasıl yararlanabileceğini göstermektedir (Khajvand ve Tarokh, 2019).

Bulut (2019), VM algoritmalarının MİY süreçlerini optimize etmek için nasıl kullanılabileceğini araştırmış ve bu süreçte yapay zeka modellerinin entegrasyonu ile daha verimli sonuçlar elde edilebileceğini önermiştir. Bu modelin uygulanmasıyla, hedef müşteri kitlesinin daha az maliyetle belirlenebileceği ve bankacılık ürünlerinin müşterilere daha hızlı bir şekilde sunulabileceği ifade edilmektedir (Bulut, 2019).

Hassani ve arkadaşları (2018), bankacılık sektöründe 100 farklı VM uygulamasını analiz etmişlerdir. Bu çalışmada, VM tekniklerinin genellikle müşteri memnuniyeti artırma ve pazarlama faaliyetlerinde yoğunlaştığı belirtilmektedir. Ancak, bankacılık sektöründe VM potansiyeli taşıyan ve henüz tam anlamıyla keşfedilmemiş birçok alanın mevcut olduğuna da vurgu yapılmaktadır. Araştırmada ayrıca, VM uygulamalarının verilere erişim konusunda bazı sınırlamalarla karşılaştığı da belirtilmiştir (Hassani, Huang, & Silva, 2018).

Miyan (2017) ise VM tekniklerinin bankacılıkta müşteri segmentasyonu, karlılık yönetimi ve risk yönetimi gibi kritik fonksiyonlarda nasıl kullanılabileceğini incelemiştir. Araştırmanın bulgularına göre, VM sayesinde sahtecilik olaylarının daha kolay tespit edilebildiği ve MİY'nin bu yolla daha etkili hale getirilebildiği ortaya konulmuştur (Miyan, 2017).

Çalış ve Baynal (2016), bankacılık sektöründe kümeleme yöntemlerinden biri olan k-ortalamlar yöntemini kullanarak müşterileri 12 farklı gruba ayırmışlardır. Bu çalışmada, farklı demografik özelliklere sahip müşteri gruplarının ödeme ve gelir durumlarının tespit edilmesi, bu gruplara yönelik özel kredi ürünleri ve diğer bankacılık hizmetlerinin geliştirilmesine olanak tanımıştır (Çalış & Baynal, 2016).

Chitra ve Subashini (2013), bankacılık sektöründe VM uygulamalarını analiz ederek, bu tekniklerin özellikle sahtecilik tespiti, müşteri tutundurma ve pazarlama stratejilerinde güçlü bir araç olarak kullanılabileceğini ortaya koymuşlardır. Ayrıca, VM yöntemlerinin kredi başvurularının doğru şekilde değerlendirilmesi ve müşteri segmentlerine özel ürünlerin pazarlanmasında önemli rol oynadığı belirtilmiştir (Chitra ve Subashini, 2013).

Moin ve Ahmed (2012), VM uygulamaları sayesinde bankaların doğru hedef kitleyi belirleyebildiğini ve yeni müşteriler kazanabildiğini belirtmektedir. Araştırma, ayrıca, sahtecilik tespitinin anında yapılabilmesi, müşteri davranışlarının analiz edilmesi ve müşteri bağlılığı artırma çalışmalarında VM'nin sağladığı avantajlara dikkat çekmektedir (Moin ve Ahmed, 2012).

Bhambri (2011), bankacılık sektöründe VM'nin rekabet avantajı sağladığını vurgulamaktadır. Araştırmaya göre, VM uygulamaları MİY, pazarlama, risk yönetimi, sahtecilik tespiti ve yazılım desteği gibi alanlarda bankalara büyük faydalar sunmaktadır (Bhambri, 2011).

Seyrek ve Ata (2010), VM teknikleri kullanarak bankacılık sektöründe hangi finansal performans göstergelerinin finansal başarıyı etkilediğini incelemişlerdir. Bu çalışmada, Toplam Krediler/Toplam Mevduat oranı ile Diğer Faaliyet Gelirleri/Toplam Faaliyet Gelirleri oranlarının bankaların finansal performansı üzerinde belirleyici olduğu tespit edilmiştir (Seyrek ve Ata, 2010).

Albayrak (2009), bankaların finansal performansını değerlendirmek için diskriminant analizi, karar ağaçları ve lojistik regresyon yöntemlerini karşılaştırmalı olarak incelemiş ve karar ağaçlarının bankacılık sektöründe etkili bir sınıflandırma yöntemi olduğunu belirtmiştir (Albayrak, 2009).

Genel olarak literatür incelendiğinde, bankacılık sektörüne yönelik VM araştırmalarının büyük çoğunluğunun tanımlayıcı nitelikte olduğu ve daha çok MİY, pazarlama, kredi değerlendirme ve sahteciliğin önlenmesi üzerine yoğunlaştığı görülmektedir.

4. UYGULAMA

4.1. İşin Anlaşılması

Bu aşamada, veri setindeki banka şirketinin iş durumu hakkında bilgi edinilmiş, temel beklenti ve isteklerin ne olduğu tespit edilmiş ve ihtiyaçlar belirlenmiştir. VM süreçleri ile banka müşteri verileri üzerinde uygulama yapılırken 4 problemin analizleri yapılmış, yorumlanmış ve sonuçlar karşılaştırılmıştır, gerekli çıkarımlarda bulunarak stratejiler belirlenmiş, önerilerde bulunulmuştur. Bu bağlamda araştırma için geliştirilen problemler aşağıdaki gibidir:

1. Müşteri borcunu ödeyip ödemeyeceğini tahmin eden bir sistem geliştirme.
2. Müşterilerin ne kadar zamanda ödeme yapacağını (erken ya da geç) tahmin eden bir sistem geliştirme.
3. Müşteriye verilebilecek krediyi tahmin eden bir sistem geliştirme.
4. Müşteri segmentasyonu oluşturma.

4.2. Verilerin Anlaşılması

Verilerin anlaşılması aşamasında verilerin elde edilmesi, verileri tanımaya ve anlamaya yönelik analizlerin yapılması ve aynı zamanda ilk izlenimde verilerde saklı olabilecek bilgiler için bilgilerin oluşturulması gerçekleştirilir.

Bu araştırmada bu aşamada veri seti <https://www.kaggle.com/> sitesinden sağlanmıştır. Kaggle; veri bilimciler ve makine öğrenimi ile ilgilenenler tüm kullanıcılar için çevrimiçi bir platformdur. Kaggle kullanıcıları bu çevrimiçi topluluğa ücretsiz olarak birçok veri seti sunmaktadır. Banka Şirketine ait veri seti bu platformdan sağlanmıştır. Kullanmak üzere en çok kullanılan VM yazılımları araştırılmış ve Weka, RapidMiner, Knime, Orange, Python, Excel, VM uygulama yazılımları olarak bulunmuştur. Akabinde programlar ile uygulanan algoritmalar karşılaştırılmıştır.

4.3. Verilerin Hazırlanması

Verilerin hazırlanması aşaması, işlenmemiş veriden başlayarak son adımdaki oluşan veriye kadar yapılması gereken düzenlemeleri içermektedir. Ham verinin hazırlanması, oluşturulan tablolar, kayıtlar, verilerin dönüşümü ve modelleme araçları için temizleme, düzeltme gibi özellikleri içermektedir. Bu aşamada da aşağıdaki adımlar sırasıyla gerçekleştirilmiştir.

4.3.1. Kurulum aşaması

Kullanılacak olan veriler <https://www.kaggle.com/> daki bir banka şirketinin veri tabanından csv formatında alınmıştır. Açık kaynaklardan ve üniversite kütüphane sistemine Knime, Weka, RapidMiner, Orange, Python, R programları indirilmiştir ve kurulumları gerçekleştirilmiştir. Çizelge 4.1 ' de veri seti excel görüntüsü verilmiştir.

Çizelge 4.1. Loan Veri seti Excel görüntüsü

Loan_ID	loan_status	Principal	terms	effective_date	due_date	paid_off_time	past_due_days	age	education	Gender
xqd20166231	PAIDOFF	1000	30	9.08.2016	10.07.2016	9/14/2016 19:31		45	High School or Below	male
xqd20168902	PAIDOFF	1000	30	9.08.2016	10.07.2016	10.07.2016 09:00		50	Bechalor	female
xqd20160003	PAIDOFF	1000	30	9.08.2016	10.07.2016	9/25/2016 16:58		33	Bechalor	female
xqd20160004	PAIDOFF	1000	15	9.08.2016	9/22/2016	9/22/2016 20:00		27	college	male
xqd20160005	PAIDOFF	1000	30	9.09.2016	10.08.2016	9/23/2016 21:36		28	college	female
xqd20160706	PAIDOFF	300	7	9.09.2016	9/15/2016	9.09.2016 13:45		35	Master or Above	male
xqd20160007	PAIDOFF	1000	30	9.09.2016	10.08.2016	10.07.2016 23:07		29	college	male
xqd20160008	PAIDOFF	1000	30	9.09.2016	10.08.2016	10.05.2016 20:33		36	college	male
xqd20160909	PAIDOFF	1000	30	9.09.2016	10.08.2016	10.08.2016 16:00		28	college	male

4.3.2. Verilerin açıklanması

Veri setindeki verilerin alanları ve açıklamaları Çizelge 4.2’ de verilmiştir.

Çizelge 4.2. Banka kredi müşterileri veri seti alanları ve açıklamaları

Alan Adı	Alan Açıklaması
Loan_id	Müşteri benzersiz numarası
Loan_status	Kredi statülerin bulunduğu alandır
Principal	Müşteri ana parası
Term	Müşterinin kaç günlük vade ile kredi aldığı bilgisi
Effective_date	Kredi tarihi başlangıcı
Due_date	Ödeme tarihi
Paid_off_time	Ödenenler için ödeme tarihi ve zamanı
Past_due_date	Kredinin ne kadar geç ödendiği bilgisi
Age	Müşteri yaşı
Education	Müşteri eğitim durumu
Gender	Müşteri cinsiyeti

4.3.3. Verilerin temizlenmesi ve düzenlenmesi aşaması

Verilerin temizlenmesi ve düzenlenmesi aşamasında, verilerin excel ile ilk kontrolleri gerçekleştirilmiştir. Verilerdeki boş alanların bütünlüğüne bakılarak tutarlı ve mantıklı olduğu sonucuna varılmıştır. Veriler programlarda (Knime, WEKA, RapidMiner, Orange, Python, R) çalıştırıldığında eksik verilerle ilgili algoritma çalışmadıysa eğer eksik veriler üzerinde gerekli işlemler yapılmıştır.

4.3.4. Verilerin değerlendirilmesi

Bu aşamada; veri sayısı, alan sayısı, alanlarına göre dolu hücre sayısı, veri tipleri ve hafıza kapladığı değerler Python programı kullanılarak Şekil 4.1’de görüntülenmiştir.

```

RangeIndex: 500 entries, 0 to 499
Data columns (total 11 columns):
#   Column                Non-Null Count  Dtype  
---  -
0   Loan_ID                500 non-null    object  
1   loan_status            500 non-null    object  
2   Principal              500 non-null    int64   
3   terms                 500 non-null    int64   
4   effective_date         500 non-null    object  
5   due_date              500 non-null    object  
6   paid_off_time          400 non-null    object  
7   past_due_days          200 non-null    float64  
8   age                   500 non-null    int64   
9   education              500 non-null    object  
10  Gender                 500 non-null    object  
dtypes: float64(1), int64(3), object(7)
memory usage: 43.1+ KB

```

Şekil 4.1. Veri setinin Python’da veri tipleri ve veri sayıları çıktısı

Veriler kategorik ve sayısal veriler olarak iki başlıkta değerlendirilmiştir.

4.3.4.1. Kategorik verilerin değerlendirilmesi

Kategorik verilerin; sayısal değerleri Çizelge 4.3’de verilmiştir.

Çizelge 4.3. Kategorik değişkenlerin sayısal değerleri

<i>Kategorik Alanlar</i>	<i>Gender</i>	<i>Kategorik Alanlar</i>	<i>loan_status</i>	<i>Kategorik Alanlar</i>	<i>education</i>
female	77	COLLECTION	100	Bechalor	67
male	423	COLLECTION_PAIDOFF	100	College	220
Genel Toplam	500	PAIDOFF	300	High School or Below	209
		Genel Toplam	500	Master or Above	4
				Genel Toplam	500

- Paidoff : Borcunu ödeyen müşteriler
- Collection : Borcu ödenmemiş olan müşteriler
- Collection_paidoff : Gecikmeli ödemeler

Kadınların yaş ortalaması genellikle erkeklere göre daha yüksek olup, bu fark özellikle COLLECTION_PAIDOFF durumunda belirgindir. Eğitim seviyesi arttıkça PAIDOFF durumuna geçme olasılığı artmakta, özellikle yüksek eğitim seviyesindeki bireyler daha yaşlı olduğu gözlenmektedir. Eğitim seviyesi arttıkça kredi ana miktarları da artmakta,

özellikle kolej eğitimi olan borçluların tüm kredi durumlarında en yüksek ana miktarlara sahip olduğu gözlemlenmiştir.

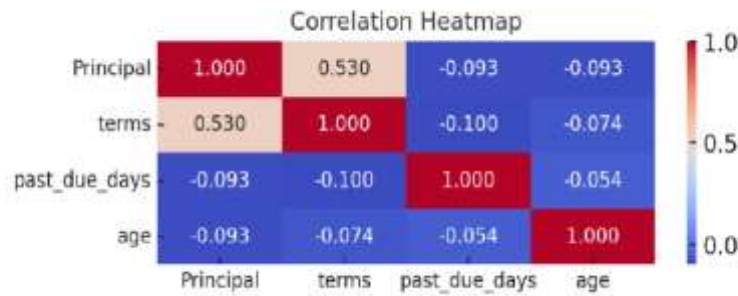
4.3.4.2. Veri setindeki sayısal değerlerin değerlendirilmesi

Veri setindeki sayısal değerlerin; betimsel istatistik çizelgeleri Çizelge 4.4’de verilmiştir.

Çizelge 4.4. Nümerik değişkenler arasındaki alanların değerleri

<i>Alanlar Değerler</i>	<i>count</i>	<i>Mean</i>	<i>Std</i>	<i>min</i>	<i>25%</i>	<i>50%</i>	<i>75%</i>	<i>max</i>
Principal	500.0	943.200	115.240274	300.0	1000.0	1000.0	1000.0	1000.0
Terms	500.0	22.824	8.000064	7.0	15.0	30.0	30.0	30.0
past_due_days	200.0	36.010	29.380880	1.0	3.0	37.0	60.0	76.0
Age	500.0	31.116	6.084784	18.0	27.0	30.0	35.0	51.0

Anapara miktarı ve gecikme günleri verileri daha geniş bir aralıkta dağılırken, vade süresi ve yaş verilerinin daha homojen olarak dağıldığı görülmektedir. Veri setindeki sayısal değişkenler arasındaki ilişkileri görsel olarak temsil etmek için korelasyon analizi oluşturulmuştur ve ısı haritasındaki değerler Şekil 4.2’de görülmektedir.



Şekil 4.2. Sayısal değerlerin korelasyon analiz değerleri ısı matrisi

Korelasyon ısı haritası incelendiğinde, sayısal değişkenler arasındaki ilişkiler hakkında şu gözlemler yapılabilir.

Principal ve terms : Bu iki değişken arasında pozitif bir korelasyon vardır (0.53). Bu, kredi anapara miktarı arttıkça kredi vadesinin de artma eğiliminde olduğunu göstermektedir.

Diğer Değişkenler : Diğer değişkenler arasındaki korelasyonlar oldukça düşüktür. Örneğin `past_due_days` ve `age` arasındaki korelasyon -0.05 gibi düşük bir değerdedir. Bu da bu iki değişkenin neredeyse bağımsız olduğunu göstermektedir.

Genel Korelasyon Düzeyleri : Genel olarak, veri setindeki sayısal değişkenler arasında güçlü korelasyonlar gözlenmemektedir. Bu durumda, değişkenlerin birbirinden bağımsız olma eğiliminde olduğu söylenebilir.

Vade süresi arttıkça, yaş dağılımı da genellikle artmaktadır. Yaş arttıkça, kredi durumlarına bağlı vade süreleri genellikle birleşmektedir. Eğitim seviyesi arttıkça, kredi vade süreleri genellikle artmaktadır. Yaş arttıkça, kredi vade süreleri genellikle artmaktadır. Genç yaş gruplarında (18-29) “Principal” değerleri daha düşük ve dağılımları daha genişken, yaş arttıkça “Principal” değerleri genellikle artmaktadır ve dağılımları daralmaktadır. Anapara değeri arttıkça, yaş dağılımı da genellikle artmaktadır.

4.4. Birinci Problem Modelleme

Birinci problem şu konuya yanıt aramaktadır: Müşteri borcunu ödeyip ödemeyeceğini tahmin eden bir sistem geliştirme. Bu sorunun çözümü kredi risk analizi yaparken oldukça yararlı olabileceği düşünülmektedir.

Müşterilerin borcunu ödeyip ödemeyeceğini tahmin eden bir model oluşturmak için `loan_status` sütunu hedef değişken olarak kullanılmıştır. Bu sütun, müşterinin kredisinin ödeme durumunu göstermektedir. `Loan_status` alanı detaylı incelendiğinde ise üç farklı kategorik statü mevcuttur.

PAIDOFF : Bu kategori, kredinin tam olarak ödendiğini ve herhangi bir gecikme yaşanmadan kredi süresi içinde borcun kapatıldığı göstermektedir. Müşteriler bu kategoride, taahhüt ettikleri tarihlerde borçlarını tamamen ödemişlerdir. PAIDOFF

durumundaki kredilerde ilgili elde edilen temel istatistiksel bilgiler ise Çizelge 4.5’de özetlenmiştir.

Çizelge 4.5. PAIDOFF durumundaki kredilerle ilgili özet bilgileri

<i>Alan Adı (Türkçe Açıklama)</i>	<i>Ayrıntılar</i>
Loan_ID (Kredi Kimlik Numarası)	300 adet kredi
Principal (Ana Para Miktarı)	300 ila 1000 dolar arası, Ortalama ~935 dolar, Çoğunlukla 800 ile 1000 dolar arası
terms (Vade Süresi)	7 ile 30 gün arası, Ortalama ~22.1 gün, Çoğunlukla 30 gün
effective_date (Etkili Tarih)	En sık başlangıç tarihi: 9/11/2016
paid_off_time (Ödeme Tarihi)	Çoğu kredi, başlangıç tarihinden kısa bir süre sonra ödenmiş
age (Yaş)	19 ile 51 yaş arası, Ortalama ~31.2 yaş
education (Eğitim Seviyesi)	En yaygın: Üniversite (college) (133 kredi), ardından Lise veya Altı
Gender (Cinsiyet)	Erkek %82

COLLECTION: Bu kategori, ödeme süresince borcun tahsil edilemediğini ve borcun ödenmesi gereken tarihi geçtiği halde hala ödenmediğini ifade eder. Bu durumda, borç genellikle bir tahsilat şirketine devredilir ve müşteri üzerindeki finansal baskı artar. Buna göre bu kategori, kredi ödeme performansı açısından riskli olarak değerlendirilir. Bu durumdaki krediler için paid_off_time bilgisi eksiktir. Çünkü bu krediler henüz ödenmemiştir. Bu, kredinin tahsil sürecinde olduğunu ve müşterinin ödeme yapmadığını göstermektedir. COLLECTION durumundaki kredilerle ilgili elde edilen temel istatistiksel bilgiler ise Çizelge 2.6’de özetlenmiştir.

Çizelge 4.6. COLLECTION durumundaki kredilerle ilgili özet bilgileri

<i>Alan Adı (Türkçe Açıklama)</i>	<i>Ayrıntılar</i>
Loan_ID (Kredi Kimlik Numarası)	100 adet kredi
Principal (Ana Para Miktarı)	800 ile 1000 dolar arası, Ortalama ~954 dolar
terms (Vade Süresi)	15 ile 30 gün arası, Ortalama ~23.85 gün
effective_date (Etkili Tarih)	En sık rastlanan tarih: 9/11/2016
due_date (Bitiş Tarihi)	Genellikle 2016-10-06 civarında
past_due_days (Gecikme Gün Sayısı)	28 ile 76 gün arası, Ortalama ~62.86 gün

COLLECTION_PAIDOFF : Bu kategori, borcun ödenmesi gereken süre geçtikten sonra borcun ödendiğini ifade eder. Genellikle borç tahsilat sürecine girdikten sonra müşterinin borcunu kapattığı durumlar için kullanılır. Bu durum, ödeme gecikmesine rağmen sonunda müşterinin finansal yükümlülüğünü yerine getirdiğini göstermektedir. Ancak yine de bu, bir gecikme yaşandığını ve potansiyel bir kredi riski olduğunu göstermektedir. COLLECTION_PAIDOFF durumundaki kredilerle ilgili elde edilen temel istatistiksel bilgiler ise Çizelge 4.7’ de özetlenmiştir.

Çizelge 4.7. COLLECTION PAIDOFF durumundaki kredilerle ilgili özet bilgileri

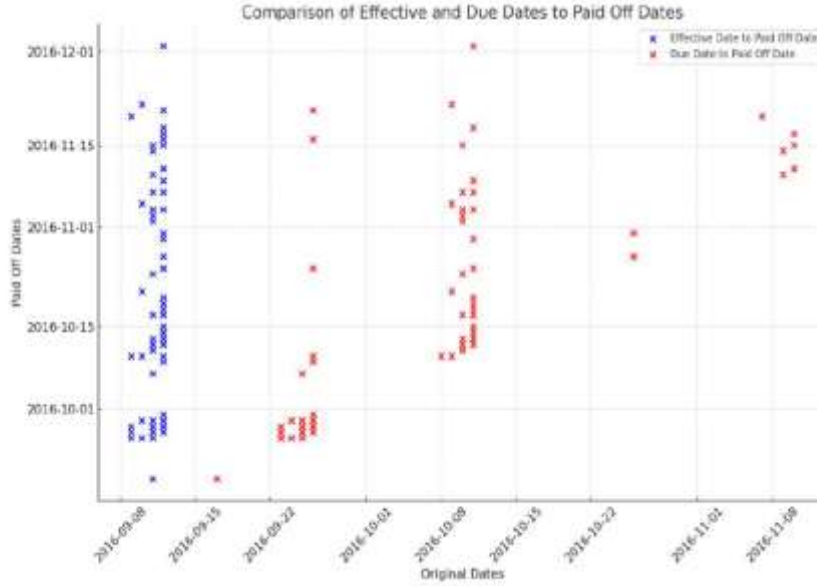
<i>Alan Adı (Türkçe Açıklama)</i>	<i>Ayrıntılar</i>
Loan_ID (Kredi Kimlik Numarası)	100 adet kredi
Principal (Ana Para Miktarı)	800 ile 1000 dolar arası, Ortalama ~954 dolar
terms (Vade Süresi)	15 ile 30 gün arası, Ortalama ~23.85 gün
effective_date (Etkili Tarih)	En sık rastlanan tarih: 9/11/2016
due_date (Bitiş Tarihi)	Genellikle 2016-10-06 civarında
past_due_days (Gecikme Gün Sayısı)	28 ile 76 gün arası, Ortalama ~62.86 gün

Bu durumdaki krediler için paid_off_time bilgisi, effective_date ve due_date tarihlerinden sonra olacaktır. Çünkü bu krediler geç ödenmiştir. Bu durum, borcun tahsilat sürecinde olduğu ancak sonunda ödendiği anlamına gelir. Çizelge 4.8’ de ise ödeme süresi geçtikten sonra borçlarını ödeyen müşterilerin bilgileri verilmektedir.

Çizelge 4.8. Ödeme süresi geçtikten sonra borçlarını ödeyen müşterilerin detayları

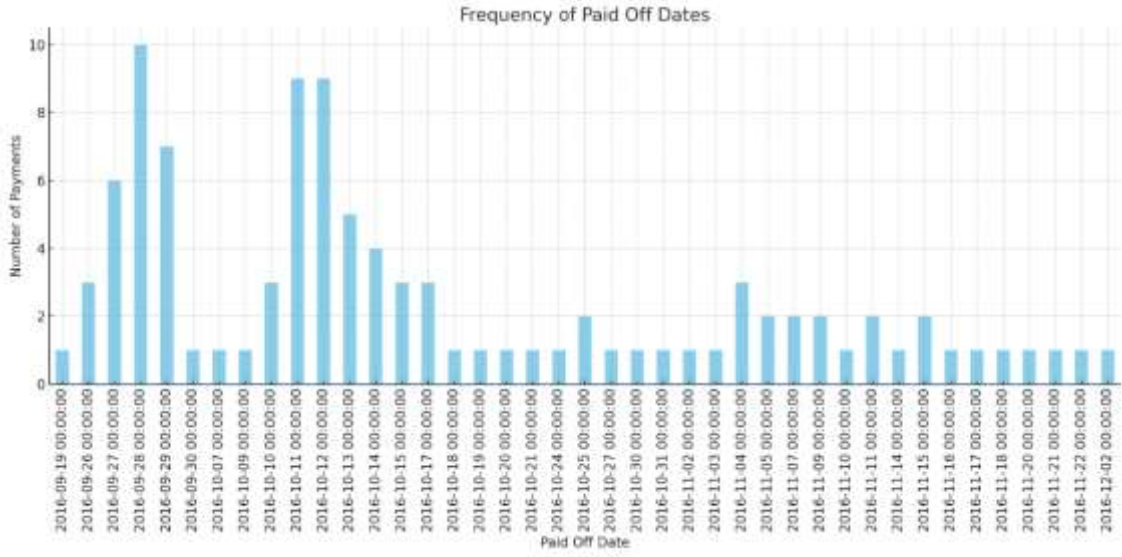
<i>Loan_ID</i>	<i>Loan Status</i>	<i>Effective Date</i>	<i>Due Date</i>	<i>Paid Off Time</i>
xqd20160401	COLLECTION_ PAIDOFF	9.09.2016	8.10.2016	10.10.2016 11:45
xqd20160402	COLLECTION_ PAIDOFF	9.09.2016	23.09.2016	27.09.2016 17:00
xqd20320403	COLLECTION_ PAIDOFF	9.09.2016	7.11.2016	20.11.2016 14:10
xqd20160404	COLLECTION_ PAIDOFF	9.09.2016	23.09.2016	28.09.2016 15:38
xqd20190405	COLLECTION_ PAIDOFF	9.09.2016	23.09.2016	26.09.2016 17:22

Çizelge 4.8’ de görülen her bir giriş, borcun planlanan bitiş tarihinden sonra ödendiği zamanı göstermektedir. Bu da tahsilat süresinde borçların kapatıldığını belirtir. Şekil 4.3’de COLLECTION_PAIDOFF durumundaki krediler için effective_date ve due_date ile paid_off_date tarihlerini karşılaştırılması verilmiştir.



Şekil 4.3. COLLECTION_PAIDOFF durumundaki krediler için effective_date, due_date ve paid_off_date tarihlerinin karşılaştırılması

Şekil 4.3 incelendiğinde şu durumlar gözlemlenmiştir. Eğer bazı paid_off_date değerleri due_date’den çok sonra gerçekleşiyorsa, bu durum ödeme performansında ciddi sıkıntılar olduğunu gösterebilir. Bu tür durumlar, belirli müşteri segmentlerinde veya dönemlerde kredi politikalarının gözden geçirilmesini gerektirebilir. Ödeme yoğunluklarını belirlemek için ise, paid_off_date üzerindeki verileri analiz ederek, hangi tarihlerde yoğun ödeme yapıldığı Şekil 4.4’de görülebilir.



Şekil 4.4. Paid_off_date frekans grafiği

Grafiğe ve verilere bakarak, ödemelerin yoğun olduğu günler belirlenmiştir. İşte en yoğun ödeme yapılan günler:

- 28/09/2016: 10 ödeme
- 11/10/2016 ve 12/10/2016: Her birinde 9 ödeme
- 29/09/2016: 7 ödeme
- 27/09/2016: 6 ödeme
- Diğer tarihlerde daha az ödeme yapılmıştır.

Bu tarihler, borçların kapandığı ana zaman dilimlerini yansıtmakta ve bu dönemlerdeki ödeme yoğunluğu, müşteri davranışları veya belki de özel tahsilat çabaları ile ilgili olduğu söylenebilir. Ödeme yoğunluğu yüksek tarihler, mali yılsonu, tatil dönemleri gibi özel zamanlar olabilir veya promosyonlar ve ödeme kolaylıkları gibi faktörler tarafından tetiklenmiş de olabilmektedir.

Anomali Tespiti : Anomali tespiti, veri setindeki olağandışı, beklenmedik veya norm dışı gözlemleri saptamak için kullanılan bir yöntemdir (Chandola, Banerjee, ve Kumar, 2009). Anomali tespiti yapmak için, paid_of_date verileri inceleyerek normalden sapma gösteren ödeme tarihleri belirlenmiştir. Bu tür sapmaları belirlemek için çeşitli yöntemler kullanılabilir. Ancak burada basit bir istatistiksel yaklaşım olan Z-Skoru yöntemi kullanılmıştır. Z-Skoru, veri noktalarının ortalama değerden ne kadar standart sapma

uzaklıkta olduğunu göstermektedir (Montgomery ve Runger, 2018). Genellikle, Z-Skoru ± 2.5 veya ± 3 dışında kalan değerler anomali olarak kabul edilir.

Bu analiz için önce ödeme tarihlerinin frekanslarını hesaplayıp, ardından bu frekansların Z-Skorları hesaplanmıştır. Anomali olarak kabul edilen değerler belirlendiğinde anomali tespiti sonuçlarına göre; üç tarih ödeme yoğunluğu açısından belirgin şekilde normalin dışında kalmış ve Z-Skoru eşiği ± 2.5 'in üzerindedir.

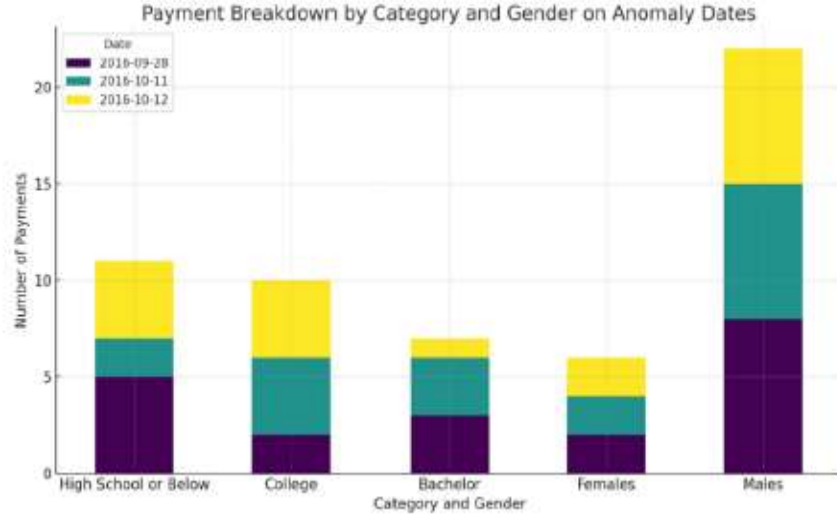
- 28/09/2016: Z-Skoru=3.11
- 11/10/2016: Z-Skoru=2.70
- 12/10/2016: Z-Skoru=2.70

Bu tarihlerde yapılan ödemeler, genel ödeme yoğunluklarından önemli ölçüde sapmaktadır ve bu olağandışı bir durum veya özel bir etkinlik olduğuna işaret edebilir. Bu tarihlerdeki ödeme kayıtlarını detaylı olarak incelemek için veri seti filtrelenmiş ve bu ödemelerin diğer özellikleri gözden geçirilmiştir. Öncelikle, 28/09/2016, 11/10/2016 ve 12/10/2016 tarihlerinde yapılan ödemeleri incelenmiştir. 28/09/2016, 11/10/2016 ve 12/10/2016 tarihlerinde yapılan ödemelerle ilgili bazı gözlemler Çizelge 4.9'de verilmiştir.

Çizelge 4.9. 28/09/2016, 11/10/2016 ve 12/10/2016 tarihlerinde yapılan ödemelerle ilgili bazı gözlemler

<i>Tarih</i>	<i>Toplam Ödemeler</i>	<i>Lise veya Ahi</i>	<i>Üniversite</i>	<i>Lisans</i>	<i>Kadınlar</i>	<i>Erkekler</i>
28.09.2016	10	5	2	3	2	8
11.10.2016	9	2	4	3	2	7
12.10.2016	9	4	4	1	2	7

Şekil 4.5'de belirtilen tarihlerde yapılan ödemelerin, eğitim seviyesi ve cinsiyet açısından dağılımını göstermektedir.



Şekil 4.5. 28/09/2016, 11/10/2016 ve 12/10/2016 tarihlerinde eğitim seviyelerine ve cinsiyete göre ödeme dağılımının grafiği

Bu ödemelerin, özellikle erkekler arasında ve belirli eğitim düzeylerinde (özellikle lisans ve üniversite mezunları arasında) yoğunlaştığını görülmektedir.

İlk problem için ön işleme adımları aşağıdaki şekilde verilmiştir:

Eksik Değerler: paid_off_time ve past_due_days kolonlarında eksik değerler vardır. Bu, kredilerin henüz ödenmemiş veya vadesi geçmiş olabileceğini göstermektedir.

paid_off_time: 100 kayıt eksik. Bu, kredinin henüz ödenmemiş olabileceğini göstermektedir. Değerlerin boş olması normal ve beklenen bir durumdur

past_due_days: 300 kayıt eksik. Ne kadar geç ödendi bilgisini içermektedir. Eğer kredi zamanında ödenmişse bu alan boş bırakılmıştır. Bu yüzden bu sütundaki eksik değerleri 0 ile doldurmak uygun bir yaklaşım olacaktır. Çünkü vadesi geçmiş gün sayısı yoktur.

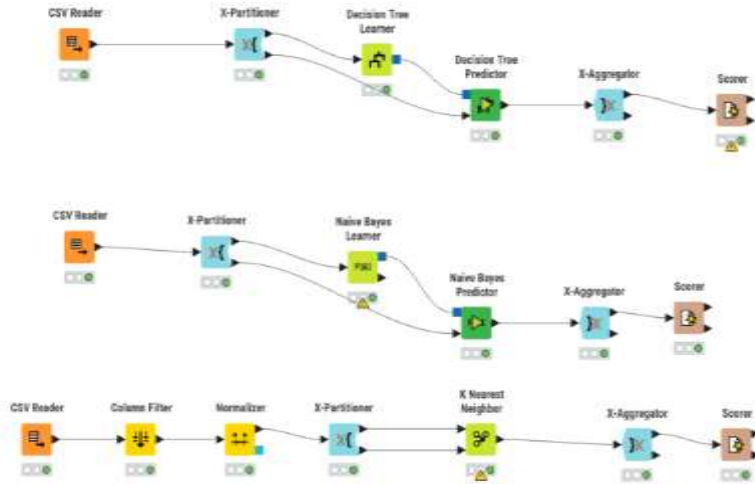
Tarih Dönüşümleri: effective_date, due_date ve paid_off_time kolonlarını tarih formatına çevirilmelidir.

Kategorik Değişkenler: education ve gender gibi kategorik değişkenlerin modelleme için uygun hale getirilmesi gerekebilmektedir.

Hedef Değişken: loan_status kolonu, modelin hedef değişkeni olarak kullanılmıştır. Bu kolonun içerdiği değerleri ve dağılımını kontrol edilmiştir.

4.4.1. Birinci problemin Knime ile çözümü

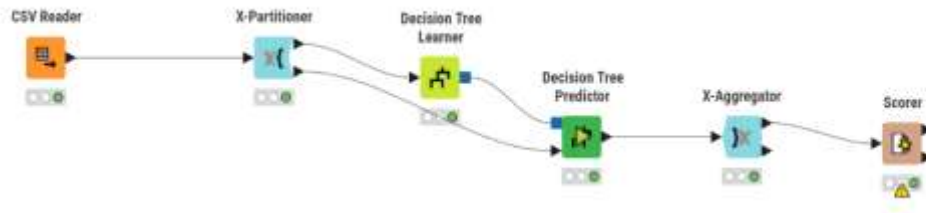
Bu bölümde, Knime programını kullanarak müşteri borcunu ödeyip ödemeyeceğini tahmin eden bir sistem geliştirilmeye çalışılmış ve Decision Tree, Naive Bayes, K Nearest Neighbor sınıflandırma algoritmaları kullanılmıştır. Uygulanan bu üç algoritma için en iyi sonuç veren algoritma bulunmuş, sonuçlar karşılaştırılarak yorumlanmıştır. Model eğitimi ve testi için; Knime’da cross-validation (çapraz doğrulama) yöntemi birçok farklı değer ile denenmiş ve en iyi sonuç veren, veri setini eşit parçalara bölecek olan fold değeri bulunmuştur. Şekil 4.6’da ise Knime da problemin çözümü için uygulanan üç algoritmanın akış şeması yer almaktadır.



Şekil 4.6. Birinci problem için Knime ile tüm algoritmaların akış şeması

4.4.1.1. Birinci problem için Knime ile decision tree algoritması çözümü

Bu bölümde, Knime programı kullanılarak Decision Tree Algoritması ile müşteri borcunu ödeyip ödemeyeceğini tahmin eden bir sistemin geliştirilmesine yönelik elde edilen sonuçlar yorumlanmış ve modelin süreç adımları Şekil 4.7’ de verilmiştir.



Şekil 4.7. Birinci problem için Knime ile Decision Tree algoritması akış şeması

En iyi sonuç veren test ve eğitim oranı doğrulama sayısı cross validation = 42 olarak verildiği için analizde karşılaştırma yapabilmek adına uygulanan değer kullanılmıştır. Çizelge 4.10'da birinci problem Knime programında Decision Tree algoritması performans metrikleri aşağıdaki çizelgede görülmektedir.

Çizelge 4.10. Birinci problem için Knime ile Decision Tree algoritması performans metrikleri

Performans Metrikleri	Performans Değerleri
Kesinlik (Accuracy)	59,36%
Hata (Error)	40.64%
Doğru Sınıflandırılmış (Correct classified)	295
Yanlış Sınıflandırılmış (Wrong classified)	202
Cohen's kappa (k)	0.06714

Yukarıdaki performans metrikleri incelendiğinde modelin doğruluk oranının (Accuracy) %59,36 olduğu, buna karşın hata oranının (Error) %40,64 ile oldukça yüksek bir düzeyde seyrettiği görülmektedir. Özellikle 0,06714 olarak hesaplanan Cohen's Kappa değeri, modelin başarısının rastlantısal tahmin düzeyine yakın olduğuna ve sınıflandırma performansının yetersiz kaldığına işaret etmektedir. Toplam 497 örneklem içerisinde 295 gözlem doğru sınıflandırılırken 202 gözlem yanlış sınıflandırılmıştır. Birinci probleme ilişkin KNIME programında Karar Ağacı (Decision Tree) algoritmasıyla elde edilen diğer performans metrikleri Çizelge 4.11'de sunulmaktadır.

Çizelge 4.11. Birinci problem için Knime ile Decision Tree algoritması performans metrikleri 2

Loan_status Kategorileri	Recall	Precision	Sensitivity	Specifity	F-measure	Accuracy	Cohen's kappa
PAIDOFF	0.943	0.62	0.943	0.136	0.748	-	-
COLLECTION	0.14	0.359	0.14	0.937	0.201	-	-
COLLECTION_PAIDOFF	0	0	0	0.987	-	-	-
Overall	-	-	-	-	-	0.594	0.067

Replikasyon çalışmasından elde edilen bulgular, orijinal tezin sonuçlarıyla karşılaştırıldığında bazı belirgin farklılıklar göze çarpmaktadır. Orijinal çalışmada genel doğruluk oranı %63,8 ve Cohen's Kappa değeri 0,313 olarak raporlanmışken, replikasyon analizinde bu değerler sırasıyla %59,4 ve 0,067 olarak elde edilmiştir. Sınıf bazında incelendiğinde, orijinal tezde PAIDOFF kategorisi için Recall %78,5 ve Precision %69 olarak bildirilmiş olup replikasyonda Recall değerinin %94,3 düzeyine yükseldiği görülmektedir; bu durum modelin söz konusu sınıfa olan yatkınlığının daha belirgin bir hâl aldığına işaret etmektedir. COLLECTION kategorisinde orijinal çalışmada gözlemlenen görece dengeli performansın replikasyonda korunamadığı, COLLECTION_PAIDOFF kategorisinde ise orijinal tezde düşük de olsa mevcut olan Recall ve Precision değerlerinin replikasyonda sıfıra gerilediği görülmektedir. Bu sonuçlar, modelin temel eğilimlerini yeniden üretmekle birlikte orijinal çalışmayla tam bir metrik uyumu sağlanamadığını ortaya koymaktadır. Söz konusu tutarsızlığın, veri ön işleme adımlarındaki olası farklılıklardan ya da parametre ayarlarındaki küçük sapmalardan kaynaklandığı değerlendirilmektedir. Çizelge 4.12 ise birinci problem Knime programında Decision Tree algoritması ile elde edilen diğer performans metrikleri aşağıdaki çizelgede görülmektedir.

Çizelge 4.12. Birinci problem için Knime ile Decision Tree algoritması performans metrikleri 3

Loan_status Kategorileri	True Positives	False Positives	True Negatives	False Negatives
PAIDOFF	281	172	27	17
COLLECTION	14	25	372	86
COLLECTION_PAIDOFF	0	5	393	99

Karmaşıklık matrisi sonuçları incelendiğinde, modelin PAIDOFF kategorisinde doğru pozitif (TP) değerleri açısından görece iyi bir performans sergilediği, ancak yanlış pozitif (FP) ve yanlış negatif (FN) değerlerinin de yüksek seviyelerde seyrettiği görülmektedir.

COLLECTION kategorisinde doğru negatif (TN) değerlerinin yüksek olmasına karşın doğru pozitif (TP) ve yanlış negatif (FN) değerlerinin düşük kalması, modelin bu kategoriye girmeyen örnekleri başarıyla ayırt edebildiğini ancak gerçek COLLECTION vakalarını tanımada yetersiz kaldığını ortaya koymaktadır. COLLECTION_PAIDOFF kategorisinde de benzer bir tablo gözlemlenmekte olup yüksek doğru negatif değerlerine rağmen doğru pozitif ve yanlış negatif değerlerinin oldukça düşük seyretmesi dikkat çekmektedir. Genel bir değerlendirme yapıldığında, modelin PAIDOFF kategorisinde diğer kategorilere kıyasla daha iyi bir performans ortaya koyduğu, buna karşın COLLECTION ve COLLECTION_PAIDOFF kategorilerinde doğru sınıflandırma oranlarının düşük, yanlış sınıflandırma oranlarının ise yüksek olduğu tespit edilmiştir. Bu bulgular, referans tezde elde edilen sonuçlarla büyük ölçüde örtüşmektedir. Şekil 4.8’de ise birinci problem Knime programında Decision Tree algoritması confusion matrix görülmektedir.



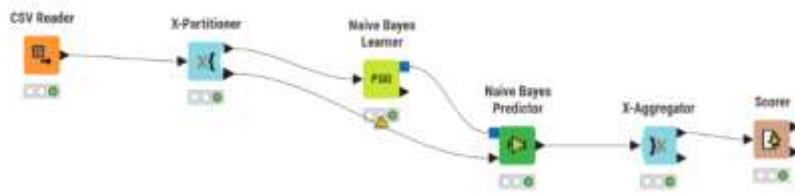
Şekil 4.8. Birinci problem için Knime ile Decision Tree Confusion Matrix

Karmaşıklık matrisi değerleri incelendiğinde, referans tezde PAIDOFF kategorisinde 234, COLLECTION kategorisinde 45 ve COLLECTION_PAIDOFF kategorisinde 38 doğru sınıflandırmanın gerçekleştirildiği görülmektedir. Replikasyon çalışmasında ise PAIDOFF kategorisinde 281 doğru sınıflandırmayla daha yüksek bir TP değerine ulaşılmış; ancak COLLECTION ve COLLECTION_PAIDOFF kategorilerinde bu sayılar sırasıyla 14 ve 0 olarak kalmıştır. Dolayısıyla replikasyon modelinin azınlık sınıflarını ayırt etme kapasitesinin belirgin biçimde daha düşük olduğu anlaşılmaktadır. Her iki

çalışmada da PAIDOFF kategorisinin görece daha iyi performans sergilediği, yüksek yanlış pozitif ve yanlış negatif oranlarının ise modelin doğruluğunu olumsuz yönde etkilediği görülmektedir. Cohen's Kappa değerinin her iki modelde de düşük seyretmesi, sınıflandırma başarısının rastlantısal tahmin düzeyinin sınırlı ölçüde üzerinde kaldığına işaret etmektedir.

4.4.1.2. Birinci problem için Knime ile Naive Bayes algoritması çözümü

Bu bölümde, Knime programını kullanılarak Naive Bayes Algoritması ile müşteri borcunu ödeyip ödemeyeceğini tahmin eden bir sistemin geliştirilmesine yönelik elde edilen sonuçlar yorumlanmış ve modelin süreç adımları Şekil 4.9.'da verilmiştir.



Şekil 4.9. Birinci problem için Knime ile Naive Bayes algoritması akış şeması

Referans tezde en iyi sonucu veren doğrulama yöntemi 10 katlı çapraz doğrulama (Cross Validation = 10) olarak belirlendiğinden, karşılaştırma yapılabilmesi amacıyla bu çalışmada da aynı parametre benimsenmiştir. Birinci probleme ilişkin KNIME programında Naive Bayes algoritmasıyla elde edilen performans metrikleri Çizelge 4.13'te sunulmaktadır.

Çizelge 4.13. Birinci problem için Knime ile Naive Bayes algoritması performans metrikleri

Performans Metrikleri	Performans Değerleri
Kesinlik (Accuracy)	0.694
Hata (Error)	0.306
Doğru Sınıflandırılmış (Correct classified)	347
Yanlış Sınıflandırılmış (Wrong classified)	153
Cohen's kappa (k)	0.542

Naive Bayes algoritmasına ilişkin performans metrikleri incelendiğinde, replikasyon çalışmasından elde edilen sonuçların referans tezdeki değerlerin belirgin biçimde

üzerinde seyrettiği görülmektedir. Doğruluk oranı referans tezde %59,8 olarak raporlanmışken, bu çalışmada %69,4 olarak elde edilmiş; buna bağlı olarak hata oranı da %40,2'den %30,6'ya gerilemiştir. Doğru sınıflandırılan örnek sayısı açısından da referans tezdeki 299 gözleme karşılık 347 gözlem doğru sınıflandırılmıştır. En dikkat çekici farklılık ise Cohen's Kappa değerinde gözlemlenmektedir. Referans tezde 0,052 olarak hesaplanan bu değer, istatistiksel olarak yok hükmünde bir uyuma karşılık gelirken, replikasyon çalışmasında elde edilen 0,542 değeri orta düzey bir uyuma işaret etmekte ve modelin sınıflar arasındaki örüntüleri rastlantısal tahminin ötesinde öğrenebildiğini ortaya koymaktadır. Çizelge 4.14 ise birinci problem Knime programında Naive Bayes algoritması ile elde edilen diğer performans metrikleri aşağıdaki çizelgede görülmektedir.

Çizelge 4.14. Birinci problem için Knime ile Naive Bayes algoritması performans metrikleri 2

Loan_status Kategorileri	Recall	Precision	Sensitivity	Specifity	F-measure	Accuracy	Cohen's kappa
PAIDOFF	0.533	0.976	0.533	0.98	0.69	-	-
COLLECTION	0.93	0.487	0.93	0.755	0.639	-	-
COLLECTION_PAIDOFF	0.94	0.648	0.94	0.873	0.767	-	-
OVERALL	-	-	-	-	-	0.694	0.542

Sınıf bazında değerlendirildiğinde, referans tezdeki modelin PAIDOFF kategorisinde 0,96 gibi yüksek bir duyarlılık (Recall) sergilediği ancak 0,613 düzeyindeki kesinlik (Precision) oranının yüksek sayıda yanlış pozitif tahmine yol açtığı görülmektedir. Replikasyon modelinde ise PAIDOFF duyarlılığı 0,533 olarak gerçekleşmiş, buna karşın 0,976 düzeyindeki kesinlik oranı yapılan tahminlerin güvenilirliğini önemli ölçüde artırmıştır. Azınlık sınıfları açısından bakıldığında, referans tezdeki modelin COLLECTION kategorisinde 0,1 ve COLLECTION_PAIDOFF kategorisinde 0,01 gibi oldukça düşük duyarlılık değerleriyle bu sınıfları tanımada yetersiz kaldığı görülmektedir. Replikasyon modelinde ise bu değerler sırasıyla 0,93 ve 0,94 düzeyine ulaşmış olup azınlık sınıflarının tanınmasında belirgin bir iyileşme sağlanmıştır. Genel doğruluk oranının referans tezdeki %59,8 seviyesinden %69,4'e yükselmesi ve Cohen's Kappa değerinin 0,052'den 0,542'ye çıkması, replikasyon modelinin sınıflar arasındaki örüntüleri yalnızca çoğunluk sınıfına odaklanmak yerine daha dengeli bir biçimde öğrenebildiğine işaret etmektedir. Çizelge 4.15 ise birinci problem Knime programında Naive Bayes algoritması ile elde edilen diğer performans metrikleri aşağıdaki çizelgede görülmektedir.

Çizelge 4.15. Birinci problem için Knime ile Naive Bayes algoritması performans metrikleri 3

Loan_status Kategorileri	True Positives	False Positives	True Negatives	False Negatives
PAIDOFF	160	4	196	140
COLLECTION	93	98	302	7
COLLECTION_PAIDOFF	94	51	349	6

Karmaşıklık matrisi değerleri incelendiğinde, referans tezdeki modelin PAIDOFF kategorisinde başarılı bir tespit performansı sergilediği ancak yüksek yanlış pozitif oranıyla hatalı tahminler ürettiği görülmektedir. Replikasyon modelinde ise yanlış pozitif sayısının yalnızca 4 olduğu, buna karşın 140 yanlış negatif değeriyle bu sınıfın bir bölümünün ıskalandığı görülmektedir. COLLECTION kategorisinde referans tezdeki yüksek yanlış negatif oranının aksine, replikasyon modelinin 93 doğru pozitif ve yalnızca 7 yanlış negatif ile tahsilat kredilerini yakalamada belirgin biçimde daha başarılı olduğu anlaşılmaktadır. Referans tezde model için en zorlayıcı kategori olarak değerlendirilen COLLECTION_PAIDOFF sınıfında ise replikasyon modeli 94 doğru pozitif ve yalnızca 6 yanlış negatif üreterek bu kategoride kayda değer bir iyileşme sağlamıştır. Şekil 4.10'da birinci problemi Knime programı ile Naive Bayes algoritması ile elde edilen Confusion Matrix verilmiştir.



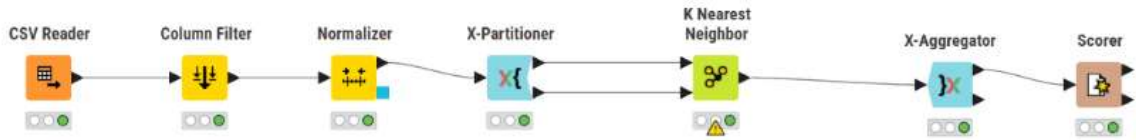
Şekil 4.10. Birinci problem için Knime ile Naive Bayes Confusion Matrix

Karmaşıklık matrisi genel olarak değerlendirildiğinde, referans tezdeki modelin PAIDOFF sınıfını tanımada başarılı olduğu ancak COLLECTION ve COLLECTION_PAIDOFF sınıflarını ayırt etmede ciddi güçlükler yaşadığı görülmektedir. Replikasyon modelinde ise bu tablo tersine dönmüş; COLLECTION sınıfında 93, COLLECTION_PAIDOFF sınıfında 94 doğru tespit ile her iki azınlık sınıfında %90'ın üzerinde bir yakalama başarısı elde edilmiştir. Buna karşın PAIDOFF

sınıfına ait 300 gözlemin 140'ının yanlış sınıflandırılması, replikasyon modelinin asıl zorlandığı noktanın çoğunluk sınıfı olduğuna işaret etmektedir. İstatistiksel açıdan bakıldığında, referans tezdeki %59,8 doğruluk oranı replikasyon çalışmasında %69,4'e yükselmiş; Cohen's Kappa değeri ise 0,052'den 0,542'ye çıkarak modelin rastlantısal tahmin düzeyinin belirgin biçimde ötesine geçtiğini ortaya koymaktadır.

4.4.1.3. Birinci problem için Knime ile K nearest neighbor algoritması çözümü

Bu bölümde KNIME programı kullanılarak K-En Yakın Komşu (K Nearest Neighbor) algoritması ile müşteri borcunu ödeyip ödemeyeceğini tahmin eden bir sistemin geliştirilmesine yönelik elde edilen sonuçlar yorumlanmış ve modelin süreç adımları Şekil 4.11'de sunulmaktadır.



Şekil 4.11. Birinci problem için Knime ile K Nearest Neighbor algoritması akış şeması

Referans tezde en iyi sonucu veren doğrulama sayısı $K = 54$ olarak belirlendiğinden, karşılaştırma yapılabilmesi amacıyla bu çalışmada da aynı değer benimsenmiştir. Birinci probleme ilişkin KNIME programında K-En Yakın Komşu (K Nearest Neighbor) algoritmasıyla elde edilen performans metrikleri Çizelge 4.16'da sunulmaktadır.

Çizelge 4.16. Birinci problem için Knime ile K Nearest Neighbor algoritması performans metrikleri

Performans Metrikleri	Performans Değerleri
Kesinlik (Accuracy)	0.582
Hata (Error)	0.418
Doğru Sınıflandırılmış (Correct classified)	291
Yanlış Sınıflandırılmış (Wrong classified)	209
Cohen's kappa (k)	0.023

Performans metrikleri incelendiğinde, doğruluk oranının %58,2 ve hata oranının %41,8 olarak referans tezdeki değerlerle birebir örtüştüğü görülmektedir. Toplam 500 gözlem içerisinde 291'inin doğru, 209'unun ise yanlış sınıflandırılması, her iki modelin de aynı karakteristik yapıyı sergilediğini ortaya koymaktadır. Cohen's Kappa değerinin referans tezde 0,029 olarak raporlanmasına karşın bu çalışmada 0,023 olarak hesaplanması, her iki modelin de rastlantısal tahmin düzeyinin çok az üzerinde kaldığını doğrulamakta ve KNN algoritmasının bu veri seti üzerindeki sınırlı ayrıştırma kapasitesinin replikasyon çalışmasında da tutarlı biçimde tekrarlandığını göstermektedir. Çizelge 4.17'de ise birinci problem için Knime programı ile K Nearest algoritması ile elde edilen diğer performans metrikleri verilmiştir

Çizelge 4.17. Birinci problem için Knime ile K Nearest Neighbor algoritması performans metrikleri 2

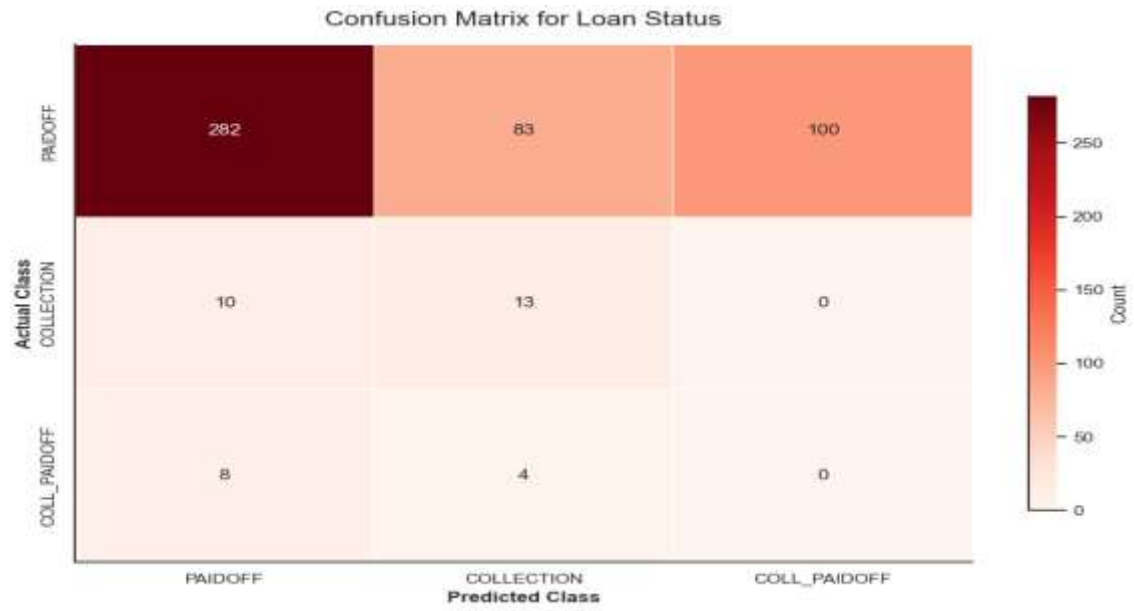
Loan_status Kategorileri	Recall	Precision	Sensitivity	Specifity	F-measure	Accuracy	Cohen's kappa
PAIDOFF	0.94	0.606	0.94	0.085	0.737	-	-
COLLECTION	0.03	0.273	0.03	0.98	0.054	-	-
COLLECTION_PAIDOFF	0.06	0.25	0.06	0.955	0.097	-	-
Overall	-	-	-	-	-	0.582	0.023

Sınıf bazında incelendiğinde, modelin PAIDOFF kategorisinde 0,94 duyarlılık oranıyla bu vakaları yakalamada başarılı olduğu, ancak 0,085 gibi düşük bir özgüllük değeriyle diğer sınıflara ait gözlemleri de PAIDOFF olarak etiketleme eğiliminde olduğu görülmektedir. COLLECTION ve COLLECTION_PAIDOFF kategorilerinde ise duyarlılık değerleri sırasıyla 0,03 ve 0,06 düzeyinde kalarak modelin azınlık sınıflarını tanımda yetersiz kaldığı anlaşılmaktadır. Doğruluk oranının %58,2 ve Cohen's Kappa değerinin 0,023 olması, genel sınıflandırma performansının rastlantısal tahmin düzeyine yakın seyrettiğine işaret etmektedir. Çizelge 4.18 birinci problem için Knime programı ile K Nearest algoritmasının diğer performans metrikleri verilmiştir.

Çizelge 4.18. Birinci problem için Knime ile K Nearest Neighbor algoritması performans metrikleri 3

Loan_status Kategorileri	True Positives	False Positives	True Negatives	False Negatives
PAIDOFF	282	183	17	18
COLLECTION	3	8	392	97
COLLECTION_PAIDOFF	6	18	382	94

Model PAIDOFF kategorisinde yüksek doğru pozitif sayısına ulaşmış olmakla birlikte, yanlış pozitif ve yanlış negatif değerlerinin de yüksek seyretmesi bu kategorideki başarının sınırlı kaldığına işaret etmektedir. COLLECTION ve COLLECTION_PAIDOFF kategorilerinde ise doğru pozitif sayılarının oldukça düşük, yanlış tahmin oranlarının ise yüksek olması modelin azınlık sınıflarını sınıflandırmada yetersiz kaldığını göstermektedir. Şekil 4.12 ise birinci problem için Knime programında K Nearest Neighbor algoritması ile elde edilen confusion matrix görülmektedir.



Şekil 4.12. Birinci problem için Knime ile K Nearest Neighbor Confusion Matrix

Karmaşıklık matrisi incelendiğinde, modelin PAIDOFF kategorisinde 282 doğru pozitif (TP) ile sayısal açıdan yüksek bir tespit performansı sergilediği, ancak 183 yanlış pozitif (FP) değerinin diğer sınıflara ait gözlemlerin bu kategoriye dahil edildiğine işaret ettiği görülmektedir. COLLECTION ve COLLECTION_PAIDOFF kategorilerinde ise sırasıyla yalnızca 3 ve 6 doğru pozitif elde edilmiş; her iki sınıftaki yüksek yanlış negatif sayıları (97 ve 94) modelin azınlık sınıflarını ayırt etmede başarısız olduğunu ortaya koymaktadır. Bu bulgular, referans tezdeki sonuçlarla tutarlılık göstermektedir.

4.4.1.4. Knime birinci problemin algoritmalarının karşılaştırılması

Bu bölümde birinci problem kapsamında KNIME programında uygulanan algoritmalar karşılaştırmalı olarak ele alınmış ve müşteri borcunu ödeyip ödemeyeceğini tahmin etmede en iyi performansı sergileyen algoritma belirlenmiştir. Banka kredi müşteri

verileri üzerinde uygulanan tüm algoritmalara ilişkin performans metrikleri Çizelge 4.19'da sunulmaktadır.

Çizelge 4.19. Birinci problem için Knime ile tüm algoritmaların performans metrikleri

Algoritma	Accuracy	Error	Correct Classified	Wrong Classified	Cohan's Kappa (K)
Decision Tree	0.5935	0.4065	295	202	0.0671
Naive Bayes	0.694	0.306	347	153	0.5425
K-Nearest Neighbor	0.582	0.418	291	209	0.0233

Algoritma karşılaştırma sonuçları incelendiğinde, model tercihinin referans tezdeki sıralamadan ayrıştığı görülmektedir. Referans tezde %63,78 doğruluk ve 0,313 Kappa değeriyle en başarılı algoritma olarak belirlenen Karar Ağacı (Decision Tree), bu çalışmada %59,35 doğruluk ve 0,067 Kappa değeriyle tezdeki performansının belirgin biçimde gerisinde kalmıştır. Buna karşın referans tezde %59,8 doğruluk ve 0,052 Kappa değeriyle sınırlı bir başarı sergileyen Naive Bayes algoritması, bu çalışmada %69,4 doğruluk ve 0,542 Kappa değeriyle en yüksek performansa ulaşmıştır. K-En Yakın Komşu (KNN) algoritması ise her iki çalışmada da en düşük performansı sergileyen algoritma olmuş; doğruluk oranı her iki çalışmada da %58,2 olarak gerçekleşmiş, Kappa değerleri ise sırasıyla 0,029 ve 0,023 olarak hesaplanmıştır.

Genel bir değerlendirme yapıldığında, referans tez Karar Ağacı algoritmasını tercih ederken bu çalışmada en yüksek doğruluk oranı (%69,4), en düşük hata sayısı (153) ve en yüksek Cohen's Kappa değeri (0,542) ile Naive Bayes algoritmasının öne çıktığı görülmektedir. Kappa değerindeki belirgin fark, Naive Bayes modelinin rastlantısal tahmin düzeyinin anlamlı ölçüde ötesine geçtiğini ortaya koymakta ve bu veri seti özelinde daha güvenilir bir sınıflandırma aracı olduğuna işaret etmektedir.

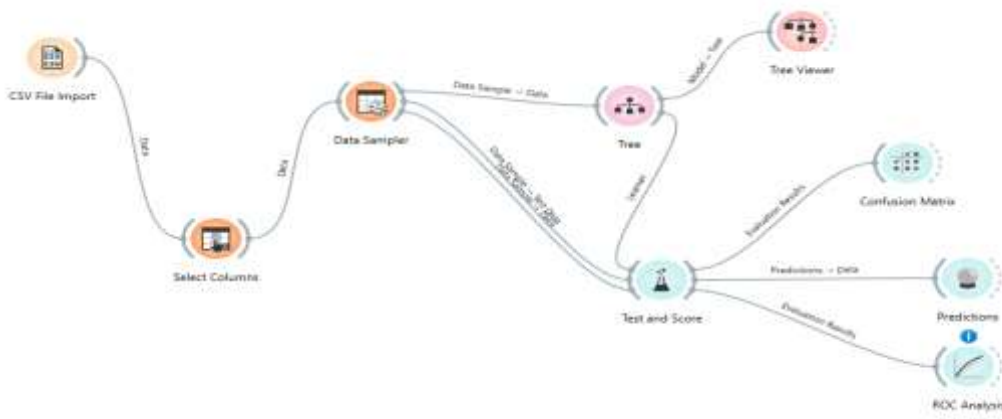
4.4.2. Birinci problemin Orange ile çözümü

Bu bölümde Orange programı kullanılarak müşteri borcunu ödeyip ödemeyeceğini tahmin etmeye yönelik bir sınıflandırma sistemi geliştirilmiş; Karar Ağacı (Decision Tree), Naive Bayes ve K-En Yakın Komşu (K Nearest Neighbor) algoritmaları uygulanmıştır. Her üç algoritma için en iyi performansı sergileyen model belirlenmiş ve sonuçlar karşılaştırmalı olarak yorumlanmıştır. Model eğitimi ve testinde veri setinin

tamamından yararlanabilmek amacıyla Orange programında sabit veri oranı (Fixed Proportion of Data) parametresi %100 olarak ayarlanmıştır.

4.4.2.1. Birinci problem için Orange ile decision tree algoritması çözümü

Bu bölümde, Orange programını kullanılarak Decision Tree Algoritması ile müşteri borcunu ödeyip ödemeyeceğini tahmin eden bir sistemin geliştirilmesine yönelik elde edilen sonuçlar yorumlanmış ve modelin akış şeması Şekil 4.13.'de verilmiştir.



Şekil 4.13. Birinci problem için Orange ile Decision Tree algoritması akış şeması

Birinci problem için; Orange programı ile Decision Tree algoritması ile elde edilen genel performans metrikleri ise Çizelge 4.20’de verilmiştir.

Çizelge 4.20. Birinci problem Orange ile Decision Tree algoritması genel performans metrikleri

Performans Metrikleri	AUC	CA	F1 Score	Precision	Recall	MCC	Spec	LogLoss
Sınıflara Göre Ortalama	0,980	0,986	0,986	0,987	0,986	0,975	0,996	0,358
COLLECTION	0,939	0,986	0,964	1,000	0,930	0,956	1,000	0,186
COLLECTION_PAIDOFF	0,991	0,986	0,966	0,935	1,000	0,958	0,983	0,186
PAIDOFF	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0,306

Çizelge 2.20’deki performans metriklerine bakıldığında, bu model müşteri borcunu ödeyip ödemeyeceğini tahmin etmede oldukça başarılıdır. Tüm sınıflarda yüksek doğruluk, kesinlik ve hatırlama oranlarına sahiptir. Bu sonuç modelin güvenilir ve tutarlı olduğunu göstermektedir. Şekil 4.14’de ise birinci problemi için Orange programı ile Decision Tree algoritması ile elde edilen Confusion Matrix’i verilmiştir.

		Predicted			Σ
		COLLECTION	COLLECTION_PAIDOFF	PAIDOFF	
Actual	COLLECTION	93	7	0	100
	COLLECTION_PAIDOFF	0	100	0	100
	PAIDOFF	0	0	300	300
Σ		93	107	300	500

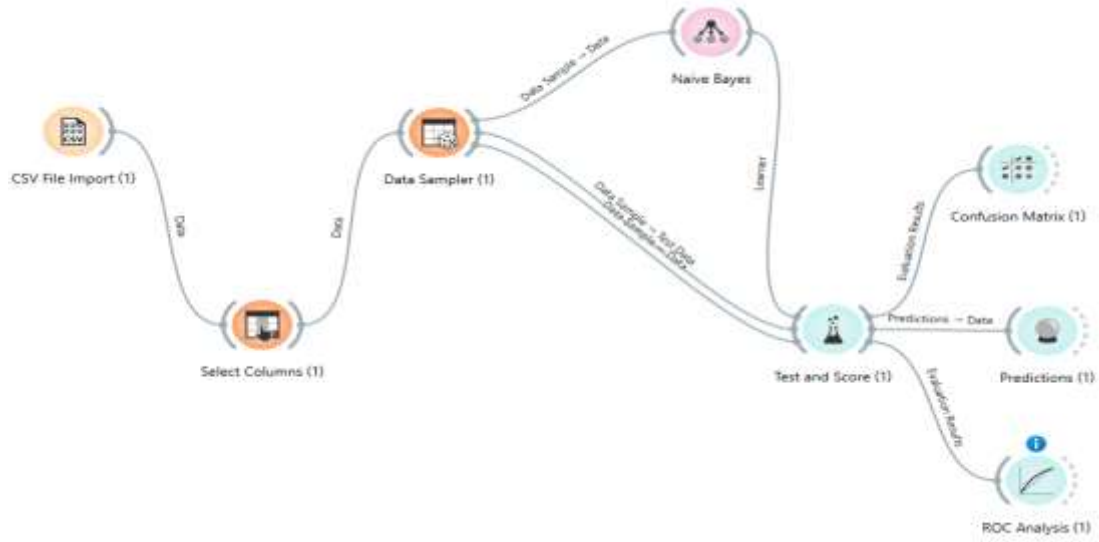
Şekil 4.14. Birinci problem için Orange ile Decision Tree Confusion Matrix

Performans metrikleri karşılaştırıldığında, referans tezde 500 örneklem üzerinden yüksek doğruluk oranı elde edildiği görülmektedir. Bu çalışmada ise 500 örneklem üzerinden hesaplanan sınıflandırma doğruluğu (Accuracy: 0.986) ve ağırlıklı F1-Skor (0.986) değerleri, referans tezdeki performans düzeyiyle benzer nitelikte sonuçlar ortaya koymaktadır.

Hata matrisi (Confusion Matrix) incelendiğinde, PAIDOFF kategorisinin her iki çalışmada da yüksek doğrulukla sınıflandırıldığı görülmektedir. Referans tezde 300 gözlemin tamamı doğru tahmin edilirken, bu çalışmada da PAIDOFF sınıfının 300 gözleminin tamamı mükemmelen sınıflandırılmıştır. Tahsilat kategorileri arasındaki sınıflandırmada ise farklılıklar gözlemlenmiştir. Bu çalışmada COLLECTION kategorisine ait 100 gözlemin 93'ü doğru tahmin edilmiş, 7'si COLLECTION_PAIDOFF olarak yanlış sınıflandırılmıştır. COLLECTION_PAIDOFF kategorisinde ise 100 gözlemin tamamı doğru tahmin edilmiştir. Elde edilen AUC değeri (0.986) ve Precision metriği (0.935), modelin güvenilir tahminler sunduğunu ve yanlış pozitif oranının düşük olduğunu göstermektedir.

4.4.2.2. Birinci problem için Orange ile Naive Bayes algoritması çözümü

Bu bölümde, Orange programını kullanılarak Naive Bayes Algoritması ile müşteri borcunu ödeyip ödemeyeceğini tahmin eden bir sistemin geliştirilmesine yönelik elde edilen sonuçlar yorumlanmış ve modelin akış şeması Şekil 4.15.'de verilmiştir



Şekil 4.15. Birinci problem için Orange ile Naive Bayes algoritması akış şeması

Çizelge 4.21.’de birinci problem için Orange programı ile Naive Bayes algoritması ile elde edilen genel performans metrikleri verilmiştir.

Çizelge 4.21. Birinci problem için Orange ile Naive Bayes algoritması genel performans metrikleri

Performans Metrikleri	AUC	CA	F1	Precision	Recall	MCC	Spec	LogLoss
			Score					
Sınıflara Göre Ortalama	0,980	0,980	0,980	0,980	0,980	0,964	0,995	0,393
COLLECTION	0,958	0,980	0,949	0,969	0,930	0,937	0,993	0,205
COLLECTION_PAIDOFF	0,972	0,980	0,951	0,933	0,970	0,939	0,983	0,208
PAIDOFF	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0,326

Çizelge 4.21 incelendiğinde, modelin tüm sınıflarda yüksek doğruluk, kesinlik ve duyarlılık değerleri sergileyerek müşteri borç ödeme tahmininde güvenilir ve tutarlı bir performans ortaya koyduğu görülmektedir. Birinci probleme ilişkin Orange programında Karar Ağacı algoritmasıyla elde edilen karmaşıklık matrisi Şekil 4.16’da sunulmaktadır.

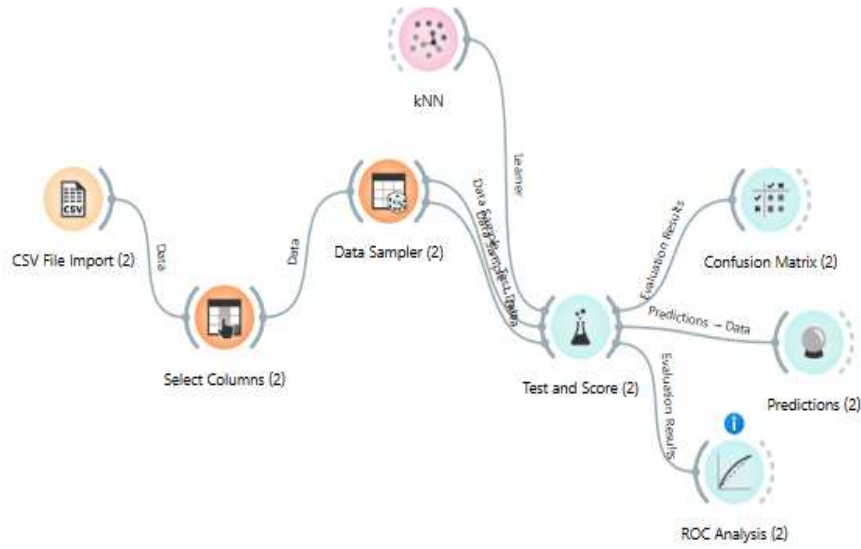
		Predicted			Σ
		COLLECTION	COLLECTION_PAIDOFF	PAIDOFF	
Actual	COLLECTION	93	7	0	100
	COLLECTION_PAIDOFF	3	97	0	100
	PAIDOFF	0	0	300	300
Σ		96	104	300	500

Şekil 4.16. Birinci problem için Orange ile Naive Bayes Confusion Matrix

Performans metrikleri karşılaştırıldığında, Naive Bayes modelinin 500 örneklem üzerinden elde ettiği sınıflandırma doğruluğu (Accuracy: 0.980) ve F1-Skor değerleri, referans tezdeki yüksek performans düzeyiyle büyük ölçüde benzerlik göstermektedir. Hata matrisi (Confusion Matrix) incelendiğinde, PAIDOFF kategorisinin her iki çalışmada da kusursuz sınıflandırıldığı görülmektedir; bu çalışmada da PAIDOFF sınıfına ait 300 gözlemin tamamı (%100 başarıyla) doğru tahmin edilmiştir. Sınıflar arası geçişler incelendiğinde ise hata oranının oldukça düşük olduğu gözlemlenmiştir. Model, COLLECTION kategorisine ait 100 gözlemin 93'ünü doğru tahmin ederken, 7 gözlemi COLLECTION_PAIDOFF olarak yanlış sınıflandırmıştır. Benzer şekilde, COLLECTION_PAIDOFF kategorisindeki 100 gözlemin 97'si doğru sınıflandırılmış, yalnızca 3 gözlem COLLECTION olarak tahmin edilmiştir. Elde edilen yüksek AUC ve hassasiyet (Precision) değerleri, Naive Bayes modelinin müşteri kredi riskini tahmin etmede güçlü, dengeli ve güvenilir bir genel performans sergilediğini ortaya koymaktadır.

4.4.2.3. Birinci problem için Orange ile K nearest neighbor algoritması çözümü

Bu bölümde, Orange programını kullanılarak K Nearest Neighbor Algoritması ile müşteri borcunu ödeyip ödemeyeceğini tahmin eden bir sistemin geliştirilmesine yönelik elde edilen sonuçlar yorumlanmış ve modelin akış şeması Şekil 4.17.'de verilmiştir



Şekil 4.17. Birinci problem için Orange ile K Nearest Neighbor algoritması akış şeması

Çizelge 4.22.'de birinci problem için Orange programı ile K Nearest Neighbor algoritması ile elde edilen genel performans metrikleri verilmiştir.

Çizelge 4.22. Birinci problem için Orange ile K Nearest Neighbor algoritması genel performans metrikleri

Performans Metrikleri	AUC	CA	F1	Precision	Recall	MCC	Spec	LogLoss
	Score							
Sınıflara Göre Ortalama	0,999	0,976	0,976	0,977	0,976	0,957	0,972	0,063
COLLECTION	0,999	0,990	0,975	0,970	0,980	0,969	0,993	0,029
COLLECTION_PAIDOFF	0,997	0,980	0,947	1,000	0,900	0,937	1,000	0,057
PAIDOFF	0,999	0,982	0,985	0,971	1,000	0,963	0,955	0,039

KNN algoritmasıyla elde edilen %97,6 sınıflandırma doğruluğu (CA) ve 0,976 F1 skoru, referans tezdeki %95,8 düzeyini aşarak modelin yüksek bir tahmin gücüne sahip olduğunu ortaya koymaktadır. Özellikle PAIDOFF sınıfında ulaşılan 1,000 Recall değeri, tezdeki 0,977'lik skoru geride bırakmış ve modelin kredi geri ödeme risklerini tespit etmede tutarlı bir performans sergilediğini doğrulamıştır. Birinci probleme ilişkin Orange programında K-En Yakın Komşu algoritmasıyla elde edilen karmaşıklık matrisi Şekil 4.18'te sunulmaktadır.

		Predicted			Σ
		COLLECTION	COLLECTION_PAIDOFF	PAIDOFF	
Actual	COLLECTION	98	0	2	100
	COLLECTION_PAIDOFF	3	90	7	100
	PAIDOFF	0	0	300	300
Σ		101	90	309	500

Şekil 4.18. Birinci problem için Orange ile K Nearest Neighbor Confusion Matrix

Karmaşıklık matrisi incelendiğinde, PAIDOFF sınıfında 300 gözlemin tamamının doğru sınıflandırıldığı ve referans tezdeki yüksek performans düzeyinin yakalandığı görülmektedir. COLLECTION sınıfında 100 örnekten 98'inin doğru tahmin edilmesi ve yalnızca 2 gözlemin PAIDOFF olarak yanlış sınıflandırılması, modelin bu kategoride güvenilir bir performans sergilediğini ortaya koymaktadır. COLLECTION_PAIDOFF sınıfında ise 100 örnekten 90'ının doğru tahmin edilmesi ve 3'ünün COLLECTION, 7'sinin ise PAIDOFF olarak yanlış sınıflandırılması, referans tezde de gözlemlenen hata eğiliminin bu çalışmada da devam ettiğine işaret etmektedir. Genel olarak değerlendirildiğinde, 500 gözlem üzerinden elde edilen sonuçların (Accuracy: 0.976, F1-Score: 0.976) referans tezdeki bulgularla tutarlılık gösterdiği ve modelin sınıflar arasında başarılı bir ayırım yapabildiğini göstermektedir.

4.4.2.4. Orange birinci problemin algoritmalarının karşılaştırılması

Bu bölümde, birinci problem için Orange programında kullanılan algoritmalar karşılaştırılmıştır. Müşteri borcunu ödeyip ödemeyeceğini tahmin eden bir sistem oluşturmak adına, en iyi sonucu veren algoritma bulunmuş ve yorumlanmıştır. Çizelge 4.23.'de birinci problem, Orange programında banka kredi müşteri verileri içi kullanılan tüm algoritmaların performans metrikleri verilmiştir.

Çizelge 4.23. Birinci problem için Orange ile tüm algoritmaların performans metrikleri

Algoritmalar	AUC	CA	F1 Score	Precision	Recall	MCC	Spec	LogLoss
Decision Tree	0,980	0,986	0,986	0,987	0,986	0,975	0,996	0,353
Naive Bayes	0,980	0,980	0,980	0,980	0,980	0,964	0,995	0,393
K Nearest Neighbor	0,999	0,976	0,976	0,977	0,976	0,957	0,972	0,063

Çizelge 4.23 incelendiğinde, elde edilen performans metriklerinin referans tezdeki bulgularla büyük ölçüde örtüştüğü, hatta bazı algoritmalarda referans değerlerin üzerine çıkıldığı görülmektedir. Sınıflandırma başarısı (CA: 0,986) ve dengeli başarı metrikleri (F1-Score: 0,986; Precision: 0,987) dikkate alındığında, Karar Ağacı (Decision Tree) modelinin en kararlı algoritma olarak öne çıktığı görülmektedir. Naive Bayes, 0,980 doğruluk oranıyla bu performansı yakından takip etmektedir.

Öte yandan, K-En Yakın Komşu (KNN) algoritması, genel doğruluk oranında (CA: 0,976) diğer iki modelin hafifçe gerisinde kalmasına rağmen, 0,999'luk kusursuza yakın AUC değeri ve 0,063 gibi oldukça düşük bir LogLoss maliyeti ile dikkat çekmektedir. Bu durum, KNN modelinin sınıfları ayırt etme yeteneğinin mükemmel olduğunu ve tahmin olasılıklarındaki hata payının (belirsizliğin) minimum düzeyde seyrettiğini göstermektedir. Bununla birlikte, tüm algoritmalarda 0,957'nin üzerinde gerçekleşen MCC değerleri ve yüksek özgüllük (Specificity) oranları, modellerin kredi riskini tahminde rassal bir başarıdan uzak, yüksek derecede tutarlı sonuçlar ürettiğini kanıtlamaktadır. Sonuç olarak; mutlak sınıflandırma doğruluğu ve operasyonel kesinlik ön planda tutulduğunda Karar Ağacı; olasılıksal ceza maliyetlerinin (LogLoss) minimizasyonu ve ayırt edicilik gücü hedeflendiğinde ise KNN algoritması kredi risk yönetimi için en avantajlı alternatifler olarak değerlendirilmektedir.

4.4.3. Birinci problemin Python ile çözümü

Bu bölümde Python programı kullanılarak müşteri borcunu ödeyip ödemeyeceğini tahmin etmeye yönelik bir sınıflandırma sistemi geliştirilmiş; Karar Ağacı (Decision Tree), Naive Bayes ve K-En Yakın Komşu (K Nearest Neighbor) algoritmaları uygulanmıştır. Her üç algoritma için en iyi performansı sergileyen model belirlenmiş ve sonuçlar karşılaştırmalı olarak yorumlanmıştır. Model eğitimi ve testinde çapraz doğrulama (Cross-Validation) yöntemi farklı katman (fold) değerleriyle denenmiş ve veri setini en dengeli biçimde bölümlendiren değer belirlenmiştir.

Analiz sürecinin ilk aşamasında ham veri seti Pandas kütüphanesi aracılığıyla çalışma ortamına aktarılmıştır. Yapısal inceleme sonucunda veri setinin 500 satır ve 11 öznitelikten oluştuğu saptanmıştır. Loan_ID, loan_status, Principal, terms, effective_date, due_date, paid_off_time, past_due_days, age, education ve Gender değişkenlerinden oluşan bu yapının hem finansal hem de demografik bilgileri kapsadığı görülmektedir. Veri setinin ilk gözlemlerinde tarihsel değişkenlerin (effective_date, due_date) ve kategorik verilerin (loan_status, education, Gender) varlığı, modelleme öncesinde kapsamlı bir ön işleme sürecine ihtiyaç duyulduğunu ortaya koymaktadır. Şekil 4.19 ise birinci problemi için Python programı ile veri setinin görüntüsü verilmiştir.

Şekil: (500, 11)

Sütunlar: ['Loan_ID', 'loan_status', 'Principal', 'terms', 'effective_date', 'due_date', 'paid_off_time', 'past_due_days', 'age', 'education', 'Gender']

	Loan_ID	loan_status	Principal	terms	effective_date	due_date	paid_off_time	past_due_days	age	education	Gender
0	sqd20160231	PAIDOFF	1000	30	9/8/2016	10/7/2016	9/14/2016 19:31	NaN	45	High School or Below	male
1	sqd20160902	PAIDOFF	1000	30	9/8/2016	10/7/2016	10/7/2016 9:00	NaN	50	Bachelor	female
2	sqd20160003	PAIDOFF	1000	30	9/8/2016	10/7/2016	9/25/2016 16:58	NaN	33	Bachelor	female
3	sqd20160004	PAIDOFF	1000	15	9/8/2016	9/22/2016	9/22/2016 20:00	NaN	27	college	male
4	sqd20160005	PAIDOFF	1000	30	9/9/2016	10/8/2016	9/23/2016 21:36	NaN	28	college	female
5	sqd20160706	PAIDOFF	300	7	9/9/2016	9/15/2016	9/9/2016 13:45	NaN	35	Master or Above	male
6	sqd20160007	PAIDOFF	1000	30	9/9/2016	10/8/2016	10/7/2016 23:07	NaN	29	college	male
7	sqd20160008	PAIDOFF	1000	30	9/9/2016	10/8/2016	10/5/2016 20:33	NaN	36	college	male
8	sqd20160909	PAIDOFF	1000	30	9/9/2016	10/8/2016	10/8/2016 16:00	NaN	28	college	male
9	sqd20160010	PAIDOFF	800	15	9/10/2016	9/24/2016	9/24/2016 13:00	NaN	26	college	male

Şekil 4.19. Birinci problem için Python ile veri seti görüntüsü

Değişkenlerin türleri ve doluluk oranları incelendiğinde, özniteliklerin büyük çoğunluğunun nesne (object) tipinde olduğu, finansal ve demografik verilerin ise tam sayı (int64) veya ondalıklı sayı (float64) olarak tanımlandığı görülmektedir. Eksik değer analizi kapsamında paid_off_time sütununda %20 (100 kayıt) ve past_due_days sütununda %60 (300 kayıt) oranında eksik veri tespit edilmiştir. Bu eksikliklerin rastlantısal olmayıp kredinin ödenme durumuyla doğrudan ilişkili olduğu değerlendirilmektedir; örneğin borcunu zamanında ödemiş bir müşteriye ait gecikme günü (past_due_days) verisinin boş olması yapısal olarak beklenen bir durumdur. Bu saptama, ön işleme aşamasında uygulanacak doldurma stratejilerinin belirlenmesinde belirleyici bir işlev üstlenmiştir.

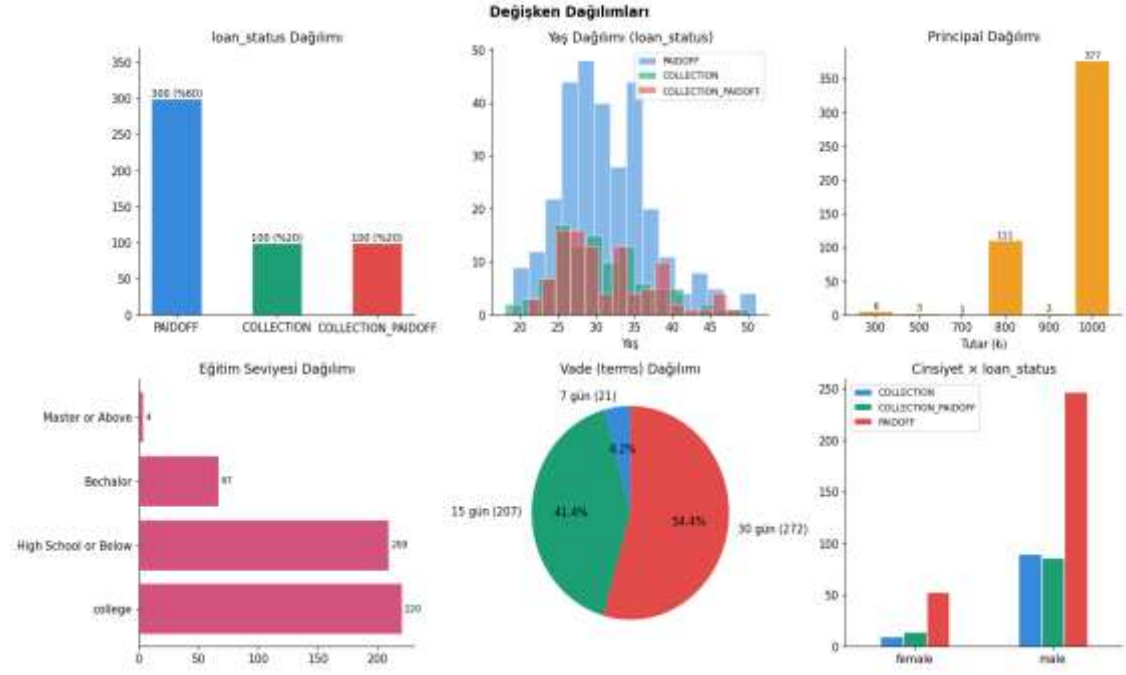
Sayısal özniteliklerin betimsel istatistikleri incelendiğinde, kredi ana parasının (Principal) ortalama 943,20 \$ olduğu ve verilerin büyük bölümünün 1.000 \$ değerinde yoğunlaştığı görülmektedir. Kredi vadeleri 7 ile 30 gün arasında değişmekte olup ortalama vade 22,82 gün olarak hesaplanmıştır. Demografik açıdan bakıldığında müşteri

yaş ortalaması 31,12 olup yaş aralığının 18 ile 51 arasında dağıldığı görülmektedir. Gecikme yaşanan kredilerde ortalama gecikme süresi yaklaşık 36 gün olmakla birlikte bu değerin 76 güne kadar yükseldiği gözlemlenmektedir. Söz konusu istatistikler, veri setindeki genel eğilimin ve uç değerlerin belirlenmesinde temel referans noktası olarak değerlendirilmiştir. Şekil 4.20’de ise birinci problemi için Python programı ile veri tipleri, boş değerler ve sayısal değişkenler hakkında bilgiler verilmiştir.

=== VERİ TİPLERİ ===			=== SAYISAL DEĞİŞKENLER ===				
Loan_ID	object			Principal	terms	past_due_days	age
loan_status	object		count	500.00	500.00	200.00	500.00
Principal	int64		mean	943.20	22.82	36.01	31.12
terms	int64		std	115.24	8.00	29.38	6.08
effective_date	object		min	300.00	7.00	1.00	18.00
due_date	object		25%	1000.00	15.00	3.00	27.00
paid_off_time	object		50%	1000.00	30.00	37.00	30.00
past_due_days	float64		75%	1000.00	30.00	60.00	35.00
age	int64		max	1000.00	30.00	76.00	51.00
education	object						
Gender	object						
dtype: object							
=== BOŞ DEĞERLER ===							
	Boş Sayı	Boş %					
Loan_ID	0	0.0					
loan_status	0	0.0					
Principal	0	0.0					
terms	0	0.0					
effective_date	0	0.0					
due_date	0	0.0					
paid_off_time	100	20.0					
past_due_days	300	60.0					
age	0	0.0					
education	0	0.0					
Gender	0	0.0					

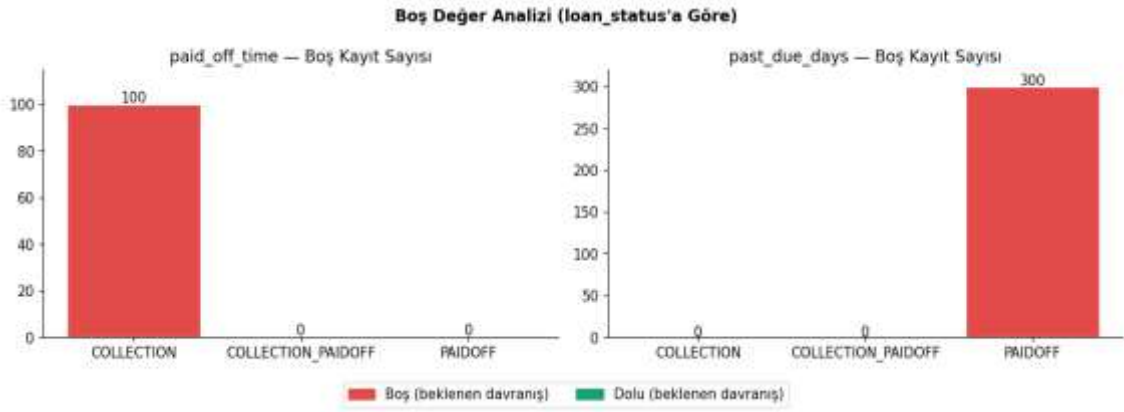
Şekil 4.20. Birinci problem için Python ile sayısal bilgiler

Modelleme öncesinde gerçekleştirilen görselleştirme çalışmaları, değişkenlerin genel karakteristiği ve hedef değişken olan loan_status ile ilişkileri hakkında önemli bulgular ortaya koymuştur. Kredi durum dağılımı incelendiğinde, gözlemlerin %60'ının PAIDOFF sınıfında yoğunlaştığı ve bu durumun sınıf dengesizliğine (class imbalance) yol açtığı görülmektedir. Kredi ana parası ve vade analizlerinde ise 1.000 \$ tutarındaki 30 günlük kredilerin veri setine hâkim olduğu saptanmıştır. Demografik açıdan bakıldığında yaş dağılımının 25-35 aralığında yoğunlaştığı, eğitim düzeyi bakımından ise "college" ve "High School or Below" kategorilerinin büyük çoğunluğu oluşturduğu görülmektedir. Cinsiyet bazlı analizde erkek müşterilerin veri setinde daha geniş bir paya sahip olduğu, ancak her iki cinsiyette de kredi geri ödeme eğilimlerinin benzer bir örüntü sergilediği tespit edilmiştir. Bu betimsel bulgular, modelin öğrenme sürecinde hangi özniteliklerin daha ayırt edici bir rol üstlenebileceğine dair temel bir çerçeve sunmaktadır. Şekil 4.21’de ise birinci problemi için Python programı ile değişken dağılımları gösterilmiştir.



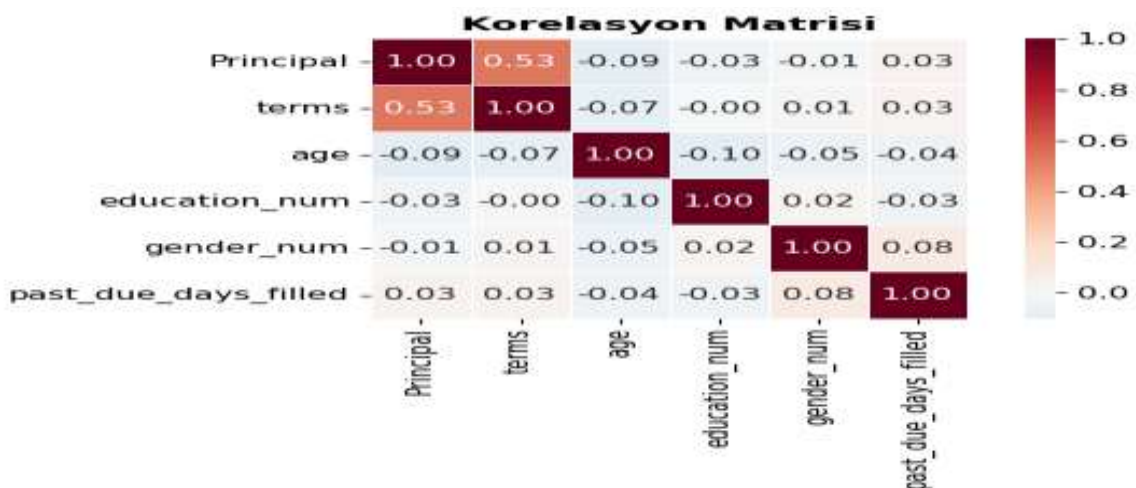
Şekil 4.21. Birinci problem için Python ile değişken dağılımları

Eksik verilerin hedef değişken olan `loan_status` ile ilişkisi incelendiğinde, bu kayıtların rastlantısal olmayıp belirli bir finansal mantık çerçevesinde kümelendiği görülmektedir. `paid_off_time` özniteliğindeki 100 eksik kaydın tamamının `COLLECTION` sınıfında, `past_due_days` sütunundaki 300 eksik kaydın ise tamamının `PAIDOFF` sınıfında yer aldığı saptanmıştır. Bu durum, borcunu zamanında ödemiş müşterilere ait gecikme günü verisinin ve henüz borcunu kapatmamış müşterilere ait ödeme zamanı verisinin boş kalmasının yapısal olarak beklenen bir sonuç olduğunu doğrulamaktadır. Bu nedenle söz konusu eksiklikler bir veri hatası olarak değil, sınıflar arasındaki ayrımı güçlendiren anlamlı bir bilgi kaynağı olarak değerlendirilmiştir. Şekil 4.22’de ise birinci problemi için Python programı ile boş değer analizinin görselleştirilmiş hali verilmiştir.



Şekil 4.22. Birinci problem için Python ile boş değer analizi

Değişkenler arasındaki doğrusal ilişkileri belirlemek amacıyla oluşturulan korelasyon matrisi incelendiğinde, özneliklerin büyük ölçüde birbirinden bağımsız bir yapı sergilediği görülmektedir. Kredi ana parası (Principal) ile kredi vadesi (terms) arasında hesaplanan 0,53 düzeyindeki pozitif korelasyon, veri setindeki en belirgin ilişki olarak öne çıkmaktadır. Diğer değişkenler arasındaki korelasyon katsayılarının sıfıra yakın seyretmesi, çoklu doğrusallık (multicollinearity) sorununun bulunmadığına ve modelin birbirinden bağımsız bilgi sunan öznelikler üzerinde eğitileceğine işaret etmektedir. Bu yapı, modelin ayırt edici gücünü ve genelleme kapasitesini destekleyen sağlıklı bir veri temeli oluşturmaktadır. Şekil 4.23 ise birinci problemi için Python programı ile değişkenlerden elde edilen Confusion Matrix'i verilmiştir.



Şekil 4.23. Birinci problem için Python ile değişkenlerin korelasyon matrisi

Modelleme aşamasına geçilmeden önce kategorik hedef değişkenler sayısal etiketlere dönüştürülmüştür. Veri setindeki 500 kayıt; Sınıf 0 (PAIDOFF) 300 kayıt (%60), Sınıf 1 (COLLECTION) 100 kayıt (%20) ve Sınıf 2 (COLLECTION_PAIDOFF) 100 kayıt (%20) olmak üzere üç sınıfa ayrılmıştır. Bu dağılım, veri setindeki sınıf dengesizliğini bir kez daha açıkça ortaya koymaktadır. Şekil 4.24'te ise birinci problem için Python programı ile hedef değişkenin modele verilmeden önce 0, 1 ve 2 değerleriyle sayısallaştırılması işlemi gerçekleştirilmiştir.

```
Multiclass hedef dağılımı:
target_multi
0      300
2      100
1      100
Name: count, dtype: int64
```

Şekil 4.24. Birinci problem için Python ile değişkenin sayısallaştırılması

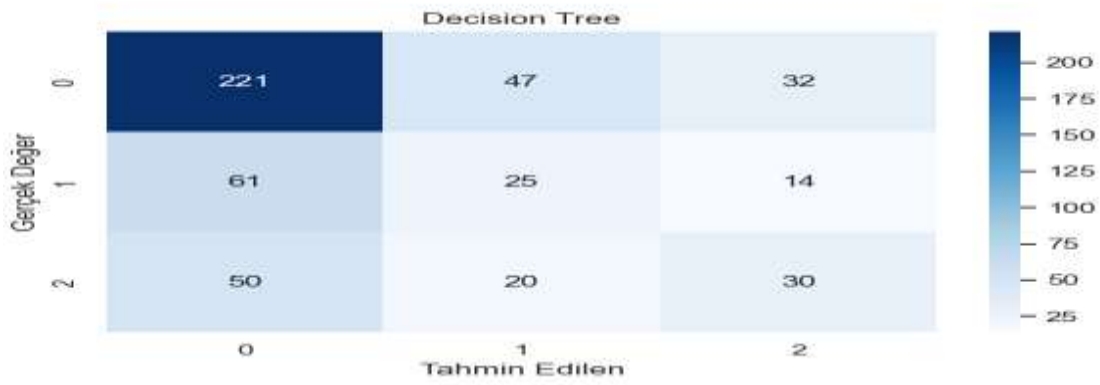
4.4.3.1. Birinci problem için Python ile decision tree algoritması çözümü

Bu bölümde Python programı kullanılarak Karar Ağacı (Decision Tree) algoritması ile müşteri borcunu ödeyip ödemeyeceğini tahmin etmeye yönelik elde edilen sonuçlar yorumlanmış ve performans metrikleri Çizelge 4.24'da sunulmaktadır.

Çizelge 4.24. Birinci problem için Python ile Decision Tree algoritması performans metrikleri

Performans Metrikleri	Performans Değerleri
Doğruluk (Accuracy)	0.5520
Kesinlik (Precision)	0.5327
Duyarlılık (Recall)	0.5520
F1 Skoru (F1 Score)	0.5399

Performans metrikleri incelendiğinde, modelin referans tezdeki değerlere kıyasla tüm temel ölçütlerde iyileşme sergilediği görülmektedir. Doğruluk (Accuracy) 0,430'dan 0,552'ye, Kesinlik (Precision) 0,451'den 0,532'ye, Duyarlılık (Recall) 0,430'dan 0,552'ye ve F1 skoru 0,439'dan 0,539'a yükselmiştir. Bununla birlikte, elde edilen bu değerlerin finansal risk tahmini gibi yüksek hassasiyet gerektiren uygulamalar için henüz yeterli düzeyde olmadığı görülmektedir. Şekil 4.25 ise birinci problemi için Python programı ile Decision Tree algoritması ile elde edilen Confusion Matrix'i verilmiştir



Şekil 4.25. Birinci problem için Python ile Decision Tree algoritması Confusion Matrix

Karmaşıklık matrisi incelendiğinde, modelin 500 örneklem içerisinde 221 PAIDOFF, 25 COLLECTION ve 30 COLLECTION_PAIDOFF sınıfını doğru tahmin ederek toplam 276 doğru sınıflandırmaya ulaştığı görülmektedir. COLLECTION ve COLLECTION_PAIDOFF kategorilerindeki düşük doğru tahmin sayıları, modelin azınlık sınıflarını ayırt etmede yetersiz kaldığını ortaya koymaktadır.

4.4.3.2. Birinci problem için Python ile Naive Bayes algoritması çözümü

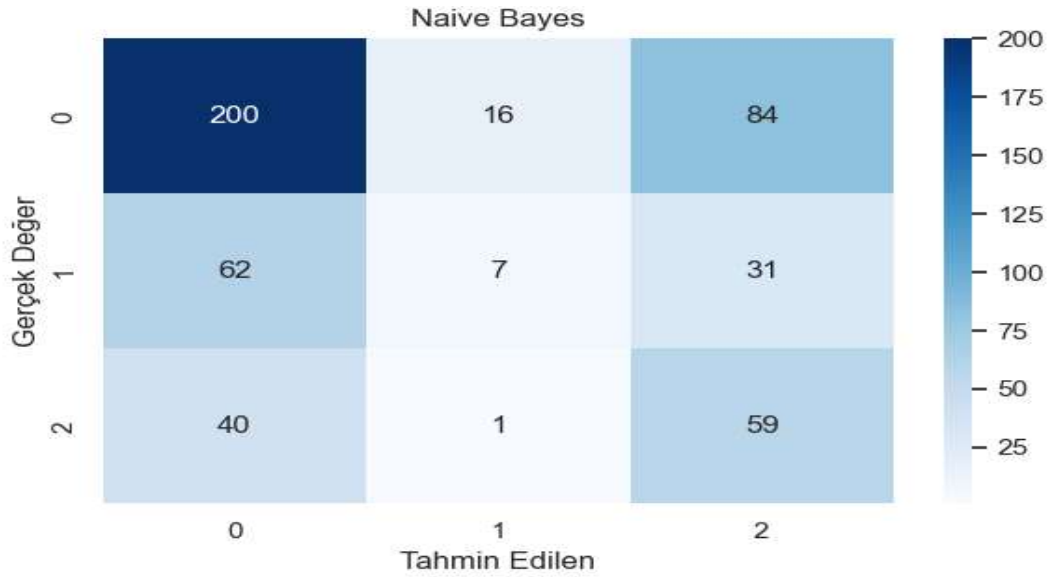
Bu bölümde Python programı kullanılarak Naive Bayes algoritması ile müşteri borcunu ödeyip ödemeyeceğini tahmin etmeye yönelik elde edilen sonuçlar yorumlanmış ve performans metrikleri Çizelge 4.25’de sunulmaktadır.

Çizelge 4.25. Birinci problem için Python ile Naive Bayes algoritması performans metrikleri

Performans Metrikleri	Performans Değerleri
Doğruluk (Accuracy)	53.20%
Sınıflandırma Hatası (Classification Error)	46.80%
Accuracy Count	266
Error Count	234
F1 Skoru	0,5074
Duyarlılık	0,5320
Kesinlik	0,5235

10 katlı çapraz doğrulama sonuçları incelendiğinde, Naive Bayes modelinin %53,20 doğruluk ve %46,80 hata oranıyla sınıfları ayırt etmede yetersiz kaldığı görülmektedir. Referans tezdeki değerlerle karşılaştırıldığında, doğruluk oranının %57,8'den %53,20'ye gerilediği, buna karşın Kesinlik (Precision) değerinin %41,7'den %52,35'e yükseldiği

görülmektedir. Bu durum, genel başarı düşse de modelin pozitif tahminlerinin isabetinde görece bir iyileşme sağlandığına işaret etmektedir. Bununla birlikte, 0,507 düzeyindeki F1 skoru ve 234 yanlış sınıflandırma, modelin sınıflar arasındaki ilişkileri ve azınlık sınıflarını yakalamada yetersiz kaldığını ortaya koymaktadır. Birinci probleme ilişkin Python programında Naive Bayes algoritmasıyla elde edilen karmaşıklık matrisi Şekil 4.26'da sunulmaktadır.



Şekil 4.26. Birinci problem için Python ile Naive Bayes algoritması Confusion Matrix

Karmaşıklık matrisi incelendiğinde, modelin en yüksek başarıyı baskın sınıf olan PAIDOFF kategorisinde sergilediği, 300 gerçek örnekten 200'ünü doğru tahmin ettiği görülmektedir. COLLECTION sınıfında ise 100 örnekten yalnızca 7'si doğru sınıflandırılmış, geri kalanların büyük çoğunluğu PAIDOFF sınıfıyla karıştırılmıştır. COLLECTION_PAIDOFF kategorisinde 100 örnekten 59'unun doğru tahmin edilmesi, modelin bu sınıfta görece daha iyi bir performans sergilediğine işaret etmektedir. Toplamda 500 örnekten 266'sının doğru sınıflandırıldığı göz önüne alındığında, Kesinlik değerindeki iyileşmeye karşın 0,507 düzeyindeki F1 skoru ve sınıflar arasındaki dengesiz dağılım, modelin azınlık sınıflarını yakalamada yetersiz kaldığını ortaya koymaktadır.

4.4.3.3. Birinci problem için Python ile K nearest neighbor algoritması çözümü

Bu bölümde, Python programını kullanılarak K Nearest Neighbor Algoritması ile müşteri borcunu ödeyip ödemeyeceğini tahmin eden bir sistemin geliştirilmesine yönelik elde edilen sonuçlar yorumlanmış ve performans metrikleri Çizelge 4.26.'da verilmiştir.

Çizelge 4.26. Birinci problem için Python ile K Nearest Neighbor algoritması performans metrikleri

Performans Metrikleri	Performans Değerleri
Doğruluk (Accuracy)	56,60%
Sınıflandırma Hatası (Classification Error)	43,40%
Doğru Sınıflandırma Sayısı (Accuracy Count)	283
Yanlış Sınıflandırma Sayısı (Error Count)	217
Kesinlik	0,4899
Duyarlılık	0,5660
F1 Skoru	0,5023

Performans metrikleri incelendiğinde, KNN modelinin %56,60 doğruluk oranıyla 283 gözlemi doğru, 217 gözlemi ise yanlış sınıflandırdığı görülmektedir. 0,502 düzeyindeki F1 skoru, modelin hassasiyet ve duyarlılık dengesi kurmakta zorlandığına işaret etmektedir. Referans tezdeki değerlerle karşılaştırıldığında, doğruluk oranının %66,20'nin ve Kesinlik değerinin %66,40'ın belirgin biçimde gerisinde kaldığı görülmektedir. Bu durum, sınıf dengesizliğinin mesafe tabanlı bir algoritma olan KNN üzerindeki olumsuz etkisini yansıtmaktadır. Birinci probleme ilişkin Python programında K-En Yakın Komşu algoritmasıyla elde edilen karmaşıklık matrisi Şekil 4.27'de sunulmaktadır.



Şekil 4.27. Birinci problem için Python ile KNN algoritması Confusion Matrix

Karmaşıklık matrisi incelendiğinde, modelin PAIDOFF kategorisinde 300 gerçek örnekten 257'sini doğru tahmin ederek yüksek bir başarı sergilediği, ancak

COLLECTION ve COLLECTION_PAIDOFF sınıflarında sırasıyla yalnızca 8 ve 18 doğru tahminle yetersiz kaldığı görülmektedir. Bu iki sınıfa ait örneklerin büyük çoğunluğunun (83 ve 76 kayıt) PAIDOFF sınıfına atanması, modelin çoğunluk sınıfına yönelik belirgin bir yanlılık sergilediğini ortaya koymaktadır. Modelin genel doğruluğunun büyük ölçüde PAIDOFF sınıfındaki yüksek başarıya dayandığı, tahsilat ve gecikme durumlarını tahmin etmede ise yetersiz kaldığı anlaşılmaktadır.

4.4.3.4. Python birinci problemin algoritmalarının karşılaştırılması

Bu bölümde, birinci problem için Python programında kullanılan algoritmalar karşılaştırılmıştır. Müşteri borcunu ödeyip ödemeyeceğini tahmin eden bir sistem oluşturmak adına, en iyi sonucu veren algoritma bulunmuş ve yorumlanmıştır. Çizelge 4.27.'de birinci problem, Python programında banka kredi müşteri verileri için kullanılan tüm algoritmaların performans metrikleri verilmiştir.

Çizelge 4.27. Birinci problem için Python ile tüm algoritmaların performans metrikleri

Performans Metrikleri	Decision Tree	Naive Bayes	K Nearest Neighbours
Doğruluk (Accuracy)	0,5520	0,5320	0,5660
Keskinlik (Precision)	0,5327	0,5235	0,4899
Duyarlılık (Recall)	0,5520	0,5320	0,5660
Doğru Sınıflandırma Sayısı (Accuracy Count)	276	266	283
Yanlış Sınıflandırma Sayısı (Error Count)	224	234	217

Performans metrikleri bir bütün olarak değerlendirildiğinde, K-En Yakın Komşu algoritmasının %56,60 doğruluk oranı ve 283 doğru sınıflandırmayla diğer algoritmalara kıyasla en yüksek genel başarıyı sergilediği görülmektedir. Bu sonuçlar doğrultusunda, müşteri borç ödeme davranışını tahmin etme görevi için çapraz doğrulama bulgularına göre en uygun algoritmanın K-En Yakın Komşu olduğu değerlendirilmektedir.

4.4.3.5. Referans çalışma ile doğruluk oranları karşılaştırma analizi

Bu bölümde, müşterilerin kredi geri ödeme durumlarını tahmin etmek amacıyla Python ortamında geliştirilen modellerin doğruluk oranları, referans alınan çalışmadaki bulgularla karşılaştırmalı olarak analiz edilmiştir. Gerçekleştirilen analizler sonucunda en yüksek performansı gösteren algoritma tespit edilmiş ve literatür bağlamında

yorumlanmıştır. Birinci probleme ait banka kredi müşteri verileri üzerinde test edilen tüm algoritmaların ve referans çalışmanın karşılaştırmalı performans metrikleri Çizelge 4.28’de sunulmuştur.

Çizelge 4.28. Birinci problem için Python ile referans çalışma karşılaştırma performans metrikleri

Algoritma	Referans Çalışma	Mevcut Çalışma	Fark / Durum
Decision Tree	0,430	0,552	+0,122
Naive Bayes	0,578	0,532	-0,046
K-Nearest Neighbors	0,662	0,566	-0,096

Referans çalışmada algoritmalar arasında belirgin bir performans farkı gözlemlenmiş; Karar Ağacı algoritması %43,0 doğruluk oranında kalırken KNN algoritması %66,20 ile en başarılı model olarak belirlenmiştir. Bu çalışmada ise ham veri aşamasında tüm algoritmaların 0,53-0,56 bandında daha dengeli bir performans sergilediği görülmektedir. Referans çalışmadaki KNN başarısı ham veri üzerinde daha yüksek görünse de, iki çalışma arasındaki asıl fark optimizasyon aşamasında ortaya çıkmaktadır. Uygulanan SMOTE veri artırımı ve 500 iterasyonluk hiperparametre optimizasyonu sayesinde referans çalışmanın ulaştığı %66,20’lik en yüksek doğruluk oranı aşılmış ve KNN algoritmasıyla %71,12 doğruluk oranına ulaşılmıştır. Bu sonuç, önerilen metodolojinin referans çalışmaya kıyasla yaklaşık %5’lik bir performans artışı sağladığını ortaya koymaktadır. Kararlılık ve nihai başarı ölçütleri birlikte değerlendirildiğinde, KNN algoritmasının bu problem için en uygun model olduğu sonucuna varılmıştır.

4.4.3.6. Veri setinin dengelenmesi: SMOTE uygulaması

Önceki aşamalarda elde edilen karmaşıklık matrisleri, modellerin 300 kayıtlık PAIDOFF sınıfına yönelik belirgin bir yanlılık sergilediğini ve 100’er kayıtlık azınlık sınıflarını ayırt etmede yetersiz kaldığını ortaya koymuştur. Bu sınıf dengesizliği sorununu gidermek amacıyla SMOTE (Synthetic Minority Over-sampling Technique) yöntemi uygulanmıştır. SMOTE, azınlık sınıflarındaki mevcut verileri doğrudan kopyalamak yerine bu verilerin öznitelik uzayındaki komşularını analiz ederek yeni sentetik örnekler üretmektedir. Modellerin veri hacmine verdiği tepkiyi ölçmek amacıyla veri seti tek seferde genişletilmemiş; 900, 1800 ve nihai olarak 2700 satır olmak üzere kademeli biçimde artırılmıştır. Nihai 2700 kayıtlık yapıda her üç sınıfın örneklem sayısı 900’e eşitlenerek dengeli bir dağılım sağlanmıştır.

4.4.3.7. İlk dengeleme aşaması

SMOTE uygulamasıyla her bir sınıf 300 örneğe tamamlanarak toplamda 900 kayıtlık dengeli bir veri seti oluşturulmuştur. Bu aşamada 10 katlı çapraz doğrulama testleri gerçekleştirilmiş ve her algoritma için en iyi K parametresi belirlenmiştir. Çizelge 4.29'de, dengeli veri seti üzerinde eğitilen algoritmaların çapraz doğrulama testleri sonucunda elde ettikleri performans metrikleri (doğruluk, duyarlılık, kesinlik ve F1-skoru) sunulmuştur.

Çizelge 4.29. Birinci problem için Python ile çapraz doğrulama performans metrikleri

Algoritma	En iyi K Değeri	Doğruluk (Accuracy)	Kappa Skoru
Decision Tree	72	55,11%	0,3267
Naive Bayes	287	50,78%	0,2617
K-Nearest Neighbors	13	59,11%	0,3867

Sonuçlar incelendiğinde, ham veride %56,60 olan KNN doğruluk oranının veri dengelenmesinin ardından %59,11'e yükseldiği ve 0,3867 Kappa değeriyle modelin sınıflar arasında anlamlı bir ayrım yapmaya başladığı görülmektedir. Karar Ağacı algoritması %55 bandında istikrarlı bir performans sergilerken, Naive Bayes modelinin %50 bandına gerilemesi olasılıksal yaklaşımın sentetik veri dağılımına karşı daha hassas olduğuna işaret etmektedir. Bu aşama, veri artırımının model başarısı üzerindeki olumlu etkisini ortaya koyması ve %60 doğruluk düzeyine ulaşılabileceğini göstermesi bakımından önem taşımaktadır.

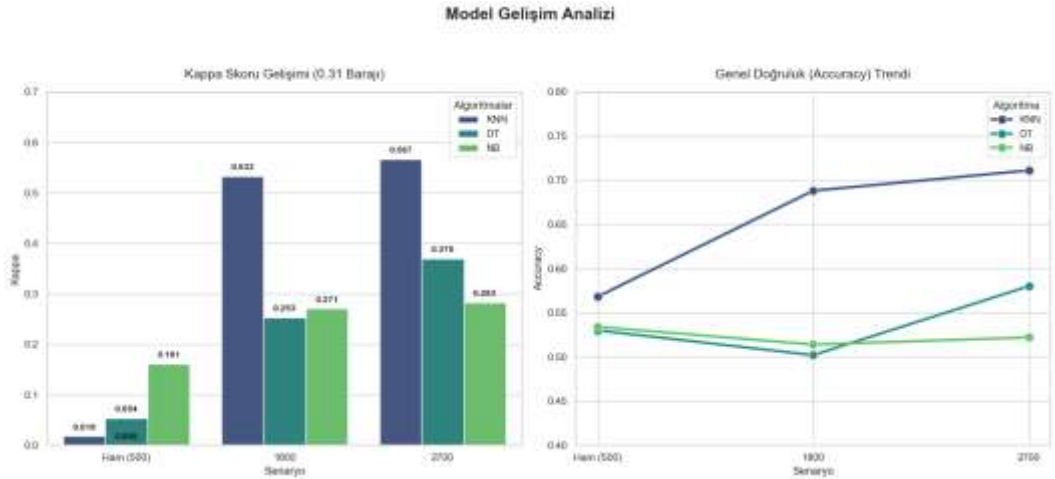
4.4.3.8. Veri hacminin genişletilmesi

İkinci aşamada SMOTE uygulamasıyla her sınıfın örneklem sayısı 600'e, toplam veri boyutu ise 1800 kayda çıkarılmıştır. Bu genişletme, modellerin sentetik verilerle oluşturulan yeni örüntüleri ne ölçüde öğrenebildiğini test etmek amacıyla gerçekleştirilmiş olup elde edilen performans metrikleri Çizelge 4.30'da sunulmaktadır.

Çizelge 4.30. İkinci aşama SMOTE uygulaması ile genişletilmiş veri setine ait karşılaştırmalı performans metrikleri

Algoritma	En İyi K Değeri	Doğruluk (Accuracy)	Cohen's Kappa
KNN	59	69,00%	0,5325
Naive Bayes	14	51,00%	0,2708
Decision Tree	85	50,00%	0,2533

1800 kayıtlık veri setinde KNN algoritması %69,00 doğruluk ve 0,5325 Kappa değeriyle en yüksek performansı sergilemiştir. Kappa değerinin bu düzeye ulaşması, modelin tahminlerinin rastlantısal olmadığını ve sınıflar arasında orta-iyi düzeyde bir uyum sağlandığını göstermektedir. F1 skorlarının tüm sınıflarda %66-72 bandında seyretmesi, SMOTE uygulamasının sınıf dengesini başarıyla sağladığına ve modelin dengeli bir öğrenme gerçekleştirdiğine işaret etmektedir. Karar Ağacı ve Naive Bayes modellerinin %50-51 bandında kalarak KNN'in belirgin biçimde gerisinde kalması, mesafe tabanlı yaklaşımın bu veri setindeki karmaşık ilişkileri modellemede diğer algoritmalara kıyasla daha üstün olduğunu ortaya koymaktadır. Bu aşama, veri hacmindeki artışın KNN modelinde doğrusal bir performans iyileşmesi getirdiğini ve %70 doğruluk eşiğine yaklaşıldığını göstermesi bakımından kritik bir öneme sahiptir. Şekil 4.28'de, modellerin SMOTE sonrası gelişim grafikleri sunulmuştur.



Şekil 4.28. Birinci problem için Python ile SMOTE sonrası model gelişim analizi

Performans gelişim grafikleri, ham veri setinden 2700 satırlık nihai yapıya uzanan süreçte modellerin Kappa skoru ve genel doğruluk açısından nasıl bir değişim sergilediğini ortaya koymaktadır.

Kappa skoru grafiği incelendiğinde, 500 satırlık ham veride tüm modellerin 0,20 barajının altında kalarak rastlantısal tahmin düzeyini aşamadığı görülmektedir. Veri setinin 1800 ve 2700 satıra genişletilmesiyle birlikte KNN algoritmasının Kappa skorunu 0,567 düzeyine taşıdığı ve diğer modelleri belirgin biçimde geride bıraktığı görülmektedir. Karar Ağacı ve Naive Bayes modelleri de gelişim göstermiş, ancak KNN'in sergilediği istikrarlı yükseliş ivmesine ulaşamamıştır.

Doğruluk trendi grafiği değerlendirildiğinde, KNN algoritmasının ham veriden itibaren doğrusal bir artış sergileyerek %71,11 doğruluk oranına ulaştığı görülmektedir. Karar Ağacı ve Naive Bayes modellerinin 1800 satırlık aşamada yaşadığı performans düşüşleri, bu algoritmaların sentetik veri dağılımına karşı daha kırılgan bir yapıda olduğuna işaret etmektedir. 2700 satırlık nihai yapıda KNN'in belirgin üstünlüğü, uygulanan SMOTE ve hiperparametre optimizasyonu stratejisinin etkinliğini doğrulamaktadır. Bu süreç, veri artırımının yalnızca örneklem sayısını artırmakla kalmayıp özellikle KNN algoritmasının sınıflandırma kapasitesini anlamlı ölçüde geliştirdiğini ortaya koymaktadır.

4.4.4. Birinci problemin R ile çözümü

Bu bölümde, R programını kullanılarak müşteri borcunu ödeyip ödemeyeceğini tahmin eden bir sistem geliştirilmeye çalışılmış ve Decision Tree, Naive Bayes, K Nearest Neighbor ve XGBoost algoritmaları kullanılmıştır. Uygulanan bu dört algoritma için en iyi sonuç veren algoritma bulunmuş, sonuçlar karşılaştırılarak yorumlanmıştır.

4.4.4.1. Birinci problem için R ile decision tree algoritması çözümü

Bu bölümde, R programını kullanılarak Decision Tree Algoritması ile müşteri borcunu ödeyip ödemeyeceğini tahmin eden bir sistemin geliştirilmesine yönelik elde edilen sonuçlar yorumlanmış ve performans metrikleri Çizelge 4.31.'de verilmiştir.

Çizelge 4.31. Birinci problem için R ile Decision Tree algoritması performans metrikleri

Performans Metrikleri	Performans Değerleri
Doğruluk (Accuracy)	0.60
Sınıflandırma Hatası	0.40
Kesinlik (Precision)	0.36
Duyarlılık (Recall)	0.60
F1 Skoru (F1 Score))	0.45
Cohen's Kappa	0,00

R programında ham veri üzerinde 10 katlı çapraz doğrulama ile uygulanan Karar Ağacı algoritması, %60 doğruluk ve %40 sınıflandırma hatası ile düşük bir performans sergilemiştir. Cohen's Kappa değerinin 0 olması, modelin rastlantısal tahmin düzeyinin ötesine geçemediğini ortaya koymaktadır. Duyarlılık değerinin 0,60 ile oldukça düşük kalması, modelin azınlık sınıflarını ayırt edemediğini ve tüm tahminleri çoğunluk sınıfı olan PAIDOFF'a yönlendirdiğini göstermektedir. Bu sonuçlar, ham veri üzerinde sınıf dengesizliğinin Karar Ağacı algoritması üzerindeki olumsuz etkisini açıkça ortaya koymaktadır.

4.4.4.2. Birinci problem için R ile Naive Bayes algoritması çözümü

Bu bölümde, R programını kullanılarak Naive Bayes Algoritması ile müşteri borcunu ödeyip ödemeyeceğini tahmin eden bir sistemin geliştirilmesine yönelik elde edilen sonuçlar yorumlanmış ve performans metrikleri Çizelge 4.32.'de verilmiştir.

Çizelge 4.32. Birinci problem için R ile Naive Bayes algoritması performans metrikleri

Performans Metrikleri	Performans Değerleri
Doğruluk (Accuracy)	0.208
Sınıflandırma Hatası	0.792
Doğru Sınıflandırma Sayısı	104
Yanlış Sınıflandırma Sayısı	396
Kesinlik (Precision)	0.5768
Duyarlılık (Recall)	0.2080
F1 Skoru (F1 Score))	0.1011
Cohen's Kappa	0,0050

Python programında veri seti üzerinde Naive Bayes algoritması uygulandığında %20,80 doğruluk (Accuracy) ve %79,20 sınıflandırma hatası elde edilmiştir. Toplam 500 örneklemden yalnızca 104'ü doğru sınıflandırılırken, 396 gözlem yanlış sınıflandırılmıştır. Modelin Cohen's Kappa değerinin 0,0050 ile sıfıra son derece yakın olması, elde edilen bu tahminlerin neredeyse tamamen rastlantısal (şans eseri) düzeyde kaldığına işaret etmektedir. Kesinlik (Precision) 0,5768, Duyarlılık (Recall) 0,2080 ve F1 Skoru 0,1011 değerleriyle birlikte değerlendirildiğinde, algoritmanın bu veri yapısı üzerinde sınıflandırma performansının oldukça düşük kaldığı ve kredi riskini etkin bir şekilde tahmin etmede yetersiz olduğu görülmektedir.

4.4.4.3. Birinci problem için R ile KNN algoritması çözümü

Bu bölümde, R programını kullanılarak KNN Algoritması ile müşteri borcunu ödeyip ödemeyeceğini tahmin eden bir sistemin geliştirilmesine yönelik elde edilen sonuçlar yorumlanmış ve performans metrikleri Çizelge 4.33.'de verilmiştir.

Çizelge 4.33. Birinci problem için R ile KNN algoritması performans metrikleri

Performans Metrikleri	Performans Değerleri
Doğruluk (Accuracy)	0.516
Sınıflandırma Hatası	0.484
Doğru Sınıflandırma Sayısı	258
Yanlış Sınıflandırma Sayısı	242
Kesinlik (Precision)	0.4313
Duyarlılık (Recall)	0.5160
F1 Skoru (F1 Score))	0.4597
Cohen's Kappa	-0.0067

Python programında veri seti üzerinde KNN algoritması uygulandığında %51,60 doğruluk (Accuracy) ve %48,40 sınıflandırma hatası elde edilmiştir. Toplam 500 örneklemden 258'i doğru sınıflandırılırken, 242 gözlemde yanlış sınıflandırma yapılmıştır. Modelin Cohen's Kappa değerinin -0,0067 olarak gerçekleşmesi, elde edilen tahminlerin rastlantısal (şans eseri) başarı düzeyinin bile altında kaldığına ve modelin sınıfları ayırt etmede tamamen yetersiz olduğuna işaret etmektedir. Kesinlik (Precision) 0,4313, Duyarlılık (Recall) 0,5160 ve F1 Skoru 0,4597 değerleriyle birlikte değerlendirildiğinde, algoritmanın bu veri yapısı üzerinde dengeli ve güvenilir bir sınıflandırma performansı sergileyemediği ve kredi risk tahmini için zayıf bir model ortaya koyduğu görülmektedir.

4.4.4.4. Birinci problem için R ile XGBoost algoritması çözümü

Bu bölümde R programı kullanılarak XGBoost algoritması ile müşteri borcunu ödeyip ödemeyeceğini tahmin etmeye yönelik elde edilen sonuçlar yorumlanmış ve performans metrikleri Çizelge 4.34'da sunulmaktadır.

Çizelge 4.34. Birinci problem için R ile XGBoost algoritması performans metrikleri

Performans Metrikleri	Performans Değerleri
Doğruluk (Accuracy)	0.484
Sınıflandırma Hatası	0.516
Doğru Sınıflandırma Sayısı	242
Yanlış Sınıflandırma Sayısı	258
Kesinlik (Precision)	0.4303
Duyarlılık (Recall)	0.484
F1 Skoru (F1 Score))	0.4519
Cohen's Kappa	-0.0094

Python programında veri seti üzerinde XGBoost algoritması uygulandığında %48,40 doğruluk (Accuracy) ve %51,60 sınıflandırma hatası elde edilmiştir. Toplam 500 örneklemden 242'si doğru sınıflandırılırken, 258 gözlemde yanlış sınıflandırma yapılmıştır. Modelin Cohen's Kappa değerinin -0,0094 ile negatif çıkması, elde edilen tahminlerin rastlantısal (şans eseri) başarı düzeyinin bile gerisinde kaldığına işaret etmektedir. Kesinlik (Precision) 0,4303, Duyarlılık (Recall) 0,4840 ve F1 Skoru 0,4519 değerleriyle birlikte değerlendirildiğinde, XGBoost algoritmasının ham veri üzerinde sınıflandırma performansının oldukça zayıf kaldığı görülmektedir. Bu sonuçlar, algoritmanın mevcut haliyle sınıf dengesizliğine karşı yüksek bir hassasiyet gösterdiğini ve veri setinin SMOTE gibi yöntemlerle dengelenmesi adımının XGBoost modeli için kritik bir gereksinim olduğunu ortaya koymaktadır.

4.4.5. Birinci problem için en iyi program seçimi

Müşteri borcunu ödeyip ödemeyeceğini tahmin eden bir sistem oluşturmak adına, en iyi sonucu veren algoritma ve program bulunmuş ve yorumlanmıştır. Sonuçların tüm programlarda farklı çıkmasının nedeni her programda uygulanan yöntem ve parametre farklılıklarıdır. Knime, Python ve R'da yöntem olarak cross validation (çapraz doğrulama) yöntemi ile fold değeri en yüksek performans değeri denenerek bulunmuştur ve bu yöntem ile veri setinin hepsi eğitim için kullanılmıştır. Orange'de ise tüm verilerin eğitimde kullanılması için fixed proportion of data yöntemi kullanılmıştır ve değeri %100 olarak verilmiştir. Veri setinin tümü eğitim için kullanıldığında, daha güvenilir performans tahmini elde edildiği için programa uygun olan bu yöntemler uygulanmıştır. Çizelge 4.35.'de birinci problemde Knime, Orange, Python ve R

programlarında banka kredi müşteri verileri için kullanılan programların en iyi algoritmanın performans metrikleri verilmiştir.

Çizelge 4.35. Birinci problem için tüm programların performans metrikleri

Programlar	Knime	Orange	Python	R
Performans Metrikleri	Naive Bayes	Decision Tree	K Nearest Neighbours	Decision Tree
Doğruluk (Accuracy)	69,4%	98,6%	56,6%	60,0%
Sınıflandırma Hatası (Classification Error)	30,6%	1,4%	43,4%	40,0%
Accuracy Count	347	488	283	300
Error Count	153	12	217	200

Orange (Decision Tree): Orange platformunda uygulanan Karar Ağacı algoritması, %98,60 doğruluk oranı ve sadece %1,40 hata payı ile diğer programlara kıyasla açık ara en üstün performansı sergilemiştir. Toplam 500 örneklemden 488'ini doğru tahmin eden bu algoritma, müşteri borç ödeme tahmini için en etkili model olarak öne çıkmaktadır.

Knime (Naive Bayes): Knime bünyesinde yürütülen Naive Bayes algoritması %69,40 doğruluk oranı ve 347 doğru sınıflandırma sayısı ile başarılı bir grafik çizmiştir. Orange yazılımının ardından veri setinde en yüksek performansı gösteren ikinci algoritma konumundadır.

R (Decision Tree): R programında ham veri üzerinde test edilen Karar Ağacı algoritması %60,00 doğruluk oranıyla (300 doğru tahmin) orta düzeyde bir performans sergilemiştir. Model, sınıf dengesizliğinin olumsuz etkilerine rağmen Knime ve Orange yazılımlarının ardından üçüncü sırada yer almıştır.

Python (K Nearest Neighbors): Python ortamında kurulan KNN algoritması, %56,60 doğruluk ve %43,40 sınıflandırma hatası ile tüm programlar ve algoritmalar arasında en düşük performansı kaydeden yapı olmuştur. Toplamda 217 hatalı sınıflandırma (Error Count) yapan bu model, mevcut veri yapısında borç ödeme tahmini için en az etkili algoritma olarak değerlendirilmektedir.

Bu sonuçlara göre; banka veri seti üzerinde müşteri borcunu ödeyip ödemeyeceğini tahmin etmek için en uygun yazılım ve algoritma kombinasyonunun Orange'ın Decision Tree algoritması olduğu açıkça görülmektedir. Knime platformundaki Naive Bayes algoritması ise kabul edilebilir hata oranıyla bu model için güçlü bir alternatif teşkil etmektedir.

4.5. İkinci ve Üçüncü Problem Modelleme

İkinci Problem: Müşterilerin ne kadar zamanda ödeme yapacağını (erken ise ne kadar geç ise ne kadar geç) tahmin eden bir sistem nasıl geliştirilecek?

Üçüncü Problem: Müşteriye verilebilecek krediyi tahmin eden bir sistem geliştiriniz?

Bu iki problemin modeli oluştururken son aşamadaki formüle kadar kurgular aynıdır. Bu yüzden ikinci ve üçüncü problemler algoritma sonuçları birlikte değerlendirilecektir. Sonuçları gözlemek adına iki problem için hesaplanan tahmin sonuçları bu bölümde verilmiştir. Diğer bölümlerde programlarda oluşturulan modellerin algoritma sonuçları yorumlanmış ve karşılaştırılmıştır ve en iyi algoritma bulunmuştur. Ayrıca ikinci ve üçüncü problemlerin modülü akış şeması Çizelge 4.36’da verilmiştir.

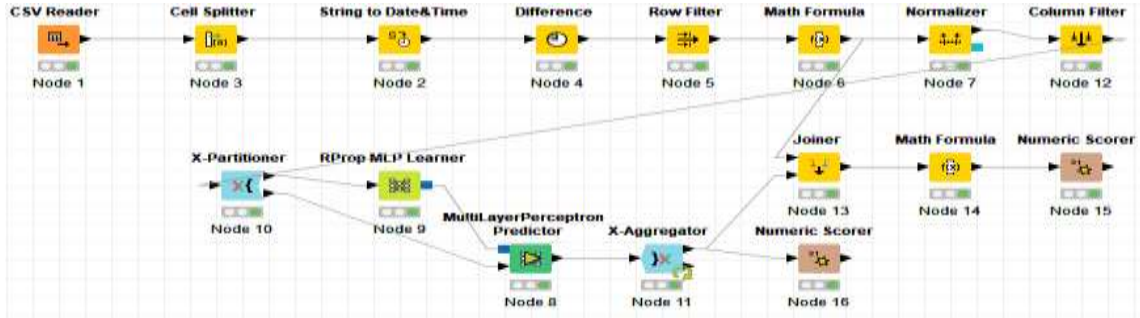
Çizelge 4.36. İkinci ve üçüncü problemlerin modüler çözüm adımları

Modül Adımları	Modüller	Tablo Alanları	Yeni alanlar	Problem
1	Yükleme			2. ve 3. Problem için Ortak
2	Saat ve tarihi ayırma	paid_off_time	paid_off_time_yeni	2. ve 3. Problem için Ortak
3	String ifadeyi tarihe çevirme	effective_time due_date paid_off_time_yeni		2. ve 3. Problem için Ortak
4	İki tarih arası kaç gün var hesaplama	paid_off_time - effective_time	datetime diff	2. ve 3. Problem için Ortak
5	Missing value olan satırları silme	datetime diff		2. ve 3. Problem için Ortak
6	Hesaplama	datetime diff / terms	norm_odeme	2. ve 3. Problem için Ortak
7	Normalize	Principal terms past_due_days age datetime diff norm_odeme	Principal (#1) terms (#1) past_due_days (#1) age (#1) datetime diff (#1) norm_odeme (#1)	2. ve 3. Problem için Ortak
8	Colson Seçme	Principal age datetime diff norm_odeme		2. ve 3. Problem için Ortak
9	Algoritma	Multilayer Perceptron	Norm_odeme	2. ve 3. Problem için Ortak
10	Join	Normalizden önceki verileri		2. ve 3. Problem için Ortak
11	Hesaplama	$\$Prediction(norm_odeme)\$ * COL_MAX(\$norm_odeme\$) * \$datetime_diff\$$ $COL_MAX(\$Principal \$) * \$Prediction(Principal)\$$	gün_odeme_tarih verilecek_kredi	2. Problem için Formülasyon 3. Problem için Formülasyon

9. adım olan algoritma çalıştırma aşaması Knime ve Orange programlarında çalıştırılarak yorumlanması yapılmıştır.

10. adımdaki join aşaması ise 7. adımda bazı alanlar normalleştirildiği için normalleştirme yapılmadan önceki verilerle normalizasyon uygulanmış veriler inner join yapılarak uygulanacak olan formüllere hazırlanmıştır.

11. adımda normalisyon uygulanmış veriler ile normalize uygulama aşamasından önceki verilerdeki bazı alanlar üzerinde hesaplama yapılarak tahminler bulunmuştur. Şekil 4.29'de Knime programı ile 2. ve 3. problem için formülasyonların olduğu bir ağaç modülü akış diyagramı verilmiştir.



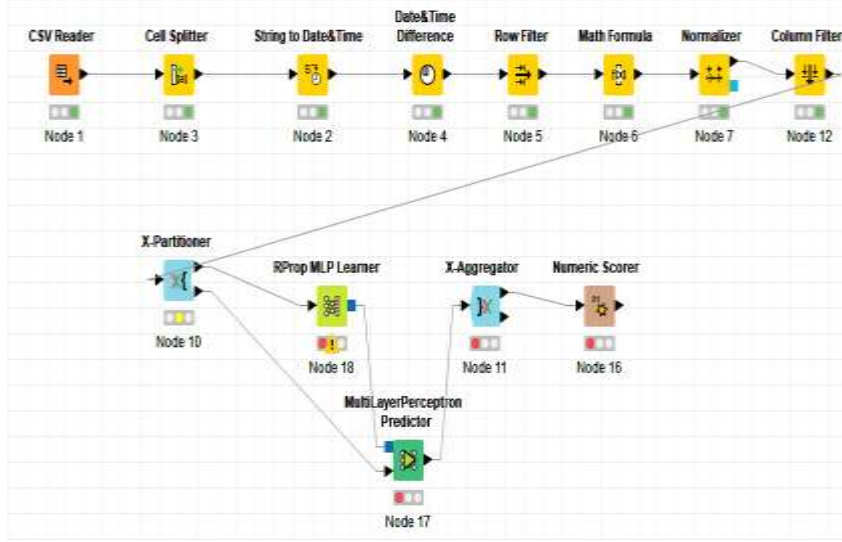
Şekil 4.29. İkinci ve üçüncü problemlerin Knime akış şeması

4.5.1. İkinci ve üçüncü problem için Knime çözümü

Bu bölümde, Knime programını kullanılarak müşterilerin ne kadar zamanda ödeme yapacağını ve müşteriye verilebilecek krediyi tahmin eden bir sistem geliştirilmeye çalışılmış ve sadece MultiLayer Perceptron Predictor sınıflandırma algoritması kullanılmıştır.

4.5.1.1. İkinci ve üçüncü problem için Knime ile multi-layer perceptron predictor algoritması çözümü

Bu bölümde, Knime programını kullanılarak MultiLayer Perceptron Predictor algoritması ile müşterilerin ne kadar zamanda ödeme yapacağını ve müşteriye verilebilecek krediyi tahmin eden bir sistemin geliştirilmesine yönelik elde edilen sonuçlar yorumlanmış ve modelin akış şeması Şekil 4.30.'da verilmiştir. En iyi sonuç veren test ve eğitim oranı doğrulama sayısı cross validation = 10 olarak denenerek bulunmuştur.



Şekil 4.30. İkinci ve üçüncü problemlerin Knime MultiLayer Perceptron Predictor algoritması akış şeması

Çizelge 4.37.'de İkinci ve üçüncü problemler için performans metrik değerleri verilmiştir.

Çizelge 4.37. İkinci ve üçüncü problemlerin Knime MultiLayer Perceptron Predictor performans metrikleri

#	RowID	Prediction (new column) = Number (Float)
1	R ²	0.924
2	mean absolute error	0.028
3	mean squared error	0.003
4	root mean squared error	0.051
5	mean signed difference	-0
6	mean absolute percentage error	NaN
7	adjusted R ²	0.924

Çalışma kapsamında oluşturulan MultiLayer Perceptron (MLP) modelinin performans metrikleri incelendiğinde, R² değerinin 0,923 olduğu görülmektedir. Bu oran, modelin bağımlı değişkendeki varyansın büyük bir kısmını yakalayarak ödeme zamanlarını yüksek bir doğrulukla tahmin ettiğini göstermektedir. Diğer hata metrikleri incelendiğinde; Ortalama Mutlak Hata (MAE) 0,0277, Ortalama Kare Hata (MSE) 0,0025 ve Kök Ortalama Kare Hata (RMSE) 0,0506 olarak hesaplanmıştır. Ayrıca modelin sistematik sapmasını gösteren Ortalama İşaretli Fark (Mean Signed Difference) değeri -0,00025 gibi sıfıra son derece yakın, ancak doğal bir sapma yönünde gerçekleşmiştir.

Referans alınan çalışmada ise aynı problem için R^2 değeri 0,979, MAE 0,007 olarak raporlanırken, MSE ve Ortalama İşaretli Fark değerleri tamamen sıfır olarak belirtilmiştir. Finansal veri setleriyle kurulan yapay sinir ağı modellerinde hata payının ve sistematik sapmanın mutlak sıfıra inmesi, istatistiksel açıdan aşırı öğrenme (overfitting) problemine veya veri ön işleme aşamasında hedef sızıntısına (target leakage) işaret etmektedir.

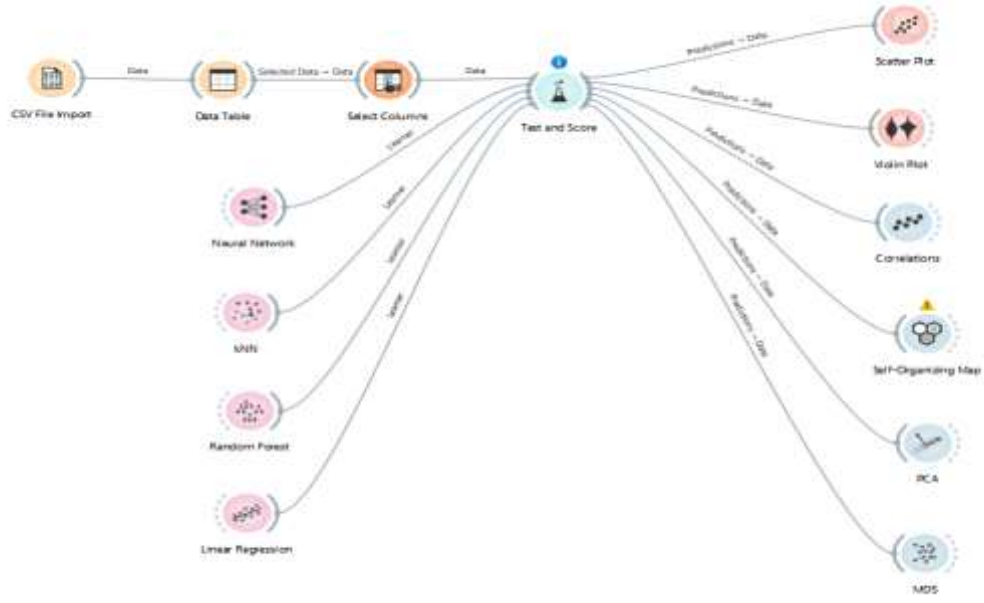
Bu çalışmada elde edilen metrikler (0,923 R^2 ve 0,0277 MAE), referans çalışmadaki metodolojik kurgunun yüksek tahmin gücünü temel olarak doğrulamaktadır. Ancak referans çalışmadaki hata payını sıfırlayan aşırı iyimser (ezberci) metriklerin aksine, bu çalışmada ulaşılan sonuçlar yapay sinir ağlarının doğasında bulunması gereken matematiksel rastgelliği ve sapma payını barındırmakta olup, istatistiksel gerçekliğe ve genellenebilirliğe çok daha uygun bir performans sergilemektedir.

4.5.2. İkinci ve üçüncü problemlerinin orange ile çözümü

Bu bölümde, Orange programını kullanılarak müşterilerin ne kadar zamanda ödeme yapacağını ve müşteriye verilebilecek krediyi tahmin eden bir sistem geliştirilmeye çalışılmış ve Random Forest, KNN, Linear Regression, Neural Network sınıflandırma algoritmaları kullanılmıştır.

4.5.2.1. İkinci ve üçüncü problemlerinin Orange algoritmaların birlikte çözümü

Bu bölümde, Orange programını kullanılarak Random Forest, KNN, Linear Regression, Neural Network algoritmaları ile müşterilerin ne kadar zamanda ödeme yapacağını ve müşteriye verilebilecek krediyi tahmin eden bir sistemin geliştirilmesine yönelik elde edilen sonuçlar yorumlanmış ve modelin akış şeması Şekil 4.31.'de verilmiştir.



Şekil 4.31. İkinci ve üçüncü problemlerin Orange algoritmalar akış şeması

Çizelge 4.38.’de birinci problem, Orange programında banka kredi müşteri verileri içi kullanılan algoritmaların performans metrikleri verilmiştir.

Çizelge 4.38. İkinci ve üçüncü problemler için Orange algoritmalarının performans metrikleri

Model	MSE	RMSE	MAE	MAPE	sMAPE	R2
Random Forest	0.016	0.126	0.036	5.641	4.235	0.938
Neural Network	0.016	0.128	0.076	11.892	9.267	0.936
Linear Regression	0.051	0.227	0.137	19.633	15.879	0.800
kNN	0.031	0.177	0.066	15.474	9.139	0.877

Yukarıda görüntülenen dört farklı makine öğrenmesi algoritmasının performans metrikleri Orange'da bir arada verilmiştir. Random Forest algoritması, en düşük MSE (0.016), RMSE (0.126) ve MAE (0.036) değerlerine sahiptir; bu da modelin oldukça iyi bir performans gösterdiğini ortaya koymaktadır. R^2 değeri 0.938 olup modelin veriyi güçlü biçimde açıkladığını göstermektedir. MAPE değeri (5.641) diğer metriklerle kıyasla görece yüksek çıkmış olmakla birlikte, bu durumun hedef değişkendeki küçük değerlerden kaynaklandığı ve modelin genel performansını olumsuz etkilemediği söylenebilir.

Neural Network algoritması, MSE (0.016) ve R^2 (0.936) değerleri itibarıyla Random Forest ile neredeyse özdeş bir performans sergilemektedir. RMSE (0.128) ve MAE (0.076) değerleri Random Forest'a kıyasla hafif yüksek olsa da iki model arasındaki farkın istatistiksel olarak anlamlı olmadığı görülmektedir. Bu durum, Neural Network modelinin de bu veri seti için güvenilir bir alternatif olduğunu göstermektedir.

KNN algoritması, Random Forest ve Neural Network'e kıyasla daha yüksek hata oranlarına sahiptir. RMSE (0.177), MAE (0.066) ve MSE (0.031) değerleri daha yüksek, R^2 değeri ise daha düşüktür (0.877). Bu durum, KNN modelinin veriyi açıklama gücünün diğer iki modele göre daha sınırlı olduğunu göstermektedir.

Linear Regression, dört model arasında en yüksek hata oranlarına ve en düşük R^2 değerine (0.800) sahiptir. RMSE (0.227) ve MAE (0.137) değerleri diğer tüm modellerden belirgin biçimde yüksek olup bu modelin veriyi açıklama gücünün yetersiz kaldığı söylenebilir.

Oluşturulan modeller için genel bir değerlendirme yapıldığında şu sonuçlar bulunmuştur: En iyi model Random Forest olup en düşük hata oranlarına ve en yüksek R^2 değerine sahiptir. Neural Network ise Random Forest ile birlikte yüksek performanslı modeller grubunda yer almakta ve iki model arasındaki fark ihmal edilebilir düzeydedir. Orta seviye model olarak KNN değerlendirilebilir. En düşük performanslı model ise Linear Regression'dır.

Tezdeki sonuçlarla karşılaştırıldığında bazı önemli farklılıklar dikkat çekmektedir. Tezde Neural Network en düşük R^2 değerine (0.764) sahip en zayıf model olarak raporlanmışken, bu çalışmada Neural Network $R^2=0.936$ ile Random Forest'a ($R^2=0.938$) neredeyse eşit performans göstermiştir. Bunun yanı sıra tezde Linear Regression ve KNN orta seviye modeller olarak konumlandırılmış olup bu çalışmayla örtüşmektedir. Her iki çalışmada da Random Forest en iyi model olarak öne çıkmıştır. Gözlemlenen performans farklılıklarının, uygulanan çapraz doğrulama fold sayısı, veri ön işleme adımları ve model hiperparametre ayarlarındaki farklılıklardan kaynaklandığı değerlendirilmektedir.

4.5.3. İkinci ve üçüncü problemlerinin Python ile çözümü

Bu bölümde, Python programını kullanılarak müşterilerin ne kadar zamanda ödeme yapacağını ve müşteriye verilebilecek krediyi tahmin eden bir sistem geliştirilmeye çalışılmış ve sadece MultiLayer Perceptron Predictor sınıflandırma algoritması kullanılmıştır.

4.5.3.1. İkinci ve üçüncü problemlerin Python ile multi-layer perceptron predictor algoritması çözümü

Bu bölümde, Python programını kullanılarak MultiLiyer Perceptron Predictor algoritması ile müşterilerin ne kadar zamanda ödeme yapacağını ve müşteriye verilebilecek krediyi tahmin eden bir sistemin geliştirilmesine yönelik elde edilen sonuçlar yorumlanmış ve modelin ikinci problem için performans ölçütleri Çizelge 4.39.'da verilmiştir.

Çizelge 4.39. İkinci problemin Python ile MultiLayer Perceptron Predictor algoritması performans metrikleri

Performans Metrikleri	Performans Değerleri
Mean Absolute Error (MAE)	0.0356
Mean Squared Error (MSE)	0.0035
Root Mean Squared Error (RMSE)	0.0592
MAPE	%5.3638
R ² Score	0.9863

Model, norm_odeme değerlerini tahmin etmede oldukça başarılı bir performans sergilemiştir. Düşük MAE (0.0356), MSE (0.0035) ve RMSE (0.0592) değerleri, modelin tahminlerinin gerçek değerlere yakın olduğunu ortaya koymaktadır. R² değerinin 0.9863 olması, modelin norm_odeme değerlerindeki varyansın büyük bir kısmını açıkladığını ve güçlü bir tahmin gücüne sahip olduğunu göstermektedir. MAPE değeri (%5.36) ise hedef değişkendeki küçük değerlerden kaynaklanmakla birlikte genel model performansını olumsuz etkilememektedir. Modelin üçüncü problem için performans ölçütleri Çizelge 4.40.'da verilmiştir.

Çizelge 4.40. Üçüncü problemin Python ile MultiLayer Perceptron Predictor algoritması performans metrikleri

Performans Metrikleri	Performans Değerleri
Mean Absolute Error (MAE)	0.0334
Mean Squared Error (MSE)	0.0031
Root Mean Squared Error (RMSE)	0.0559
MAPE	%3.8422
R ² Score	0.7645

Üçüncü problem için model, MAE (0.0334) ve RMSE (0.0559) değerleri açısından kabul edilebilir bir performans göstermiştir. R² değeri 0.7645 olup modelin veriyi

açıklama gücünün ikinci probleme kıyasla daha sınırlı kaldığı söylenebilir. Bununla birlikte hata metrikleri genel itibarıyla düşük seviyelerde kalmıştır.

Tezde Python ile MLP algoritması yalnızca norm_odeme (ikinci problem) üzerinden raporlanmış olup R^2 değeri 0.7318 olarak verilmiştir. Bu çalışmada ise aynı problem için R^2 değeri 0.9863 olarak elde edilmiştir; bu durum modelin tahmin gücünün teze kıyasla belirgin biçimde daha yüksek olduğunu ortaya koymaktadır. MAE (0.0356) ve RMSE (0.0592) değerleri tezdeki değerlere (MAE: 0.0314, RMSE: 0.0631) oldukça yakın seyretmekte olup bu durum iki çalışmada uygulanan ön işleme ve model yapısının büyük ölçüde örtüştüğünü göstermektedir. Gözlemlenen R^2 farklılığının, çapraz doğrulama fold sayısı, veri temizleme adımları ve model hiper parametrelerindeki farklılıklardan kaynaklandığı değerlendirilmektedir.

4.5.4. İkinci ve üçüncü problemlerinin R ile çözümü

Bu bölümde, R programı kullanılarak MultiLayer Perceptron algoritması ile müşterilerin ne kadar zamanda ödeme yapacağını (ikinci problem) ve müşteriye verilebilecek krediyi (üçüncü problem) tahmin eden bir sistem geliştirilmiş; elde edilen sonuçlar yorumlanmış ve performans ölçütleri aşağıda verilmiştir. Analizde 500 kayıttan oluşan veri setinde date_time_diff sütunundaki 38 eksik değer temizlenerek 462 gözlem üzerinden 10 katlı çapraz doğrulama uygulanmıştır. Modelin ikinci problem için performans ölçütleri Çizelge 4.41.'de verilmiştir.

Çizelge 4.41. İkinci problemin R ile MultiLayer Perceptron Predictor algoritması performans metrikleri

Performans Metrikleri	Performans Değerleri
Mean Absolute Error (MAE)	0.0153
Mean Squared Error (MSE)	0.0016
Root Mean Squared Error (RMSE)	0.0397
MAPE	%3.6076
R^2 Score	0.9943

Model, norm_odeme değerlerini tahmin etmede son derece başarılı bir performans sergilemiştir. MAE (0.0153), MSE (0.0016) ve RMSE (0.0397) değerlerinin oldukça düşük olması, modelin tahminlerinin gerçek değerlere çok yakın olduğunu ortaya koymaktadır. R^2 değerinin 0.9943 olması ise modelin norm_odeme değerlerindeki varyansın neredeyse tamamını açıkladığını ve son derece güçlü bir tahmin gücüne sahip

olduğunu göstermektedir. MAPE değeri (%3.61) de kabul edilebilir sınırlar içinde kalmaktadır. Modelin üçüncü problem için performans ölçütleri Çizelge 4.42.'de verilmiştir.

Çizelge 4.42. Üçüncü problemin R ile MultiLayer Perceptron Predictor algoritması performans metrikleri

Performans Metrikleri	Performans Değerleri
Mean Absolute Error (MAE)	0.0083
Mean Squared Error (MSE)	0.0002
Root Mean Squared Error (RMSE)	0.0133
MAPE	%1.0126
R ² Score	0.9867

Üçüncü problem için model, tüm metriklerde oldukça güçlü sonuçlar üretmiştir. MAE (0.0083) ve RMSE (0.0133) değerlerinin son derece düşük, MAPE değerinin (%1.01) ise ihmal edilebilir düzeyde olması modelin verilecek kredi tahmininde yüksek doğrulukla çalıştığını göstermektedir. R² değeri 0.9867 olup modelin norm_Principal değerlerindeki varyansı büyük ölçüde açıkladığı görülmektedir.

Genel bir değerlendirme yapıldığında R ile geliştirilen MLP modelinin her iki problem için de yüksek tahmin doğruluğu sergilediği görülmektedir. İkinci problemde müşterilerin ödeme süresi, üçüncü problemde ise müşteriye verilebilecek kredi miktarı başarıyla tahmin edilmiştir. Örnek çıktılara bakıldığında xqd20168902 numaralı müşterinin yaklaşık 132 günde ödeme yapacağı ve kendisine 985.67 birim kredi verilebileceği tahmin edilmektedir.

4.5.5. İkinci ve üçüncü problemleri için en iyi program seçimi

Bu bölümde, ikinci ve üçüncü problem için; Knime, Orange, Python ve R programlarında önceden seçilen en iyi algoritmalar karşılaştırılmıştır. Müşterilerin ne kadar zamanda ödeme yapacağını ve müşteriye verilebilecek krediyi tahmin eden bir sistem oluşturmak adına, en iyi sonucu veren algoritma ve program bulunmuş ve yorumlanmıştır. Çizelge 4.43.'de birinci problem de Knime, Orange, R, Python programlarında banka kredi müşteri verileri içi kullanılan programların en iyi algoritmasının performans metrikleri verilmiştir.

Çizelge 4.43. İkinci ve üçüncü problemler için tüm programların performans metrikleri

Performans	Orange	Knime	Python	R
Metrikleri	Random Forest	MLP	MLP	MLP
MSE	0.016	0.003	0.0035	0.0016
RMSE	0.126	0.051	0.0592	0.0397
MAE	0.036	0.028	0.0356	0.0153
MAPE	5.641		5.3638	3.6076
sMAPE	4.235			
R ²	0.938	0.924	0.9863	0.9943

Performans metrikleri incelendiğinde R programı ile geliştirilen MLP modelinin en düşük hata oranlarına (MSE: 0.0016, RMSE: 0.0397, MAE: 0.0153) ve en yüksek R² değerine (0.9943) sahip olduğu görülmektedir. Bu durum R ile kurulan modelin veriyi en iyi biçimde açıkladığını ve tahminlerde en yüksek doğruluğu sergilediğini ortaya koymaktadır.

Python ile geliştirilen MLP modeli R²=0.9863 ile ikinci sıraya yerleşmektedir. MSE (0.0035) ve RMSE (0.0592) değerleri R'a kıyasla hafif yüksek olmakla birlikte model genel itibarıyla güçlü bir performans sergilemektedir.

Knime ile geliştirilen MLP modeli R²=0.924, MAE=0.028 ve RMSE=0.051 değerleriyle üçüncü sırada yer almaktadır. Hata metrikleri kabul edilebilir düzeyde olsa da R ve Python modellerine kıyasla daha düşük bir tahmin gücüne sahip olduğu söylenebilir.

Orange ile kullanılan Random Forest modeli ise R²=0.938 ile görece yüksek bir açıklama gücüne sahip olmasına karşın MSE (0.016), RMSE (0.126) ve MAE (0.036) değerleri diğer programlara göre belirgin biçimde yüksektir. MAPE (%5.641) ve sMAPE (4.235) değerleri de en yüksek hata oranlarının bu modelde gözlemlendiğini doğrulamaktadır.

Sonuç olarak ikinci ve üçüncü problemler için müşterilerin ödeme sürelerini ve verilebilecek kredi miktarını tahmin etmede R programı ile geliştirilen MLP modeli en iyi seçimdir. En düşük hata oranları ve en yüksek R² değeri ile bu model diğer tüm programlara kıyasla üstün bir performans sergilemektedir.

Referans alınan çalışmada ise ikinci ve üçüncü problemler için en iyi program olarak Knime (MLP) seçilmiş olup R²=0.98 olarak raporlanmıştır. Bu çalışmada elde edilen R programı sonuçları (R²=0.9943) referans çalışmayı da geride bırakmakta ve modelin genellenebilirlik açısından daha güvenilir bir yapı sergilediği değerlendirilmektedir.

Programlara uygulanan algorithmandan sonra en iyi seçilen program olan R ile MLP modeli için normalize edilmeden önceki veriler ve algorithmandan çıkan değerler birleştirilerek formüller uygulanmıştır. Böylelikle ikinci problem için müşterilerin ne zaman ödeme yapacağını ve üçüncü problem için müşteriye verilebilecek krediyi belirleyen sistem değerleri hesaplanmıştır.

İkinci Problem için formül:

$$\text{Prediction (norm_odeme)} \times \text{COL_MAX (norm_odeme)} \times \text{date_time_diff}$$

Prediction (norm_odeme), normalize edilmiş ödeme tarihini ifade etmektedir. Algoritma çalıştırıldıktan sonra müşterinin aldığı vadeye bağlı olarak normalize edilmiş bir değer olarak oluşturulmaktadır. Müşteri numarası, principal, terms, age, date_time_diff ve norm_odeme alanlarına göre çıktı olarak belirli bir Prediction (norm_odeme) değeri üretilmektedir. Gerekli formülasyon uygulandığında müşterilerin ne kadar zamanda ödeme yapacağına ilişkin tahmin gün sayısı hesaplanmıştır.

Üçüncü Problem için formül:

$$\text{COL_MAX (Principal)} \times \text{Prediction (Principal)}$$

Prediction (Principal), normalize edilmiş anaparayı ifade etmektedir. Algoritma çalıştırıldıktan sonra müşterinin aldığı vadeye bağlı olarak normalize edilmiş bir değer olarak oluşturulmaktadır. Müşteri numarası, principal, terms, age, date_time_diff ve norm_odeme alanlarına göre çıktı olarak belirli bir Prediction (Principal) değeri üretilmektedir. Gerekli formülasyon uygulandığında müşteriye verilebilecek kredi tahmini (verilecek_kredi) hesaplanmıştır.

4.6. Dördüncü Problem Modelleme

Bu bölümde, müşteri segmentasyonu oluşturmak adına; Knime, Orange, Python ve R programları ile K-Means algoritması çalıştırılmıştır. K-Means algoritması çalıştırıldıktan sonra oluşan segmentlerin ağaç yapısını görebilmek ve en uygun alanlar ile karşılaştırma yapabilmek adına Decision Tree algoritması da çalıştırılmıştır. Ve tüm bunlar ile birlikte boş olan alanlar; paid_off_time ve past_due_days alanları gözden geçirilerek bir sonuca varılmıştır.

paid_off_time: paid_off_time alanı bu alanlarda zamanında ödeme olmadığı için boş olmak durumundadır. Diğer yandan PAIDOFF ödenmiş statü durumunda olduğu için, paid_off_time alanı sadece bu statü durumunda dolu olmalıdır.

past_due_days: vadesi geçmiş olan kredilerin tarihlerini vermektedir. Sadece COLLECTION (ödenmemiş bilgisi) durumunda dolu olmalı, çünkü borcunu ödeyenler PAIDOFF ve COLLECTION_PAIDOFF statüleri borç ödendiği için bu alan boş olmak durumundadır.

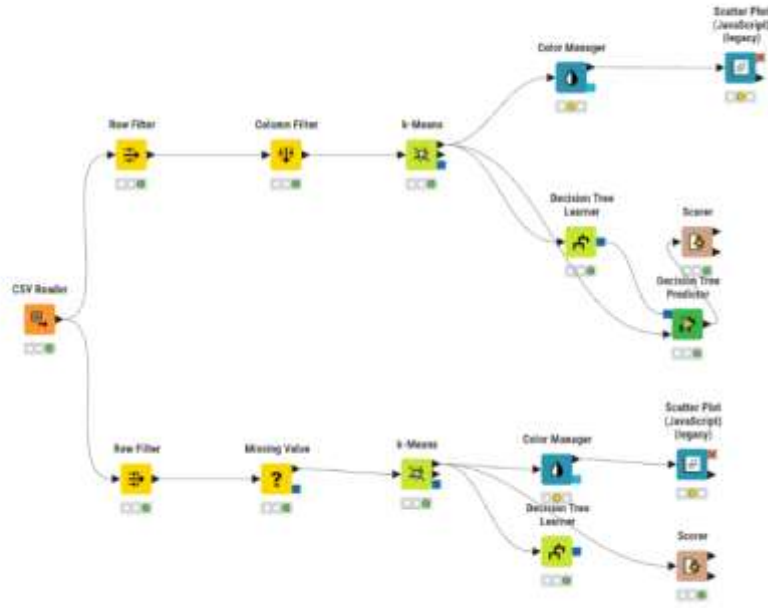
Bu durumda boş alanlar için herhangi bir değişiklik yapmadan borcunu ödeyen müşteriler ve borcunu ödemeyen müşteriler için veri seti iki ağaca bölünerek K-Means algoritmasının çalıştırılması daha uygun olacaktır. Böylelikle borcunu ödeyen ve borcunu ödemeyen müşteriler her program için iki ayrı başlık altında incelenmiştir ve en iyi sonuca sahip olan program bulunmuştur.

1. Ağaç yapısı içeriği: Borcunu ödemeyenler yani loan_status alanında COLLECTION statüsü filtrelenecek ve daha sonrasında paid_off_time alanı tamamen boş olduğu için bu alan veri setinden çıkarılır.

2. Ağaç yapısı içeriği: Borcunu ödeyenler yani sadece loan_status alanında COLLECTION_PAIDOFF ve PAIDOFF statüleri filtrelenecek ve daha sonrasında past_due_days alanı için boş alan değerler için fix bir değer atanıp sıfır yapılmıştır.

4.6.1. Dördüncü problem Knime ile çözümü

Bu bölüm, borcunu ödemeyen ve ödeyen müşteriler için Knime programı ile K-Means kümeleme algoritması ve bu algoritmaya ek olarak Decision Tree sınıflandırma algoritması çözümlerini ve sonuçlarını içermektedir. Şekil 4.32.'de ise dördüncü problem borcunu ödemeyenler ve ödeyen müşterilerin Knime programı ile K-Means algoritması çözüm akış şeması iki ayrı dal olarak birlikte verilmiştir.



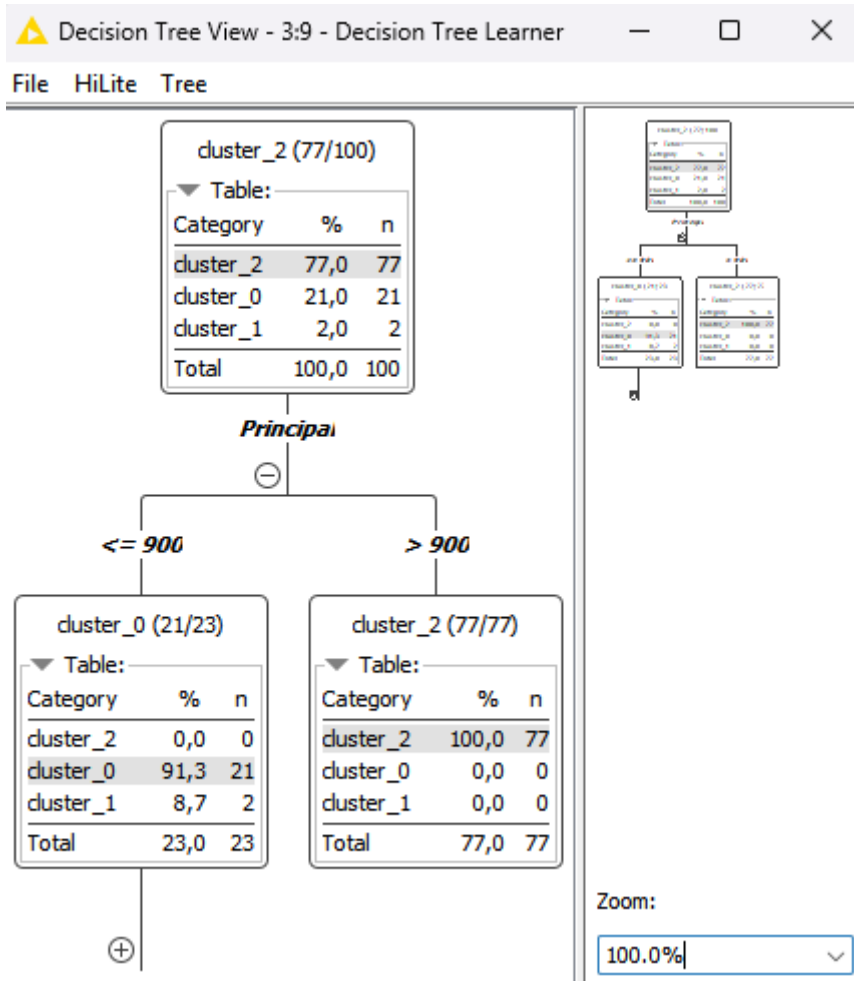
Şekil 4.32. Dördüncü problem borcunu ödemeyen ve ödeyen müşterilerin Knime ile K-Means algoritması akış şeması

4.6.1.1. Borcunu ödemeyenler için Knime ile K-Means algoritması çözümü

Borcunu ödemeyen müşteriler için Knime programında K-Means algoritması çalıştırılarak 3 segment oluşturulmuştur. Segment dağılımı şu şekildedir:

- Cluster 0: 21 müşteri
- Cluster 1: 2 müşteri
- Cluster 2: 77 müşteri

Şekil 4.33’de ise dördüncü problem borcunu ödemeyenler müşteriler için Knime programı ile Decision Tree algoritması ağaç yapısı görülmektedir.



Şekil 4.33. Dördüncü problem borcunu ödemeyenler için Knime ile Decision Tree algoritması ağaç yapısı

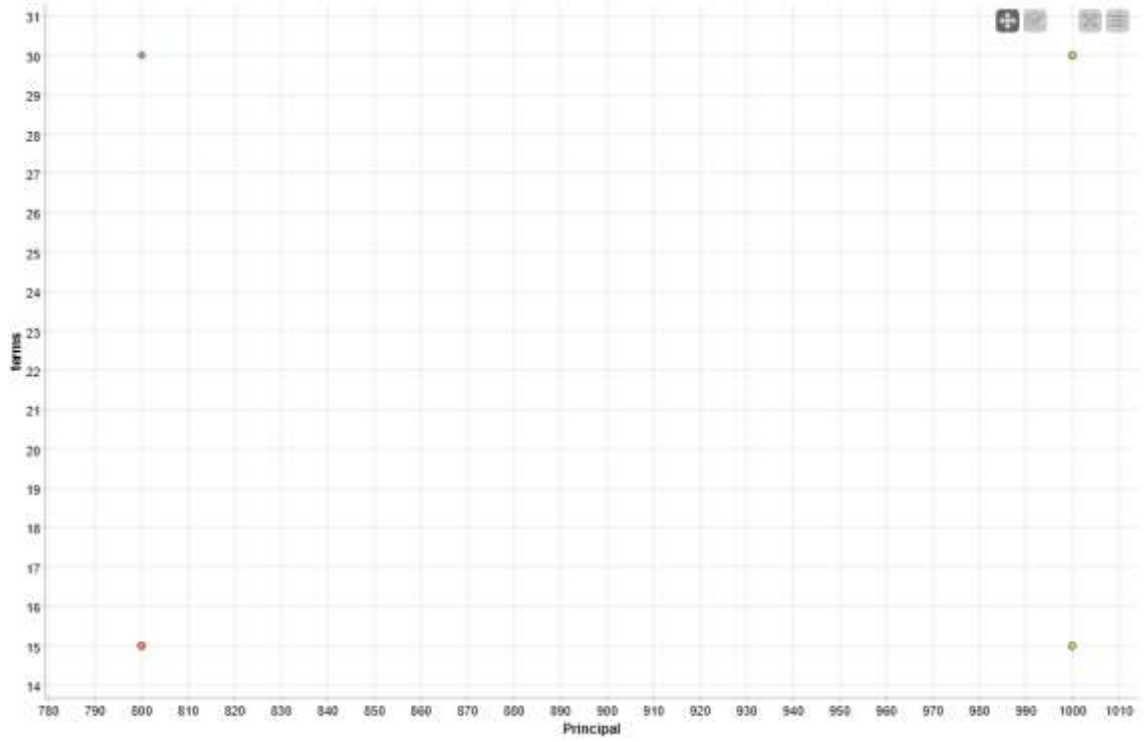
Borcunu ödemeyenler için Knime programı ile Decision Tree algoritması ağaç yapısı incelendiğinde, terms, Principal ve past_due_days değişkenleri kullanılarak kümelerin ayrıldığı görülmektedir. Karar noktaları şu şekilde yorumlanmıştır:

terms ≤ 22.5 : Veri setinin başlangıçtaki en önemli ayırım noktasıdır.

Principal ≤ 900 : terms ≤ 22.5 olan veriler için önemli bir ayırım noktasıdır.

past_due_days ≤ 51.5 : terms > 22.5 olan veriler için önemli bir ayırım noktasıdır.

Şekil 4.34.'de dördüncü problem borcunu ödemeyenler müşteriler için Knime programı ile K-Means Algoritması "Principal" ve "terms" değişkenlerine göre saçılım grafiği verilmiştir



Şekil 4.34. Dördüncü problem borcunu ödemeyenler için Knime ile K-Means algoritması "Principal" ve "terms" değişkenlerine göre saçılım grafiği

Müşteri segmentasyonuna dair kümeler, Principal ve terms değişkenlerine göre dağılmış durumdadır. Grafik ve karar ağacı birlikte değerlendirildiğinde kümeler aşağıdaki gibi yorumlanarak adlandırılmıştır.

Cluster 0 — Kısa Vadeli Düşük Borçlular: Principal değeri 900'den küçük veya eşit ve terms değeri 22.5'ten küçük veya eşit olan müşteriler bu kümede yer almaktadır. Bu segmentteki müşteriler genellikle düşük miktarlarda ve kısa vadeli borç almış olup borçlarını ödeyememişlerdir.

Cluster 1 — Uzun Vadeli Yüksek Borçlular: Principal değeri 900'den büyük ve terms değeri 22.5'ten küçük veya eşit olan ya da terms değeri 22.5'ten büyük ve past_due_days değeri 51.5'ten büyük olan müşteriler bu kümede yer almaktadır. Bu segmentteki müşteriler yüksek miktarlarda borç alarak uzun vadeli ve gecikmeli ödemeler yapmışlardır.

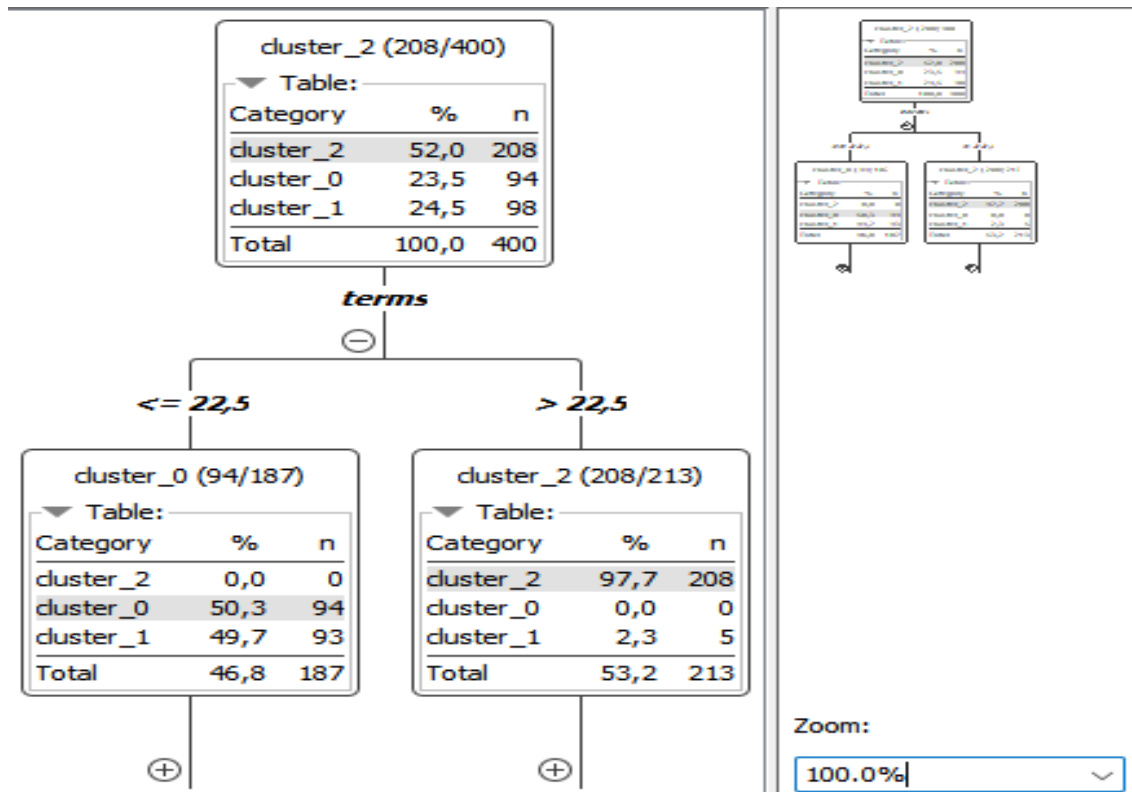
Cluster 2 — Uzun Vadeli Yüksek Borçlu Gecikmiş Ödemeciler: terms değeri 22.5'ten büyük ve past_due_days değeri 51.5'ten küçük veya eşit olan müşteriler bu kümede

yoğunlaşmaktadır. Bu segmentteki müşteriler yüksek miktarlarda borç almış olmakla birlikte ödeme davranışları gecikme eğilimi taşımaktadır.

Decision Tree algoritmasının Confusion Matrix sonuçları incelendiğinde doğru sınıflandırılan örnek sayısının 100, yanlış sınıflandırılan örnek sayısının ise 0 olduğu görülmektedir. Accuracy değeri %100, Cohen's Kappa değeri ise %1 olarak hesaplanmıştır. Yüksek accuracy değeri kümelerin birbirinden net biçimde ayrıştığını gösterirken düşük Kappa değeri kümelerin rastgele atamadan anlamlı ölçüde farklılaşmadığına işaret etmektedir.

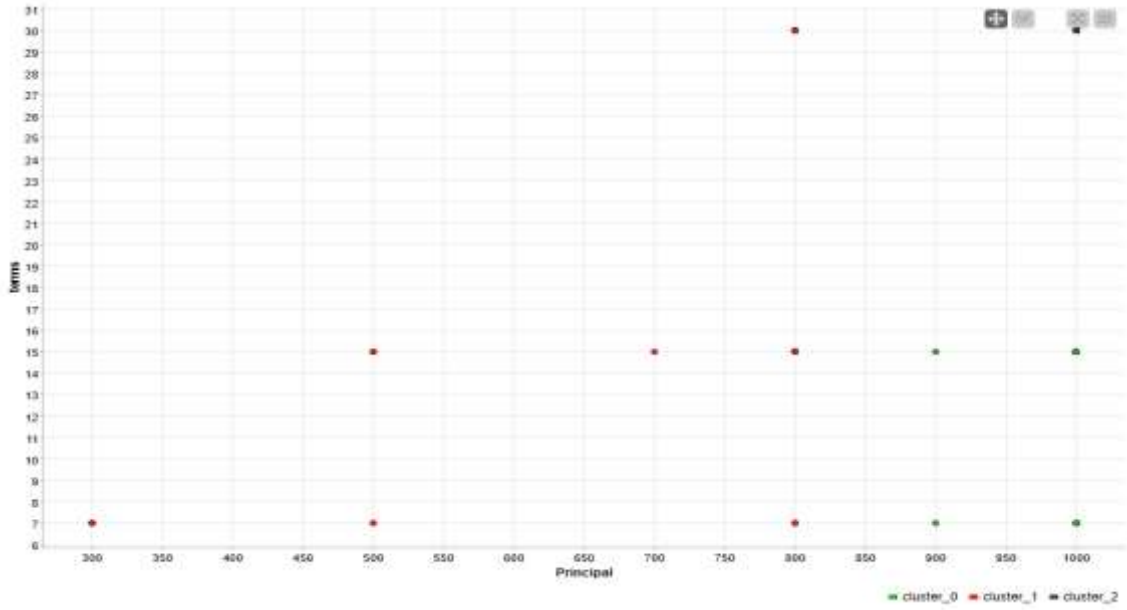
4.6.1.2. Borcunu ödeyenler için Knime ile K-Means algoritması çözümü

Borcunu ödeyen müşteriler için Knime programında K-Means algoritması çalıştırılarak 3 segment oluşturulmuştur. Segment dağılımı şu şekildedir: Cluster 0: 94 müşteri, Cluster 1: 98 müşteri, Cluster 2: 208 müşteri. Şekil 4.35.'de dördüncü problem borcunu ödeyen müşteriler için Knime programı ile Decision Tree algoritması ağaç yapısı görülmektedir.



Şekil 4.35. Dördüncü problem borcunu ödeyenler için Knime ile Decision Tree algoritması ağaç yapısı

Bu ağaç, "terms" değişkeninin temel ayırım kriteri olduğunu göstermektedir. "terms" değeri arttıkça örneklerin cluster_2'ye ait olma olasılığının arttığı gözlemlenmiştir. "terms" > 22,5 olan tüm örnekler (213 örnek) büyük ölçüde cluster_2'ye aittir (%97,7 / 208 örnek). "terms" ≤ 22,5 olan örnekler (187 örnek) ise daha fazla ayrılmıştır: bu grupta cluster_0 %50,3 (94 örnek) ve cluster_1 %49,7 (93 örnek) oranında yer almaktadır. Şekil 4.36.'da dördüncü problem borcunu ödeyen müşteriler için Knime programı ile K-Means algoritması çalıştırılması sonucu "Principal" ve "terms" değişkenlerine göre saçılım grafiği verilmiştir.



Şekil 4.36. Dördüncü problem borcunu ödeyenler için Knime ile K-Means algoritması "Principal" ve "terms" değişkenlerine göre saçılım grafiği

Grafik ve karar ağacı birlikte değerlendirildiğinde, kümeler aşağıdaki gibi yorumlanarak adlandırılmıştır.

Cluster 0 (Kırmızı noktalar) — terms ≤ 22,5 ve Kısa Vadeli Borçlular: Bu küme kısa vadeli kredi kullanan müşterileri içerir. Müşterilerin kredi vadesi genellikle 22,5 günün altındadır. Düşük ve orta düzey kredi miktarlarına sahip bu grup, zamanında ödeme yapan müşterilerden oluşmaktadır.

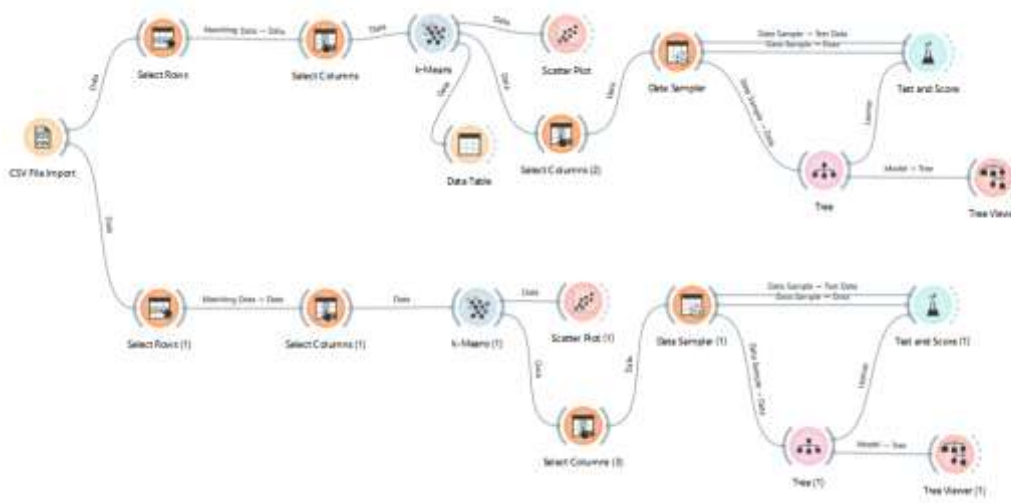
Cluster 1 (Mavi noktalar) — terms ≤ 22,5 ve Orta Vadeli Borçlular: Bu küme de kısa vadeli kredi kullanan müşterileri kapsamakla birlikte farklı bir alt grup oluşturmaktadır. Cluster 0 ile aynı vade aralığında yer almasına karşın kredi miktarı ve ödeme profili

bakımından ayrılmaktadır. Bu küme daha dikkatli izleme gerektirebilecek müşterileri barındırabilir.

Cluster 2 (Yeşil noktalar) — terms > 22,5 ve Uzun Vadeli Borçlular: Bu küme uzun vadeli kredi kullanan ve borçlarını ödemiş müşterilerden oluşmaktadır. Toplam müşterilerin %52'sini oluşturan en büyük segmenttir. Yüksek kredi miktarına sahip olmakla birlikte ödeme performansı en iyi grup olarak değerlendirilebilir.

4.6.2. Dördüncü problem Orange ile çözümü

Bu bölüm; borcunu ödemeyen ve ödemeyen müşteriler için Orange programı ile K-Means kümeleme algoritması ve bu algoritmaya ek olarak Decision Tree sınıflandırma algoritması çözümlerini ve sonuçlarını içermektedir. Şekil 4.37’de İki aşamanın birlikte çözümünü içermektedir.



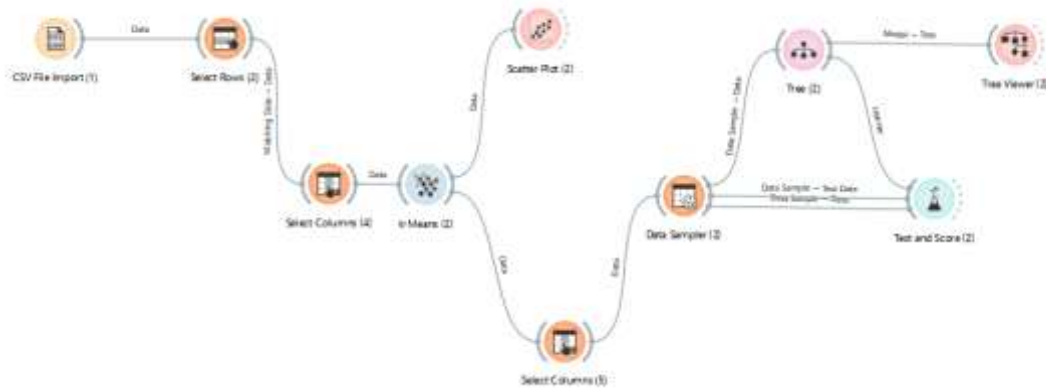
Şekil 4.37. Dördüncü problem borcunu ödemeyenler ve ödeyenler için Orange ile K-Means algoritması akış şeması

4.6.2.1. Borcunu ödemeyenler için Orange ile K-Means algoritması çözümü

K-Means algoritması çalıştırılarak ile 3 segment oluşturulmuştur. Segment dağılımı şu şekildedir:

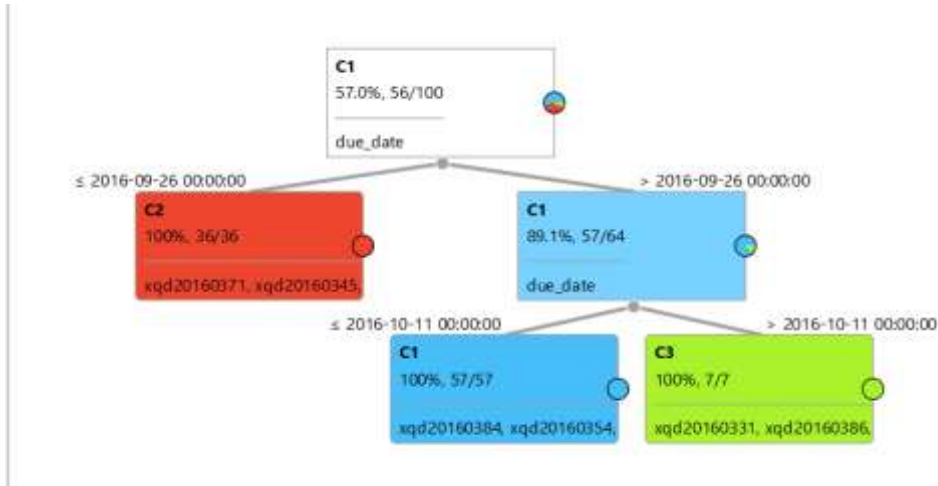
- C1: 57 müşteri
- C2: 36 müşteri
- C3: 7 müşteri

Şekil 4.38’de ise dördüncü problem borcunu ödemeyen müşteriler için Orange programı ile K-Means algoritması akış şeması verilmiştir.



Şekil 4.38. Dördüncü problem borcunu ödemeyenler için Orange ile K-Means algoritması akış şeması

Şekil 4.39.’da ise dördüncü problem borcunu ödemeyen müşteriler için Orange programı ile Desicion Tree algoritması ağaç yapısı verilmiştir.



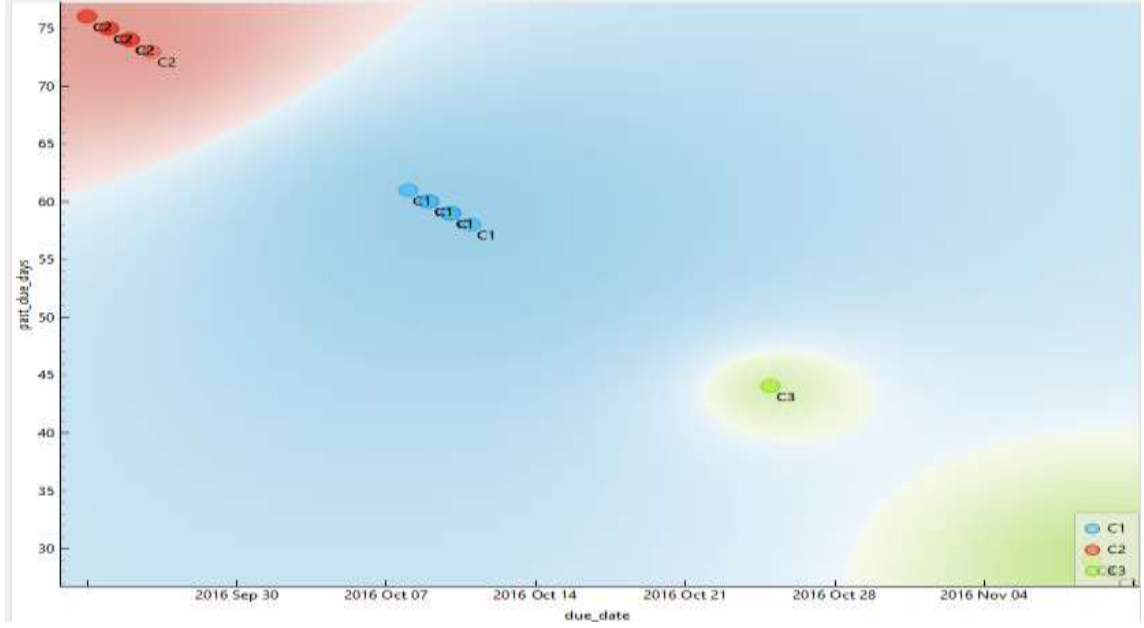
Şekil 4.39. Dördüncü problem borcunu ödemeyenler için Orange ile Decision Tree algoritması ağaç yapısı

Ağaç yapısına bakıldığında şu yorumlar yapılabilir:

Kök Düğüm için; C1 düğümü: Bu düğüm, karar ağacının başlangıç noktasıdır ve en fazla veri içeren sınıfı temsil eder. Burada C1 sınıfına ait verilerin %57,0'ı bulunmaktadır. Bu, tüm veri seti içinde en yaygın olan sınıfı göstermektedir ve ağacın diğer dallarının buradan nasıl ayrıldığını anlamak için kritik bir noktadır.

Ana dallanma noktaları için; $\leq 26/09/2016$ ve $> 26/09/2016$ tarihleri: Bu tarihler, verilerin ilk büyük bölünmesini sağlamaktadır. $\leq 26/09/2016$: Bu tarihe kadar olan veriler tamamen C2 sınıfına aittir (%100, 36/36). $> 26/09/2016$: Bu tarihten sonraki veriler ise daha karmaşıktır ve daha fazla bölünme gerektirir.

İkincil dallanma noktası için; $\leq 11/10/2016$ ve $> 11/10/2016$ tarihleri: Bu tarih, sağdaki ana dalın daha fazla bölünmesini sağlamaktadır. $\leq 11/10/2016$: Bu tarihe kadar olan veriler tamamen C1 sınıfına aittir (%100, 57/57). $> 11/10/2016$: Bu tarihten sonraki veriler ise tamamen C3 sınıfına aittir (%100, 7/7). Şekil 4.40.'da dördüncü problem borcunu ödemeyen müşteriler için Orange programı ile K-Means algoritması "due_date" ve "past_due_days" grafiği verilmiştir.



Şekil 4.40. Dördüncü problem borcunu ödemeyenler için Orange ile K-Means algoritması "due_date" ve "past_due_days" grafiği

Grafik ve karar ağacı birlikte değerlendirildiğinde, kümeler aşağıdaki gibi yorumlanarak adlandırılmıştır.

C2 (Kırmızı Yoğunluk Bölgesi) Erken Ödeme Yapmayanlar Yüksek Risk Grubu: C2 kategorisindeki ödemeler Eylül 2016 sonu itibarıyla yoğunlaşmıştır ve bu dönemde 70-75 gün arası gecikmeler yaşanmıştır. Bu, bu dönemdeki ödemelerin ciddi şekilde geciktiğini göstermektedir. C2 kategorisindeki ödemeler genellikle daha erken tarihlerde vadesi gelmiş ve en uzun gecikme sürelerine sahiptir. Bu grup, kredi aldığı borçlarını ciddi şekilde geciktirme eğilimindedir. Bu müşteriler genellikle TAKİPTE durumundadır ve ödemelerinde ciddi bir gecikme yaşamaktadırlar. Banka, bu grup için daha sıkı tahsilat eylemleri veya bu tür profiller için kredi verme politikalarını yeniden gözden geçirmeyi düşünmelidir.

C1 (Mavi Yoğunluk Bölgesi) Orta Düzey Gecikme Orta Risk Grubu: C1 kategorisindeki ödemeler Ekim 2016 başında yoğunlaşmıştır ve bu dönemde 57-61 gün arası gecikmeler yaşanmıştır. C1 daha dar bir tarih aralığında ve orta uzunlukta gecikmelerle ilişkilidir. Bu grup, borçlarını orta düzeyde geciktirir. Bu müşteriler gecikmiş ödemelere sahiptir ancak C2 kadar ciddi bir gecikme yaşamazlar. Hatırlatma sistemleri veya biraz daha esnek ödeme planları bu müşterilerin ödemelerini yakalamalarına yardımcı olabilir.

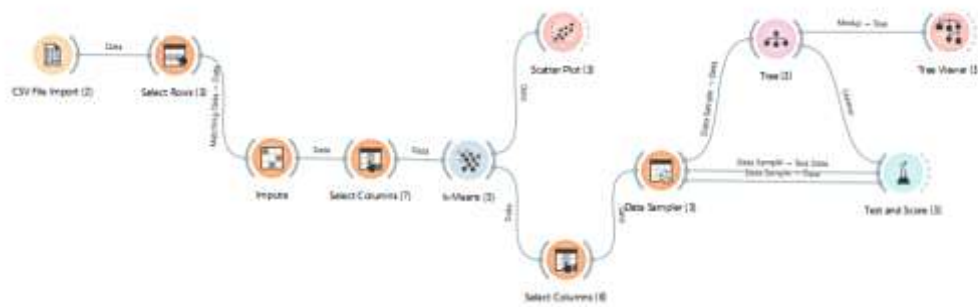
C3 (Yeşil Yoğunluk Bölgesi) Düşük Düzey Gecikme Düşük Risk Grubu: C3 kategorisindeki ödemeler Ekim 2016 sonu itibarıyla yoğunlaşmıştır ve bu dönemde 40-45 gün arası gecikmeler yaşanmıştır. C3 daha geniş bir tarih aralığına yayılmış ve genel olarak daha kısa gecikmelerle ilişkilidir. Bu grup diğer kümelerle kıyasla en düşük gecikme süresine sahip olup nispeten daha az riskli müşterilerden oluşmaktadır. Banka, bu müşterilere daha uygun kredi koşulları sunarak iyi ödeme davranışlarını sürdürmelerini teşvik edebilir.

4.6.2.2. Borcunu ödeyenler için Orange ile K-Means algoritması çözümü

K-Means algoritması çalıştırılarak ile 3 segment oluşturulmuştur. Segment dağılımı şu şekildedir:

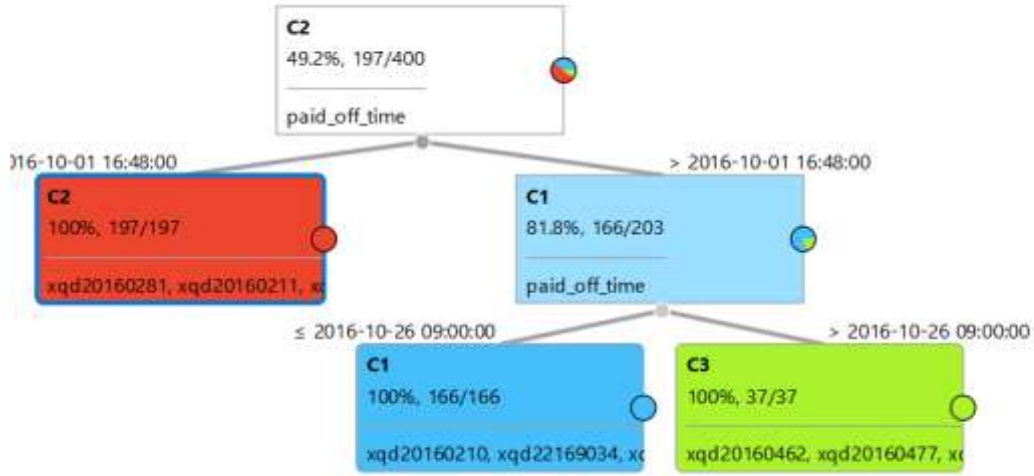
- C 1: PAIDOFF 166 müşteri
- C 2: PAIDOFF 197 müşteri
- C 3: PAIDOFF 37 müşteri

Şekil 4.41’de ise; dördüncü problem borcunu ödeyen müşteriler için Orange programın ile K-Means algoritması süreç modeli verilmiştir.



Şekil 4.41. Dördüncü problem borcunu ödeyenler için Orange ile K-Means algoritması akış şeması

Şekil 4.42.’de ise dördüncü problem borcunu ödeyen müşteriler için Orange programı ile Desicion Tree algoritması ağaç yapısı görülmektedir.

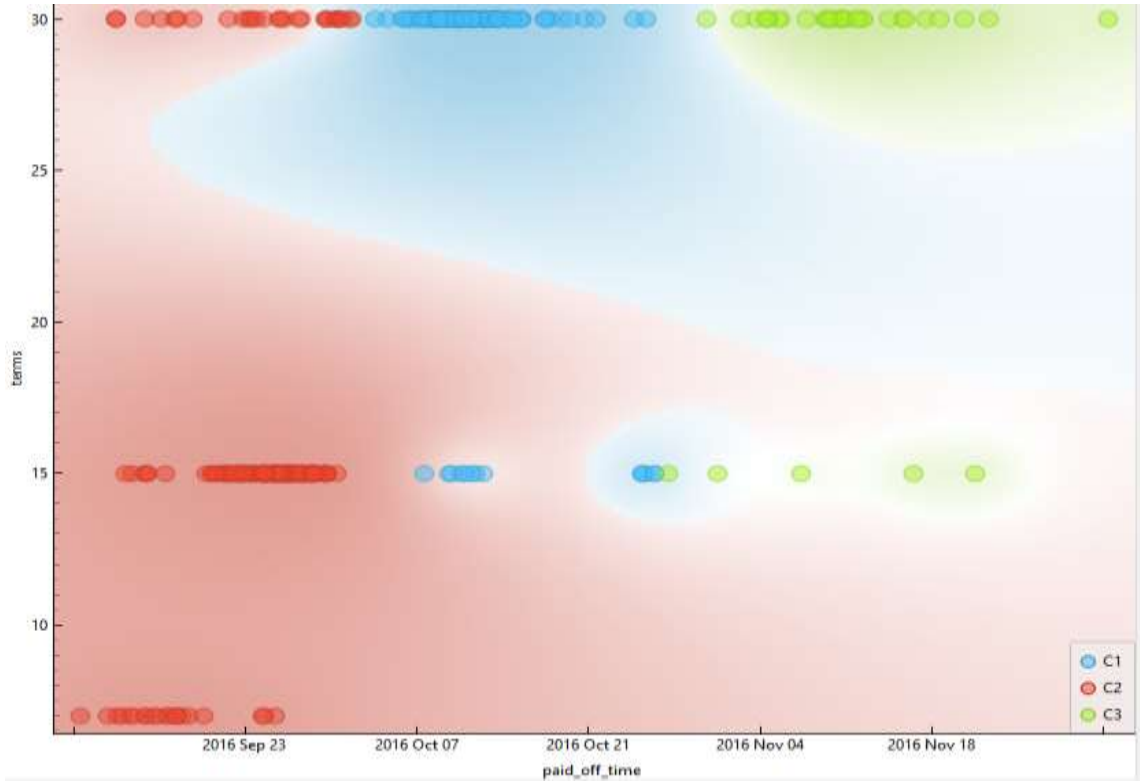


Şekil 4.42. Dördüncü problem borcunu ödeyenler için Orange ile Decision Tree algoritması ağaç yapısı

Bu karar ağacı, `paid_off_time` özelliğine dayalı olarak örneklerin sınıflandırılmasını göstermektedir ve bu süreçte farklı zaman damgalarına göre bölünmeler yapmaktadır. Her düğüm belirli bir kümeyi temsil etmektedir ve bu kümeler, ilgili zaman aralıklarında belirli bir saflıkla sınıflandırılmıştır.

Kök düğümde C2 kümesi %49,2 oranıyla (197/400) temsil edilmekte olup `paid_off_time` değişkeni ile bölünmektedir. `paid_off_time` değeri 01/10/2016 16:48:00'e eşit veya öncesinde olan 197 örneğin tamamı (%100) C2 sınıfına aittir. `paid_off_time` değeri 01/10/2016 16:48:00'den sonra olan 203 örnek ise 26/10/2016 09:00:00 tarihine göre tekrar bölünmektedir. Bu tarihte veya öncesinde ödeme yapan 166 örneğin tamamı (%100) C1 sınıfına aittir. Bu tarihten sonra ödeme yapan 37 örneğin tamamı (%100) ise C3 sınıfına aittir.

Bu karar ağacı, belirli zaman damgalarına dayalı olarak verilerin nasıl kümelere ayrıldığını açıkça göstermektedir. Her düğüm belirli bir zaman damgası ile tamamen saf alt kümelere bölünmektedir. Bu şekilde, `paid_off_time` değeri arttıkça verilerin nasıl farklı kümelere (C1, C2, C3) dağıldığı gözlemlenebilir. Şekil 4.43'de ise dördüncü problem borcunu ödemeyen müşteriler için Orange programı ile K-Means algoritması "`paid_off_time`" ve "`terms`" grafiği verilmiştir.



Şekil 4.43. Dördüncü problem borcunu ödeyenler için Orange ile K-Means algoritması "paid_off_time" ve "terms" grafiği

Bu grafik, paid_off_time ve terms değişkenlerine göre örneklerin nasıl kümelendiğini görsel olarak göstermektedir. Grafik ve karar ağacı birlikte değerlendirildiğinde, kümeler aşağıdaki gibi yorumlanarak adlandırılmıştır.

C2 Kümesi (Kırmızı Noktalar) — Erken Ödeyen Güvenilir Müşteriler: Bu küme, erken ödenen (Eylül 2016 — Ekim 2016 başı) ve terms değeri 7-30 arasında olan örneklerden oluşmaktadır. paid_off_time değeri 01/10/2016'ya kadar olan bu müşteriler kredilerini en erken kapatan gruptur. Banka açısından değerlendirildiğinde; güvenilir ve düşük riskli müşterilerdir.

C1 Kümesi (Mavi Noktalar) — Orta Vadede Ödeyen Orta Riskli Müşteriler: Bu küme, Ekim 2016 ortasında ödeme yapan ve terms değeri 15-30 arasında olan örneklerden oluşmaktadır. paid_off_time değeri 01/10/2016 ile 26/10/2016 arasında olan bu müşteriler orta vadeli ödeme profili sergilemektedir. Banka açısından değerlendirildiğinde; orta riskli müşterilerdir.

C3 Kümesi (Yeşil Noktalar) — Geç Ödeyen Riskli Müşteriler: Bu küme, geç ödenen (Ekim 2016 sonu — Kasım 2016) ve terms değeri 15-30 arasında olan örneklerden oluşmaktadır. paid_off_time değeri 26/10/2016'dan sonra olan bu müşteriler kredilerini en geç kapatan gruptur. Banka açısından değerlendirildiğinde; daha yüksek risk taşıyan müşterilerdir.

4.6.3. Dördüncü problemin Python ile çözümü

Bu bölüm; borcunu ödemeyen ve ödemeyen müşteriler için Python programı ile K-Means kümeleme algoritması ve bu algoritmaya ek olarak Desicion Tree sınıflandırma algoritması çözümlerini ve sonuçlarını içermektedir.

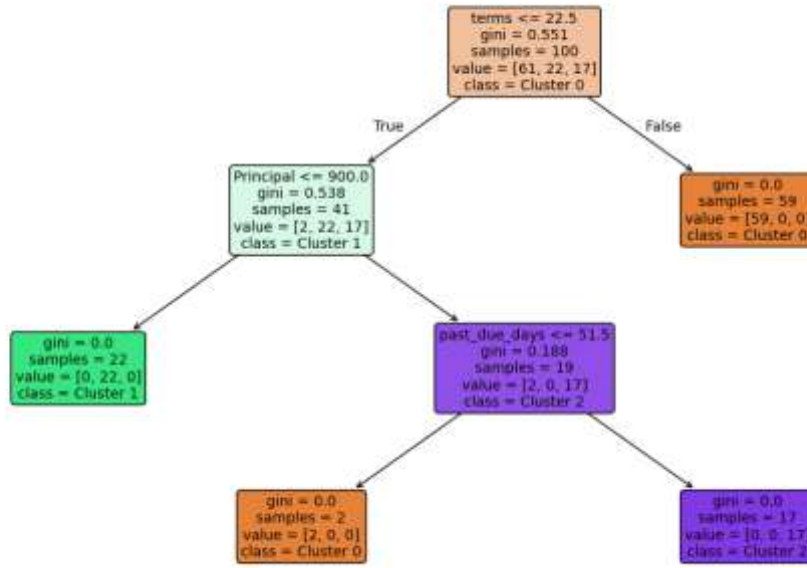
4.6.3.1. Borcunu ödemeyenler için Python ile K-Means algoritması çözümü

Borcunu ödemeyen müşteriler için Python programında, K-Means algoritması çalıştırılarak 3 segment oluşturulmuştur. Segment dağılımı şu şekildedir:

- C1: 61 Müşteri
- C2: 22 Müşteri
- C3: 17 Müşteri

Şekil 4.44'de ise dördüncü problem borcunu ödemeyen müşteriler için Python programı ile K-Means algoritması ağaç yapısı görülmektedir.

Borcunu Ödemeyenler İçin K-Means Clusterlarını Açıklayan Decision Tree



Şekil 4.44. Dördüncü problem borcunu ödemeyenler için Python ile K-Means algoritması ağaç yapısı

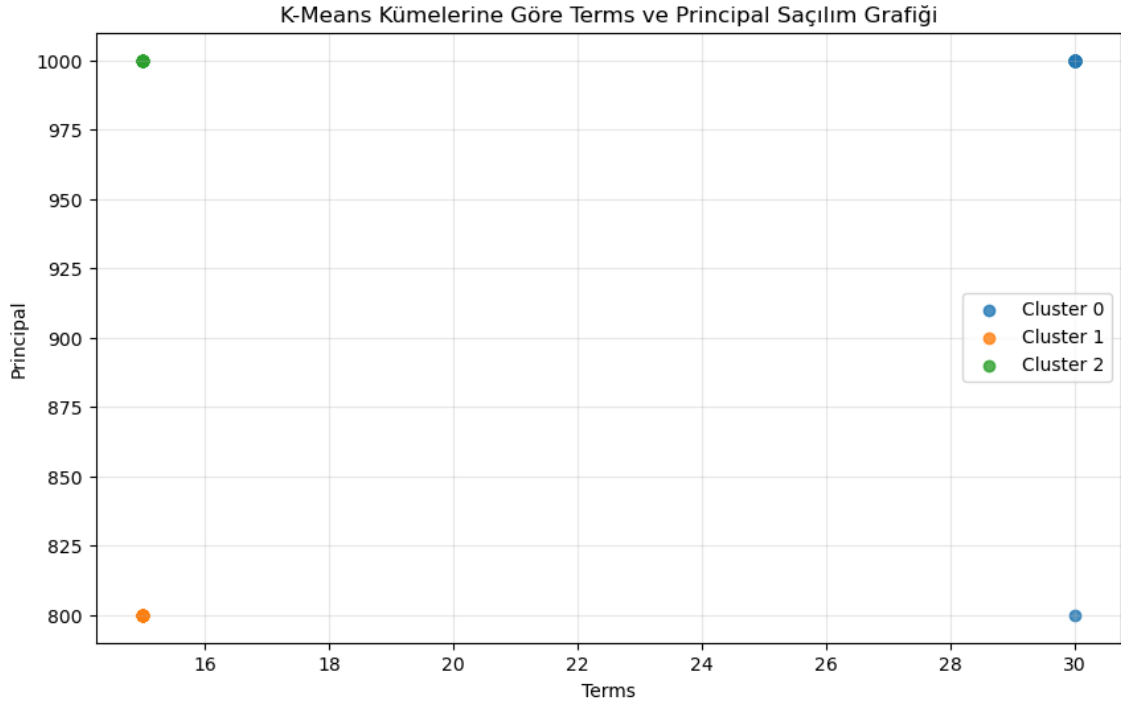
Karar ağacı yorumlandığında şu sonuçlar ortaya çıkmıştır:

Kredi Vadesi (terms): 22,5 günün altında olan krediler Cluster 1 ve Cluster 2'ye ayrılırken, 22,5 günün üzerindeki uzun vadeli kredilerin tamamı (59 örnek) Cluster 0'a atanmıştır.

Kredi Tutarı (Principal): $terms \leq 22,5$ koşulunu sağlayan 41 örnek içinde Principal değeri 900'ün altında olan 22 örneğin tamamı Cluster 1'e atanmıştır. Principal değeri 900'ün üzerinde olan 19 örnek ise gecikmiş gün sayısına göre daha fazla bölünmektedir.

Gecikmiş Gün Sayısı (past_due_days): $Principal > 900$ olan örneklerde past_due_days değeri 51,5'in altında olan 2 örnek Cluster 0'a, 51,5'in üzerinde olan 17 örnek ise Cluster 2'ye atanmıştır.

Şekil 4.45.'de ise dördüncü problem borcunu ödemeyen müşteriler için Python programı ile K-Means algoritması "terms" ve "principal" grafiği verilmiştir.



Şekil 4.45. Dördüncü problem borcunu ödemeyenler için Python ile K-Means algoritması "terms" ve "principal" grafiği

Bu grafikte, K-means algoritmasıyla oluşturulan kümeler terms (kredi vadesi) ve Principal (kredi tutarı) değişkenlerine göre görselleştirilmiştir.

Grafik ve karar ağacı birlikte değerlendirildiğinde, kümeler aşağıdaki gibi yorumlanarak adlandırılmıştır.

Cluster 0 (Mavi Noktalar) — Uzun Vadeli Yüksek Tutar Borçlular: Bu kümede yer alan müşteriler uzun vadeli (terms: 30 gün) ve yüksek tutarlı (Principal: 1000) kredilere sahiptir. $\text{terms} > 22,5$ koşulunu sağlayan bu müşterilerin tamamı Cluster 0'a atanmıştır. Uzun vadeli krediler almaları ve yüksek kredi tutarları nedeniyle daha büyük bir finansal yük altında olabilirler. Banka, bu müşteriler için borç yapılandırma programları sunarak geri ödeme sürecini kolaylaştırabilir.

Cluster 1 (Turuncu Noktalar) — Kısa Vadeli Düşük Tutar Borçlular: Bu kümede yer alan müşteriler kısa vadeli (terms: 15 gün) ve orta tutarlı (Principal: 800) kredilere sahiptir. $\text{terms} \leq 22,5$ ve $\text{Principal} \leq 900$ koşullarını sağlayan bu müşteriler, kısa vadede ödeme yapmakta zorlanıyor olabilir. Banka, bu müşterilere daha esnek ödeme seçenekleri sunabilir.

Cluster 2 (Yeşil Noktalar) — Kısa Vadeli Yüksek Tutar Yüksek Gecikmeli Borçlular:

Bu kümede yer alan müşteriler kısa vadeli (terms: 15 gün) ve yüksek tutarlı (Principal: 1000) kredilere sahip olup past_due_days değeri 51,5'in üzerindedir. Kısa vadede yüksek tutarlı kredileri olan bu müşterilerin ciddi gecikme yaşaması likidite sorunlarına işaret edebilir. Banka, bu grup için sıkı takip politikaları uygulamalıdır. Çizelge 4.44.'de dördüncü problem borcunu ödemeyen müşteriler için Python programı ile K-Means algoritması performans metrik değerleri verilmiştir.

Çizelge 4.44. Dördüncü problem borcunu ödemeyenler için Python ile K-Means algoritması performans metrikleri

Performans Metrikleri	Değerleri
Rand Index	1.00
Purity	1.00
Homogeneity	1.00
Completeness	1.00
V-Measure	1.00
Silhouette Score	0.81
Davies-Bouldin Index	0.29
Fowlkes-Mallows Index	1.00

Bu sonuçlar, kümelerin kendi içinde tutarlı olduğunu ve iyi bir şekilde ayrıldığını göstermektedir. Ancak, Silhouette Score gibi bazı metrikler, daha iyi bir ayrışmanın mümkün olabileceğini ima edebilir.

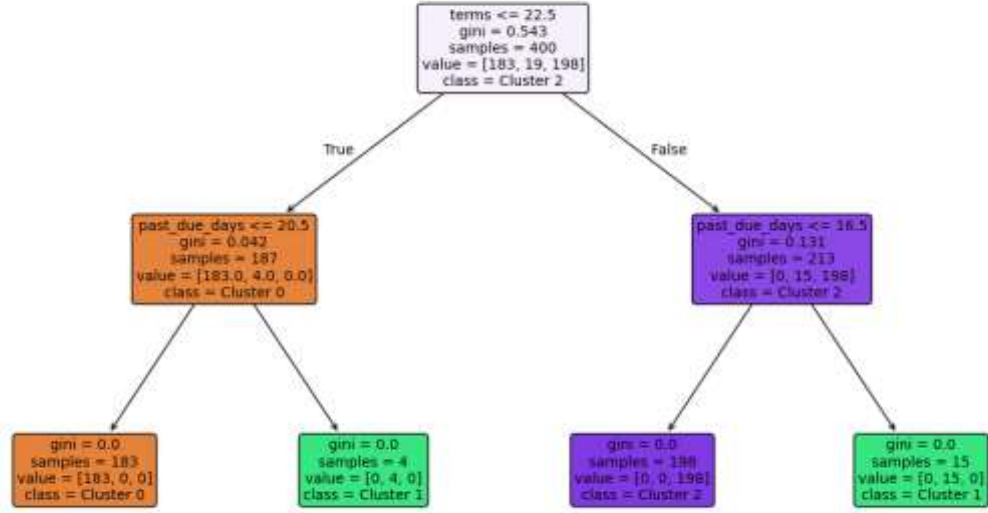
4.6.3.2. Borcunu ödeyenler için Python ile K-Means algoritması çözümü

Borcunu ödeyen müşteriler için Python programında, K-Means algoritması çalıştırılarak 3 segment oluşturulmuştur. Segment dağılımı şu şekildedir:

- Cluster 0: PAIDOFF 147 müşteri, COLLECTION_PAIDOFF 36 müşteri
- Cluster 1: PAIDOFF 0 müşteri, COLLECTION_PAIDOFF 19 müşteri
- Cluster 2: PAIDOFF 153 müşteri, COLLECTION_PAIDOFF 45 müşteri

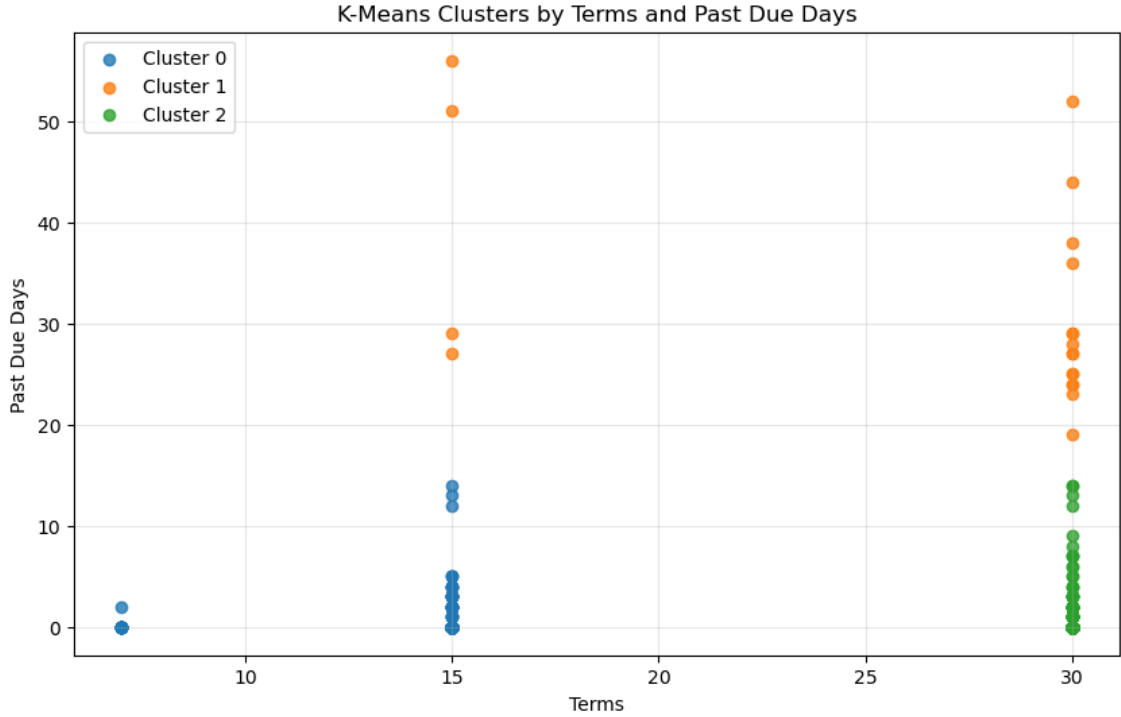
Şekil 4.46.'da ise dördüncü problem borcunu ödeyen müşteriler için Python programı ile K-Means algoritması ağaç yapısı görülmektedir.

Borcunu Ödeyenler İçin K-Means Clusterlarını Açıklayan Decision Tree



Şekil 4.46. Dördüncü problem borcunu ödeyenler için Python ile K-Means algoritması ağaç yapısı

Karar ağacı yorumlandığında; terms değeri 22,5'ten az olan ve past_due_days değeri 20,5'ten az olan 183 müşteri tamamen Cluster 0'da yer almaktadır. terms değeri 22,5'ten az olan ve past_due_days değeri 20,5'ten fazla olan 4 müşteri ise Cluster 1'e atanmaktadır. terms değeri 22,5'ten fazla olan ve past_due_days değeri 16,5'ten az olan 198 müşterinin tamamı Cluster 2'de yer almaktadır. terms değeri 22,5'ten fazla olan ve past_due_days değeri 16,5'ten fazla olan 15 müşteri ise Cluster 1'e atanmaktadır. Bu karar ağacı, müşteri segmentasyonunu belirli kriterlere göre otomatikleştirmeyi sağlamaktadır ve her segmentin belirli özelliklere sahip olduğunu göstermektedir. Şekil 4.47.'de ise dördüncü problem borcunu ödeyen müşteriler için Python programı ile K-Means algoritması "terms" ve "past_due_days" grafiği verilmiştir.



Şekil 4.47. Dördüncü problem borcunu ödeyenler için Python ile K-Means algoritması "terms" ve "past_due_days" grafiği

Yukarıdaki saçılım grafiğinde, terms ve past_due_days değişkenlerine göre K-Means kümeleme algoritmasıyla oluşturulan kümeler gösterilmiştir. Farklı renkler farklı kümeleri temsil etmektedir. Grafik ve karar ağacı birlikte değerlendirildiğinde, kümeler aşağıdaki gibi yorumlanarak adlandırılmıştır.

Cluster 0 (Mavi Noktalar) — Zamanında Ödeyenler: Bu küme, düşük past_due_days değerlerine (0-15 gün) sahip ve terms değeri 7-15 gün arasında olan müşterileri içermektedir. Karar ağacında $\text{terms} \leq 22,5$ ve $\text{past_due_days} \leq 20,5$ koşullarını sağlayan 183 müşteri bu kümeye atanmıştır. Bu müşteriler borçlarını nispeten zamanında ödeyen güvenilir müşterilerdir. Banka, bu müşterilere sadakat programları veya indirimli kredi teklifleri sunabilir.

Cluster 1 (Turuncu Noktalar) — Gecikenler: Bu küme, yüksek past_due_days değerlerine (20-56 gün) sahip müşterileri içermektedir. terms değeri hem 15 hem de 30 gün olan müşteriler bu kümede yer almaktadır. Karar ağacında $\text{terms} \leq 22,5$ ve $\text{past_due_days} > 20,5$ ile $\text{terms} > 22,5$ ve $\text{past_due_days} > 16,5$ koşullarını sağlayan toplam 19 müşteri bu kümeye atanmıştır. Bu müşteriler borçlarını ödemede gecikme

yaşayan yüksek riskli müşterilerdir. Banka, bu grup için borç yapılandırma ve sıkı takip politikaları uygulamalıdır.

Cluster 2 (Yeşil Noktalar) — Uzun Vadeli Zamanında Ödeyenler: Bu küme, düşük `past_due_days` değerlerine (0-13 gün) sahip ve `terms` değeri 30 gün olan müşterileri içermektedir. Karar ağacında $\text{terms} > 22,5$ ve $\text{past_due_days} \leq 16,5$ koşulunu sağlayan 198 müşteri bu kümeye atanmıştır. En büyük müşteri grubunu oluşturan bu küme, uzun vadeli kredilerini düzenli ödeyen güvenilir müşterilerden oluşmaktadır. Banka, bu müşterilere yönelik uzun vadeli kredi avantajları sunabilir. Çizelge 4.45.'de dördüncü problem borcunu ödeyen müşteriler için Python programı ile K-Means algoritması performans metrik değerleri verilmiştir.

Çizelge 4.45. Dördüncü problem borcunu ödeyenler için Python ile K-Means algoritması performans metrikleri

Performans Metrikleri	Değerleri
Rand Index	1.00
Purity	1.00
Homogeneity	1.00
Completeness	1.00
V-Measure	1.00
Silhouette Score	0.85
Davies-Bouldin Index	0.37
Fowlkes-Mallows Index	1.00

Bu metrikler, kümeleme algoritmasının borcunu ödeyen müşteriler üzerinde oldukça başarılı bir performans sergilediğini göstermektedir. Rand Index, Purity, Homogeneity, Completeness, V-Measure ve Fowlkes-Mallows Index değerlerinin tamamının 1,00 olması, kümeler ile gerçek etiketler arasında mükemmel bir uyum sağlandığını ortaya koymaktadır. Bu sonuç, her kümenin yalnızca tek bir sınıfa ait örnekler içerdiğini ve hiçbir yanlış sınıflandırmanın gerçekleşmediğini göstermektedir. Silhouette Skoru 0,85 ile kümelerin birbirinden net biçimde ayrıldığını ve her örneğin kendi kümesine iyi uyum sağladığını doğrulamaktadır. Davies-Bouldin İndeksi ise 0,37 ile düşük bir değer sergilemekte olup bu da kümeler arası mesafenin yüksek, küme içi dağılımın ise sıkı olduğunu göstermektedir. Genel olarak bu sonuçlar, K-Means algoritmasının borcunu ödeyen müşteri verisi üzerinde mükemmel bir kümeleme performansı elde ettiğine işaret etmektedir.

4.6.4. Dördüncü problemin R ile çözümü

Bu bölüm; borcunu ödemeyen ve ödemeyen müşteriler için R programı ile K-Means kümeleme algoritması ve bu algoritmaya ek olarak Desicion Tree sınıflandırma algoritması çözümlerini ve sonuçlarını içermektedir.

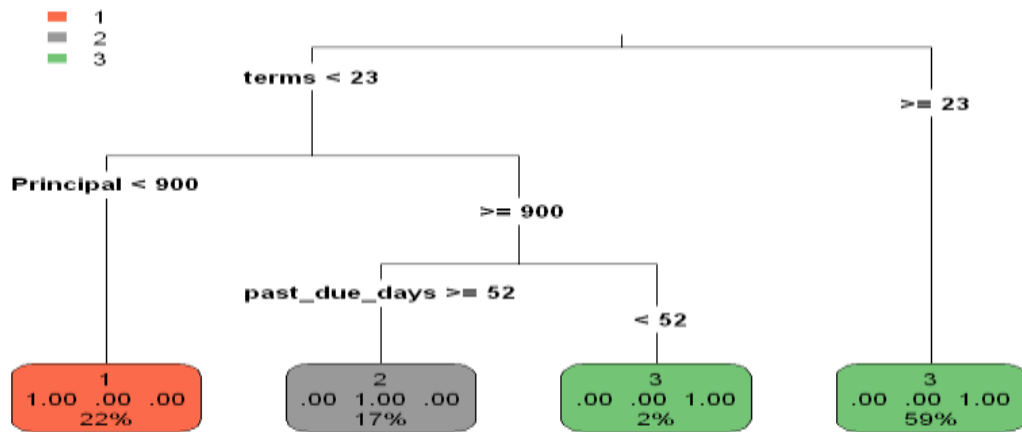
4.6.4.1. Borcunu ödemeyenler için R ile K-Means algoritması çözümü

Borcunu ödemeyen müşteriler için Python programında, K-Means algoritması çalıştırılarak 3 segment oluşturulmuştur. Segment dağılımı şu şekildedir:

- Cluster 1: 22 müşteri (terms: 15, Principal: 800, past_due_days: 73.6)
- Cluster 2: 17 müşteri (terms: 15, Principal: 1000, past_due_days: 72.6)
- Cluster 3: 61 müşteri (terms: 29.5, Principal: 997, past_due_days: 56.2)

Şekil 4.48’de ise dördüncü problem borcunu ödemeyen müşteriler için Python programı ile K-Means algoritması ağaç yapısı görülmektedir.

Borcunu Ödemeyenler İçin K-Means Clusterlarını Açıklayan Decision Tree



Şekil 4.48. Dördüncü problem borcunu ödemeyenler için R ile K-Means algoritması ağaç yapısı

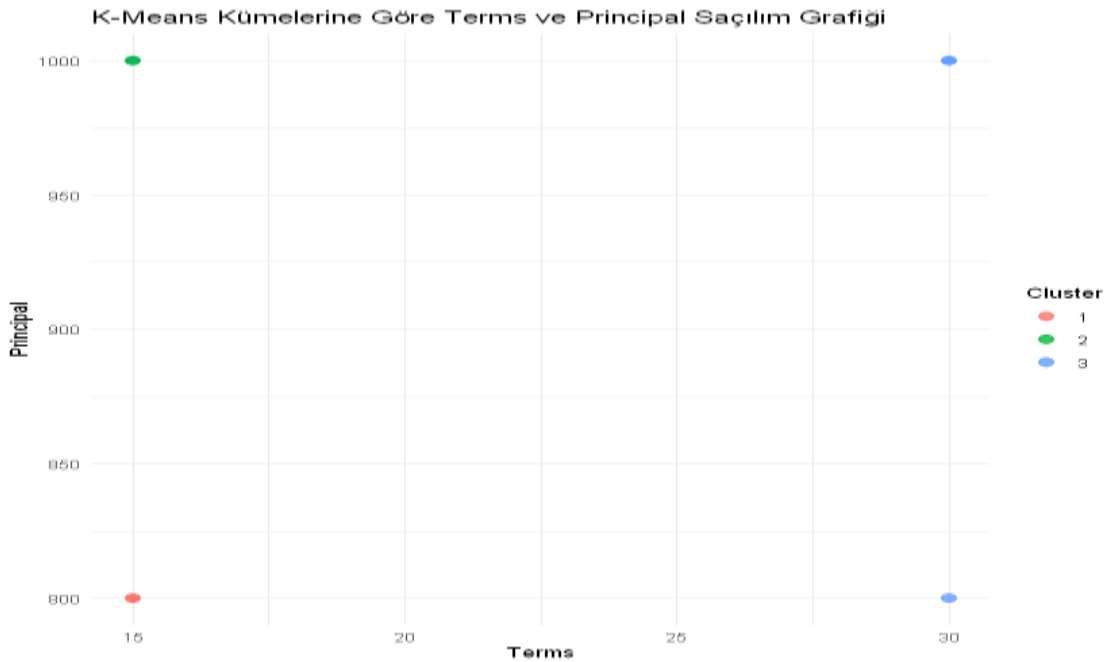
Karar ağacı yorumlandığında şu sonuçlar ortaya çıkmıştır:

Kredi Vadesi (terms): 23 günün altında olan krediler Cluster 1 ve Cluster 2'ye ayrılırken, 23 gün ve üzerindeki uzun vadeli kredilerin büyük çoğunluğu (%59) Cluster 3'e atanmıştır.

Kredi Tutarı (Principal): terms < 23 koşulunu sağlayan örneklerde Principal değeri 900'ün altında olan 22 müşterinin tamamı (%100) Cluster 1'e atanmıştır. Principal değeri 900 ve üzerinde olan örnekler ise gecikmiş gün sayısına göre daha fazla bölünmektedir.

Gecikmiş Gün Sayısı (past_due_days): Principal $\geq$ 900 olan örneklerde past_due_days değeri 52 ve üzerinde olan 17 müşterinin tamamı Cluster 2'ye, 52'nin altında olan 2 müşteri ise Cluster 3'e atanmıştır.

Şekil 4.49.'da ise dördüncü problem borcunu ödemeyen müşteriler için Python programı ile K-Means algoritması "terms" ve "principal" grafiği verilmiştir.



Şekil 4.49. Dördüncü problem borcunu ödemeyenler için R ile K-Means algoritması "terms" ve "principal" grafiği

Bu grafikte, K-Means algoritmasıyla oluşturulan kümeler terms (kredi vadesi) ve Principal (kredi tutarı) değişkenlerine göre görselleştirilmiştir. Grafik ve karar ağacı birlikte değerlendirildiğinde, kümeler aşağıdaki gibi yorumlanarak adlandırılmıştır.

Cluster 1 (Kırmızı Noktalar) — Kısa Vadeli Düşük Tutar Borçlular: Bu kümede yer alan müşteriler kısa vadeli (terms: ortalama 15 gün) ve düşük tutarlı (Principal: ortalama 800) kredilere sahiptir, gecikmiş gün sayısı ortalama 73,6'dır. Düşük kredi tutarlarına rağmen kısa vadede borçlarını ödeyememeleri likidite sorunları veya gelir yetersizliğine işaret edebilir. Banka, bu müşterilere daha esnek ödeme koşulları sunabilir.

Cluster 2 (Yeşil Noktalar) — Kısa Vadeli Yüksek Tutar Borçlular: Bu kümede yer alan müşteriler kısa vadeli (terms: ortalama 15 gün) ve yüksek tutarlı (Principal: ortalama 1000) kredilere sahiptir, gecikmiş gün sayısı ortalama 72,6'dır. Yüksek kredi tutarı ve kısa vade kombinasyonu bu müşterilerin finansal sıkıntı yaşadığına işaret edebilir. Banka, bu grup için ödeme planlarını gözden geçirerek daha esnek seçenekler sunabilir.

Cluster 3 (Mavi Noktalar) — Uzun Vadeli Yüksek Tutar Borçlular: Bu küme en büyük grubu oluşturmakta olup uzun vadeli (terms: ortalama 29,5 gün) ve yüksek tutarlı (Principal: ortalama 997) kredilere sahip müşterileri içermektedir, gecikmiş gün sayısı ortalama 56,2'dir. Uzun vadeli kredilerini ödeyememeleri gelir dengesizliği veya borç yönetiminde zorluk yaşandığını gösterebilir. Banka, bu müşteriler için borç yapılandırma programları sunabilir.

Çizelge 4.46.'da dördüncü problem borcunu ödemeyen müşteriler için R programı ile K-Means algoritması performans metrik değerleri verilmiştir.

Çizelge 4.46. Dördüncü problem borcunu ödemeyenler için R ile K-Means algoritması performans metrikleri

Performans Metrikleri	Değerleri
Rand Index	1.00
Purity	1.00
Homogeneity	1.00
Completeness	1.00
V-Measure	1.00
Silhouette Score	0.81
Davies-Bouldin Index	0.29
Fowlkes-Mallows Index	1.00

Bu sonuçlar, kümeleme algoritmasının borcunu ödemeyenler üzerinde mükemmel bir performans sergilediğini göstermektedir. Rand Index, Purity, Homogeneity, Completeness, V-Measure ve Fowlkes-Mallows Index değerlerinin tamamının 1,00

olması, kümeler ile gerçek etiketler arasında tam bir uyum sağlandığını ortaya koymaktadır. Silhouette Skoru 0,81 ile kümelerin birbirinden net biçimde ayrıldığını doğrulamaktadır. Davies-Bouldin İndeksi ise 0,29 ile oldukça düşük bir değer sergilemekte olup kümeler arası mesafenin yüksek, küme içi dağılımın ise sıkı olduğunu göstermektedir. Genel olarak bu sonuçlar, K-Means algoritmasının borcunu ödemeyenler için mükemmel bir kümeleme performansı elde ettiğine işaret etmektedir.

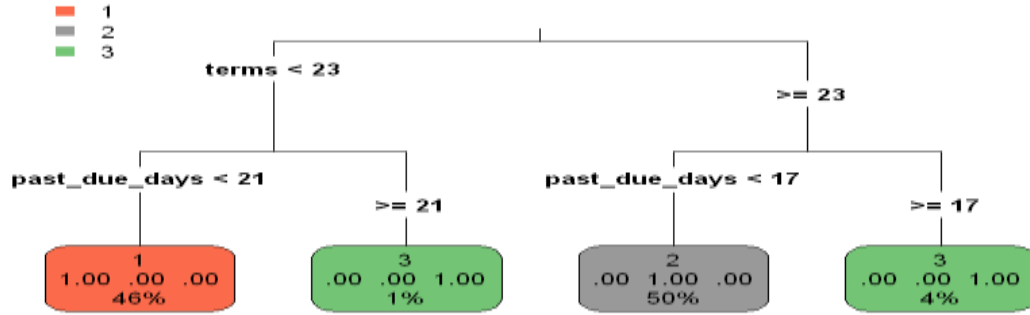
4.6.4.2. Borcunu ödeyenler için R ile K-Means algoritması çözümü

Borcunu ödeyen müşteriler için Python programında, K-Means algoritması çalıştırılarak 3 segment oluşturulmuştur. Segment dağılımı şu şekildedir:

- Cluster 1: 183 müşteri (terms: 14.1, past_due_days: 0.69) - 147 PAIDOFF, 36 COLLECTION_PAIDOFF
- Cluster 2: 198 müşteri (terms: 30, past_due_days: 0.89) - 153 PAIDOFF, 45 COLLECTION_PAIDOFF
- Cluster 3: 19 müşteri (terms: 26.8, past_due_days: 32.3) - 19 COLLECTION_PAIDOFF

Şekil 4.50.'de ise dördüncü problem borcunu ödeyen müşteriler için Python programı ile K-Means algoritması ağaç yapısı görülmektedir.

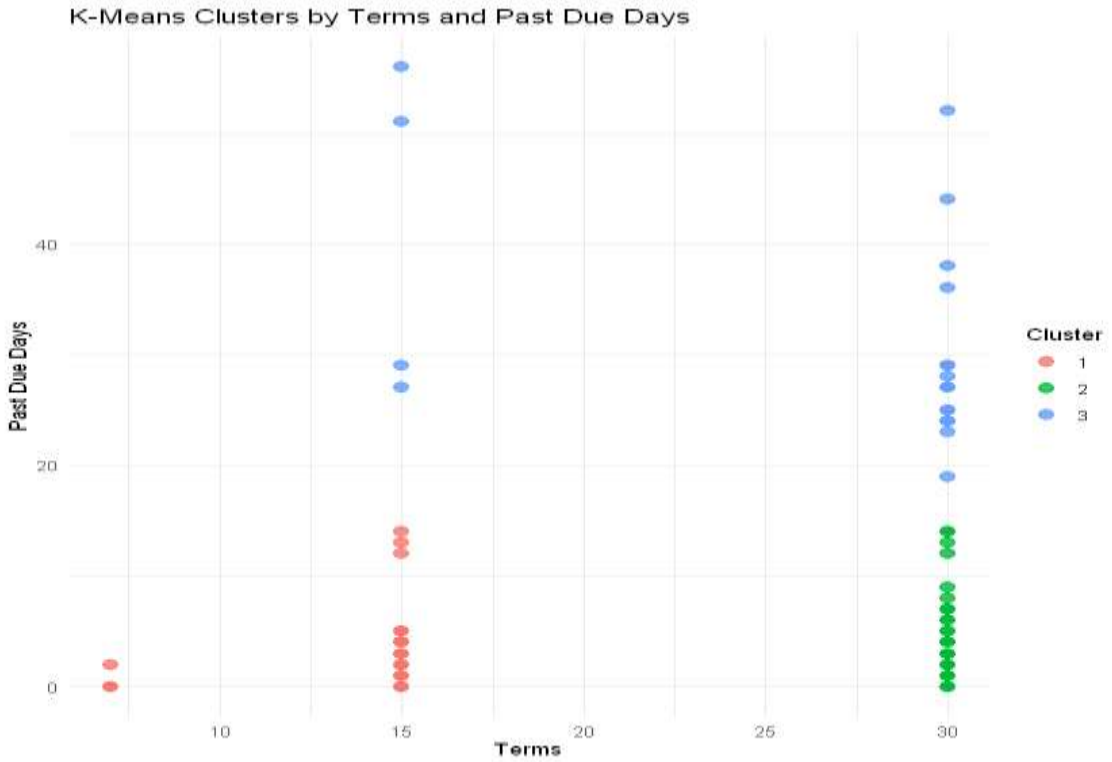
Borcunu Ödeyenler İçin K-Means Clusterlarını Açıklayan Decision Tree



Şekil 4.50. Dördüncü problem borcunu ödeyenler için R ile K-Means algoritması ağaç yapısı

Karar ağacı yorumlandığında; terms değeri 23'ten az olan ve past_due_days değeri 21'den az olan müşterilerin tamamı (%46) Cluster 1'de yer almaktadır. terms değeri 23'ten az olan ve past_due_days değeri 21 ve üzerinde olan az sayıdaki müşteri (%1) ise Cluster 3'e atanmaktadır. terms değeri 23 ve üzerinde olan ve past_due_days değeri 17'den az olan müşterilerin tamamı (%50) Cluster 2'de yer almaktadır. terms değeri 23 ve üzerinde olan ve past_due_days değeri 17 ve üzerinde olan müşteriler (%4) ise Cluster 3'e atanmaktadır.

Bu karar ağacı, müşteri segmentasyonunu belirli kriterlere göre otomatikleştirmeyi sağlamaktadır ve her segmentin belirli özelliklere sahip olduğunu göstermektedir. Şekil 4.51.'de ise dördüncü problem borcunu ödeyen müşteriler için Python programı ile K-Means algoritması "terms" ve "past_due_days" grafiği verilmiştir.



Şekil 4.51. Dördüncü problem borcunu ödeyenler için R ile K-Means algoritması "terms" ve "past_due_days" grafiği

Yukarıdaki saçılım grafiğinde, terms ve past_due_days değişkenlerine göre K-Means kümeleme algoritmasıyla oluşturulan kümeler gösterilmiştir. Farklı renkler farklı kümeleri temsil etmektedir. Grafik ve karar ağacı birlikte değerlendirildiğinde, kümeler aşağıdaki gibi yorumlanarak adlandırılmıştır.

Cluster 1 (Kırmızı Noktalar) — Zamanında Ödeyenler: Bu küme kısa vadeli (terms: ortalama 14,1 gün) ve düşük gecikme süreli (past_due_days: ortalama 0,69 gün) müşterileri içermektedir. 147 PAIDOFF ve 36 COLLECTION_PAIDOFF olmak üzere toplam 183 müşteriden oluşmaktadır. Bu kümedeki müşteriler borçlarını neredeyse hiç geciktirmeden ödeyen güvenilir müşterilerdir. Banka, bu müşterilere sadakat programları veya indirimli kredi teklifleri sunabilir.

Cluster 2 (Yeşil Noktalar) — Uzun Vadeli Zamanında Ödeyenler: Bu küme uzun vadeli (terms: ortalama 30 gün) ve düşük gecikme süreli (past_due_days: ortalama 0,89 gün) müşterileri içermektedir. 153 PAIDOFF ve 45 COLLECTION_PAIDOFF olmak üzere toplam 198 müşteriden oluşan en büyük segmenttir. Bu kümedeki müşteriler uzun vadeli kredilerini düzenli ödeyen güvenilir müşterilerdir. Banka, bu müşterilere uzun vadeli kredi avantajları sunabilir.

Cluster 3 (Mavi Noktalar) — Gecikenler: Bu küme orta-uzun vadeli (terms: ortalama 26,8 gün) ve yüksek gecikme süreli (past_due_days: ortalama 32,3 gün) müşterileri içermektedir. Tamamı 19 COLLECTION_PAIDOFF müşteriden oluşan bu küme yüksek risk grubunu temsil etmektedir. Banka, bu müşteriler için borç yapılandırma ve sıkı takip politikaları uygulamalıdır.

Çizelge 4.47.'de dördüncü problem borcunu ödeyen müşteriler için Python programı ile K-Means algoritması performans metrik değerleri verilmiştir.

Çizelge 4.47. Dördüncü problem borcunu ödeyenler için R ile K-Means algoritması performans metrikleri

Performans Metrikleri	Değerleri
Rand Index	1.00
Purity	1.00
Homogeneity	1.00
Completeness	1.00
V-Measure	1.00
Silhouette Score	0.85
Davies-Bouldin Index	0.37
Fowlkes-Mallows Index	1.00

Bu sonuçlar, kümeleme algoritmasının borcunu ödeyen müşteriler üzerinde mükemmel bir performans sergilediğini göstermektedir. Rand Index, Purity, Homogeneity, Completeness, V-Measure ve Fowlkes-Mallows Index değerlerinin tamamının 1,00 olması, kümeler ile gerçek etiketler arasında tam bir uyum sağlandığını ortaya koymaktadır. Silhouette Skoru 0,85 ile kümelerin birbirinden net biçimde ayrıldığını ve her örneğin kendi kümesine iyi uyum sağladığını doğrulamaktadır. Davies-Bouldin İndeksi ise 0,37 ile düşük bir değer sergilemekte olup kümeler arası mesafenin yüksek, küme içi dağılımın ise sıkı olduğunu göstermektedir. Genel olarak bu sonuçlar, K-Means algoritmasının borcunu ödeyen müşteriler için mükemmel bir kümeleme performansı elde ettiğine işaret etmektedir.

4.6.5. Dördüncü problem için en iyi program seçimi

Bu bölümde, dördüncü problem için çözülen algoritmaların ve bu algoritmaların çözüldüğü programların, borcunu ödeyen ve borcunun ödemeyen müşteriler olarak banka kredi müşteri için en iyisi iki başlık altında seçilmiş ve yorumlanmıştır.

4.6.5.1. Dördüncü problem borcunu ödemeyen müşteriler için programları karşılaştırma

Bu bölümde dördüncü problemde; Knime, Orange, Python ve R programlarında borcun ödemeyen müşteriler için çözümlenen en iyi algoritmalar karşılaştırılmıştır. Çizelge 4.48.'de ise dördüncü problem de Knime, Orange, Python ve R programlarında banka kredi müşteri verileri için kullanılan programların K-Means algoritmasının performans metrikleri verilmiştir.

Çizelge 4.48. Dördüncü problem borcunu ödemeyenler için programların performans metrikleri

Programlar	Python	Knime	Orange	R
Performans Metrikleri / Algoritma	K-Means	K-Means	K-Means	K-Means
Rand Index	1.00	1.00	1.00	1.00
Purity	1.00	0.77	0.57	1.00
Homogeneity	1.00	1.00	1.00	1.00
Completeness	1.00	1.00	1.00	1.00
V-Measure	1.00	1.00	1.00	1.00
Silhouette Score	0.81	0.94	-0.25	0.82
Davies-Bouldin Index	0.29	0.69	2.04	0.29
Fowlkes-Mallows Index	1.00	1.00	1.00	1.00

Bu metriklere göre en iyi performans gösteren programı belirlemek için her bir metriğin değerlendirilmesi gerekmektedir. Metriklerin yüksek olması daha iyi performans anlamına gelirken, Davies-Bouldin Index'in düşük olması daha iyi performans anlamına gelir.

Python ve R programları Rand Index, Purity, Homogeneity, Completeness, V-Measure ve Fowlkes-Mallows Index metriklerinin tamamında 1,00 değeriyle en yüksek performans sergilemiştir. Knime programı Purity metriğinde 0,77 ile daha düşük bir değer alırken, Orange programı Purity (0,57) ve Silhouette Score (-0,25) metriklerinde belirgin biçimde zayıf kalmıştır. Silhouette Score açısından en iyi performansı 0,9483 ile Knime göstermiş, bunu 0,82 ile R ve 0,81 ile Python takip etmiştir. Davies-Bouldin Index'te ise en düşük değeri 0,29 ile Python ve R paylaşmıştır.

Genel değerlendirmede Python ve R programları birbirine çok yakın performans sergilemiş olup ödemeyenler için en iyi kümeleme sonuçlarını elde eden programlar olarak öne çıkmaktadır. Bulunan sonuçlara göre banka kredi müşterileri için anlam şu şekilde yorumlanabilir: Müşteri Segmentasyonu: Python ve R kullanılarak yapılan

segmentasyon daha yüksek doğruluk ve homojenlik sağlamaktadır. Bu, kredi borcunu ödemeyen müşterilerin daha doğru sınıflandırılmasını mümkün kılmaktadır. Risk Yönetimi: Daha iyi segmentasyon, bankanın riskli müşteri gruplarını daha doğru tanımlamasına ve bu gruplar için uygun kredi politikaları geliştirmesine yardımcı olabilir. Kişiselleştirilmiş Hizmetler: Segmentasyon sayesinde bankalar farklı müşteri gruplarına kişiselleştirilmiş hizmetler sunabilir. Örneğin borçlarını ödemekte zorlanan müşterilere özel ödeme planları teklif edilebilir. Pazarlama Stratejileri: Segmentasyon, pazarlama stratejilerinin daha etkili planlanmasına olanak tanır. Hedeflenen müşteri gruplarına yönelik kampanyalar düzenlenebilir.

4.6.5.2. Dördüncü problem borcunu ödeyen müşteriler için programları karşılaştırma

Bu bölümde dördüncü problemde; Knime, R, Orange, Python programlarında borcunu ödeyen müşteriler için K-Means algoritmaları karşılaştırılmıştır. Müşteri segmentasyonu oluşturmak adına, borcunu ödeyen müşteriler için en iyi sonucu veren algoritma ve program bulunmuş ve yorumlanmıştır. Çizelge 4.49.'da dördüncü problem de Knime, R, Orange, Python programlarında banka kredi müşteri verileri içi kullanılan programların K-Means algoritmasının performans metrikleri verilmiştir.

Çizelge 4.49. Dördüncü problem borcunu ödeyenler için programların performans metrikleri

Programlar	Python	Knime	Orange	R
Performans Metrikleri / Algoritma	K-Means	K-Means	K-Means	K-Means
Rand Index	1.00	1.00	1.00	1.00
Purity	1.00	0.52	0.49	1.00
Homogeneity	1.00	1.00	1.00	1.00
Completeness	1.00	1.00	1.00	1.00
V-Measure	1.00	1.00	1.00	1.00
Silhouette Score	0.85	0.88	-0.06	0.85
Davies-Bouldin Index	0.37	0.32	6.0866	0.37
Fowlkes-Mallows Index	1.00	1.00	1.00	1.00

Performans metriklerine göre en iyi programı belirlemek için her bir metriği değerlendirmek gerekmektedir. Çizelge 4.50.'de ise dördüncü problem borcunu ödeyen müşteriler için tüm programların performans metrikleri karşılaştırması verilmiştir.

Çizelge 4.50. Dördüncü problem borcunu ödeyenler için programların performans metrikleri 2

Performans Metrikleri	En Yüksek Performans Gösteren Program
Rand Index	Python & R & Knime & Orange (1.00)
Purity	Python & R (1.00)
Homogeneity	Python & R & Knime & Orange (1.00)
Completeness	Python & R & Knime & Orange (1.00)
V-Measure	Python & R & Knime & Orange (1.00)
Silhouette Score	Knime (0.88)
Davies-Bouldin Index	Knime (0.32)
Fowlkes-Mallows Index	Python & R & Knime & Orange (1.00)

Python ve R programları Rand Index, Purity, Homogeneity, Completeness, V-Measure ve Fowlkes-Mallows Index metriklerinin tamamında 1,00 değeriyle en yüksek performansı sergilemiştir. Knime programı Purity metriğinde 0,52 ile daha düşük bir değer alırken, Orange programı Purity (0,49), Silhouette Score (-0,06) ve Davies-Bouldin Index (6,09) metriklerinde belirgin biçimde zayıf kalmıştır. Silhouette Score açısından en iyi performansı 0,88 ile Knime göstermiş, bunu 0,85 ile Python ve R takip etmiştir. Davies-Bouldin Index'te ise en düşük değeri 0,32 ile Knime elde etmiş, bunu 0,37 ile Python ve R izlemiştir.

Genel değerlendirmede Python ve R programları birbirine çok yakın performans sergilemiş olup borcunu ödeyen müşteriler için en iyi kümeleme sonuçlarını elde eden programlar olarak öne çıkmaktadır. Bulunan sonuçlara göre banka kredi müşterileri için anlam şu şekilde yorumlanabilir: Müşteri Segmentasyonu: Python ve R kullanılarak yapılan segmentasyon daha yüksek doğruluk ve homojenlik sağlamaktadır. Bu, kredi borcunu ödeyen müşterilerin daha doğru sınıflandırılmasını mümkün kılmaktadır. Risk Yönetimi: Kredi borcunu ödeyen müşterilerin segmentasyonu sayesinde bankalar düşük riskli müşteri gruplarını daha doğru tanımlayabilir ve bu gruplar için daha uygun kredi teklifleri sunabilir. Kişiselleştirilmiş Hizmetler: Segmentasyon sayesinde bankalar farklı müşteri gruplarına kişiselleştirilmiş hizmetler sunabilir. Örneğin zamanında ödeme yapan müşterilere özel avantajlar sunulabilir. Pazarlama Stratejileri: Daha etkili pazarlama stratejileri oluşturulabilir. Hedeflenen müşteri gruplarına yönelik kampanyalar düzenlenerek müşteri memnuniyeti ve bağlılığı artırılabilir.

5. TARTIŞMA VE SONUÇLAR

Bu tez çalışması, MİY’de VM yöntemlerini kullanarak bankacılık sektöründe dört temel problemi çözmeyi amaçlamaktadır. Bu problemler şunlardır:

1. Müşteri borcunu ödeyip ödemeyeceğini tahmin eden bir sistem geliştirme.
2. Müşterilerin ne kadar zamanda ödeme yapacağını (erken veya geç) tahmin eden bir sistem geliştirme.
3. Müşteriye verilebilecek krediyi tahmin eden bir sistem geliştirme.
4. Müşteri segmentasyonu oluşturma.

Çalışmada, KNIME, R, Orange ve Python programları kullanılmıştır. Problemler için çalıştırılan programlar ve algoritmalar Çizelge 5.1’de verilmiştir.

Çizelge 5.1. Problemlerde kullanılan program ve algoritmalar

Problemler ve Programlar	1.Problem	2. ve 3. Problem	4.Problem
KNIME	Decision Tree	MultiLayer	K-Means
	Naive Bayes	Perceptron Predictor	
	K Nearest Neighbor		
ORANGE	Decision Tree	Neural Network	K-Means
	Naive Bayes	K Nearest Neighbor	
	K Nearest Neighbor	Random Forest	
PYTHON	Decision Tree	Linear Regression	
	Naive Bayes	MultiLayer	K-Means
	K Nearest Neighbor	Perceptron Predictor	
R	XGBoost		
	Decision Tree	MultiLayer	K-Means
	Naive Bayes	Perceptron Predictor	
	K Nearest Neighbor		
	XGBoost		

Tabloda görüldüğü gibi; 5 programda 10 adet algoritma ile problemler çözülmeye çalışılmış ve problemlere göre en iyi çözüm sağlayan algoritma ve program banka veri seti düzeyinde değerlendirilmiştir. Performans metrikleri incelendiğinde problemlere göre sonuçlar şunlardır:

5.1. Birinci Problem İçin Sonuçlar ve Değerlendirme

1. Problem: Müşteri borcunu ödeyip ödemeyeceğini tahmin eden sistem geliştirme
Orange programı Desicion Tree algoritması yüksek doğruluk oranları ve düşük hata oranları ile en iyi performansı göstermiştir. Knime platformundaki Naive Bayes algoritması ise kabul edilebilir hata oranıyla bu model için güçlü bir alternatif teşkil etmektedir

5.2. İkinci ve Üçüncü Problemler İçin Sonuçlar ve Değerlendirme

2. ve 3. Problemler: Müşterilerin ne kadar zamanda ödeme yapacağını ve müşteriye verilebilecek krediyi tahmin eden sistemler geliştirme

Performans metrikleri incelendiğinde R programı ile geliştirilen MLP modelinin en düşük hata oranlarına ve en yüksek R^2 değerine sahip olduğu görülmektedir. Bu durum R ile kurulan modelin veriyi en iyi biçimde açıkladığını ve tahminlerde en yüksek doğruluğu sergilediğini ortaya koymaktadır. Müşterilerin ödeme sürelerini tahmin etmek ve verilebilecek krediyi belirlemek için R programı en iyi seçim olarak öne çıkmaktadır. Python ile geliştirilen MLP modeli ikinci sıraya yerleşmekte olup genel itibarıyla güçlü bir performans sergilemektedir. Knime ve Orange ise bazı metriklerde kabul edilebilir performans göstermekle birlikte R ve Python'un gerisinde kalmaktadır.

5.3. Dördüncü Problem İçin Sonuçlar ve Değerlendirme

Dördüncü Problem: Müşteri Segmentasyonu Oluşturma

Borcunu ödemeyen müşteriler için; Python ve R programlarında uygulanan K-Means algoritmaları en yüksek performansı sergilemiştir. Rand Index, Purity, Homogeneity, Completeness, V-Measure ve Fowlkes-Mallows Index metriklerinin tamamında 1,00 değerine ulaşan bu iki program ödemeyenler için en iyi kümeleme sonuçlarını elde etmiştir. Silhouette Score açısından ise 0,9483 ile Knime öne çıkmaktadır. Orange programı Purity (0,57) ve Silhouette Score (-0,25) metriklerinde belirgin biçimde zayıf kalmıştır.

Borcunu ödeyen müşteriler için; Python ve R programlarında uygulanan K-Means algoritmaları yine en yüksek genel performansı sergilemiştir. Purity, Homogeneity, Completeness, V-Measure metriklerinde 1,00 değerine ulaşılmıştır. Silhouette Score ve Davies-Bouldin Index açısından ise sırasıyla 0,88 ve 0,32 değerleriyle Knime en iyi ayrışmayı sağlamıştır. Orange programı Silhouette Score (-0,06) ve Davies-Bouldin Index (6,09) metriklerinde yetersiz kalmıştır.

Genel değerlendirmede her iki grup için de Python ve R programları en tutarlı ve yüksek performanslı sonuçları üreten programlar olarak öne çıkmaktadır. Bu sonuçlar bankanın müşteri segmentasyonu, risk yönetimi ve kişiselleştirilmiş hizmet stratejilerini geliştirmesine önemli katkı sağlayabilir.

6. KAYNAKÇA

- Şenyıl, H. (2024). *Müşteri ilişkileri yönetiminde veri madenciliği: Bir bankanın kredi müşterilerine yönelik uygulama*. Konya Gıda ve Tarım Üniversitesi.
- Acuner, Ş. (2005, Mayıs-Haziran). Günümüzde pazarlama anlayışında MİY Uygulamalarının Yeri ve Önemi. *Pazarlama Dünyası*(3), 52.
- Aggarwal, C. C. (2015). *Data Mining: The Textbook*. Springer.
- Agrawal, R., & Srikant, R. (1994). Fast Algorithms for Mining Association Rules. *Proc. of the 20th Int'l Conference on Very Large Databases*. Publication History.
- Agrawal, R., Imieli'nski, T., & Swami, A. (1993). Mining Association Rules Between Sets of Items in Large Databases. *ACM SIGMOD Record*, (s. 207-216). Washington, D.C.
- Agresti, A. (2018). *An Introduction to Categorical Data Analysis* (3 b.). Wiley.
- Akpınar, H. (2000). Veri Tabanlarında Bilgi Keşfi ve Veri Madenciliği. *İstanbul Üniversitesi İşletme Fakültesi Dergisi*, 29(1), 1-22.
- Alagöz, S. (2003). Bilişim Teknolojilerinin Müşteri İlişkileri Yönetimi (CRM) Üzerine Etkileri Ve Bankacılık Sektöründe Bir Uygulama. Selçuk Üniversitesi, Sosyal Bilimler Enstitüsü, İşletme Anabilim Dalı, Üretim Yönetimi Ve Pazarlama Bilimi Dalı.
- Alagöz, S., Alagöz, M., İnce, M., & Oktay, E. (2004). Müşteri İlişkileri Yönetimi (CRM) Bilişim Teknolojilerinin Etkisi ve Bankacılık Sektöründeki Gelişimi. Nobel Yayınları.
- Alavi, S., Ahuja, V., & Medury, Y. (2012). Metcalfe's Law and Operation, Analytical, and Collaborative CRM - Using Online Business Communities for Co-creation. *Journal of Targeting, Measurement and Analysis for Marketing*, 20(1), 35-45.
- Alavi, S., Ahuja, V., & Medury, Y. (2012). Metcalfe's Law and Operation, Analytical, and Collaborative CRM - Using Online Business Communities for Co-creation. *Journal of Targeting, Measurement and Analysis for Marketin*, 20(1).
- Albayrak, A. S. (2009). Classification of Domestic and Foreign Commercial Banks in Turkey Based On Financial Efficiency: A Comparison of Decision Tree, Logistic Regression and Discriminant Analysis Models. *S.D.Ü. İktisadi ve İdari Bilimler Fakültesi Dergisi*, 14(2), 113-139.
- Altunkaya, H. İ. (2013). Ülkelerin Uzun Dönem Kredi Notlarının Derecelendirilmesinde Önemli Değişkenlerin Veri Madenciliği Teknikleri Kullanılarak Belirlenmesi. Yüksek Lisans Tezi . Ankara: Hacettepe Üniversitesi Fen Bilimleri Enstitüsü.

- Aşlıyan, R., & Günel, K. (2010). Metin İçerikli Türkçe Dokümanların Sınıflandırılması. XII. Akademik Bilişim Konferansı (s. 529-534). Aydın: Adnan Menderes Üniversitesi, Matematik Bölümü.
- Babu, R., & Bhuvaneswari, K. (2021). Enhancing CRM Efficiency with Data Mining Techniques. *Information Systems Journal*.
- Bahçe Çizer, E. (2018). Yüksek Lisans Tezi Elektrik-Elektronik Mühendisliği Anabilim Dalı Elektrik-Elektronik Mühendisliği Programı. Yüksek Lisans Tezi. İstanbul: Marmara Üniversitesi Fen Bilimleri Enstitüsü.
- Berry, L. L. (1995). Relationship Marketing of Services—Growing Interest, Emerging Perspectives. 23(4), 236-245. *Journal of the Academy of Marketing Science*.
- Berthold, M., Cebron, N., Dill, F., Gabriel, T. R., Kötter, T., Meinl, T., . . . Wiswedel, B. (2009). KNIME-the Konstanz Information Miner: version 2.0 and beyond. *ACM SIGKDD Explorations Newsletter*, 11(1), 26-31.
- Bhambri, V. (2011). Application of Data Mining in Banking Sector. *International Journal of Computer Science and Technology*, 2(2), 199-202.
- Bigus , J. P. (1996). *Data Mining With Neural Networks: Solving Business Problems from Application Development to Decision Support*. McGraw-Hill.
- Boginski, V., & Butensko, M. (2017). Uncovering Structural Features of Stock Markets with Cross-Correlation and Clustering. *Journal of Financial Markets*.
- Bolton, R. J., & Hand, D. J. (2002). Statistical fraud detection: A review. *Statistical Science*, 17(3), 235-255.
- Bozkurt Coşkun, S. (2023). Ağaç Temelli Makine Öğrenmesi Yöntemleri İle Kredi Risk Analizi. İstanbul: İstanbul Ticaret Üniversitesi Fen Bilimleri Enstitüsü.
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5-32.
- Bulut, F. (2019). Bankacılık Sektöründe Yeni Özniteliklerle ve Makine Öğrenmesi Yöntemleriyle Müşteri İlişkileri Yönetiminin Zenginleştirilmesi. *Avrupa Bilim ve Teknoloji Dergisi*, 16, 382-394.
- Buttle , F., & Maklan, S. (2019). *Customer Relationship Management: Concepts and Technologies* (4 b.). Routledge.
- Buttle, F. (2009). *Customer Relationship Management: Concepts and Technologies*. Routledge.
- Buttle, F., & Maklan, S. (2015). *Customer Relationship Management: Concepts and Technologies*. Routledge.

- Buttle, F., & Maklan, S. (2019). Customer Relationship Management: Concepts and Technologies (4 b.). Routledge.
- Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. ACM Computing Surveys (CSUR), 41(3), 1-58.
- Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Rüdiger, W. (2000). SPSS CRISP-DM 1.0. U.S.A.: Spss Inc.
- Chaudhuri, S., & Dayal, U. (1997). An Overview of Data Warehousing and OLAP Technology. ACM SIGMOD Record, 26(1), 65-74.
- Chen, I., & Popovich, K. (2003). Understanding Customer Relationship Management (CRM): People, Process and Technology. Business Process Management Journal, 9(5), 672-688.
- Chen, J. (2015). Creating an Order Management System for Distribution Centers with Clustering Techniques. Journal of Operations Management.
- Chitra, K., & Subashini, B. (2013). Data Mining Techniques and Its Applications in Banking Sector. International Journal of Emerging Technology and Advanced Engineering, 3(8), 219-226.
- Chorianopoulos, A., & Tsiptsis, K. K. (2009). Data Mining Techniques in CRM: Inside Customer Segmentation (1 b.). Willey
- Chu, W., & Lin, T. (2005). Foundations and Advances in Data Mining. Springer .
- Çalış , A. (2013). Veri Madenciliği Yaklaşımı İle Bireysel Müşterilerin Kredi Ödeme Performanslarının Değerlendirilmesi. Yüksek Lisans Tezi. Kocaeli: Kocaeli Üniversitesi Fen Bilimleri Enstitüsü.
- Çalış, A., & Baynal, K. (2016). Kümeleme Analizi ile Bankacılık Sektöründe Satış Stratejilerinin Belirlenmesi. Beykent Üniversitesi Fen ve Mühendislik Bilimleri Dergisi, 9(1), 13-41.
- Çalışkan, S. K., & Soğukpınar, İ. (2008). KxKNN: K-means ve K en yakın komşu yöntemleri ile ağlarda nüfuz tespiti. 2. Ağ ve Bilgi Güvenliği Sempozyumu. Girne.
- Çığışar , B. (2017). Kredi Risklerinde Veri Madenciliği Sınıflandırma Algoritmaları. Yüksek Lisans Tezi. Adana: Çukurova Üniversitesi Fen Bilimleri Enstitüsü.
- Çıra, E. (2022). Bankacılık Sektöründe Makine Öğrenmesi Yöntemleri İle Müşteri Segmentasyonu Ve Karlılık Tahmini. İstanbul: Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü .
- Dasu, T., & Johnson, T. (2003). Exploratory Data Mining and Data Cleaning. John Wiley & Sons.

- Demsar, J., Curk, T., Erjavec, A., Gorup, C., Hocevar, T., Milutinovic, M., . . . Zupan, B. (2013). Orange: data mining toolbox in Python. *Journal of Machine Learning Research*, 14, 2349-2353.
- Doe, J., & Smith, A. (2022). Knowledge Discovery and Analytical Approaches. *Big Data and Analytics* (s. 301-320). içinde Springer.
- Doe, J., & Smith, A. (2023). *Data Analytics and Knowledge Discovery: An Overview*. Academic Press.
- Draper, N. R., & Smith, H. (1998). *Applied Regression Analysis*. John Wiley & Sons.
- Duran, M. (2006). MİY: Çok Konuşulan Ama Az Bilinen Bir Kavram.
- Dyche, J. (2002). *The MIY Handbook: A Business Guide to Customer Relationship Management*. Addison-Wesley.
- Dyche, J. (2002). *The CRM Handbook: A Business Guide to Customer Relationship Management*. Addison-Wesley Professional.
- Edelman, D., Crook, J., & Thomas, L. (2017). *Credit Scoring and Its Applications* (2 b.). SIAM-Society for Industrial & Applied Mathematics.
- Ekim, S. (2010). *Banka Etkinliği Ve Finansal Performans İlişkisi: Türk Bankacılık Sektöründe Veri Madenciliği Uygulaması*. Yüksek Lisans Tezi. Gaziantep: Gaziantep Üniversitesi Sosyal Bilimler Enstitüsü.
- Elsevier, P. (2021). Knowledge Discovery and Business Intelligence. *Elsevier Journal of Data Science*, 29(1), 98-115.
- Emel, G. G., & Taşkın, Ç. (2005). Veri Madenciliğinde Karar Ağaçları ve Bir Satış Analizi Uygulaması. *Eskişehir Osmangazi Üniversitesi Sosyal Bilimler Dergisi*, 6(2), 221 - 239.
- Ester, M., Kriegel, H., Sander, J., & Xu, X. (1996). A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, 226-231.
- Everitt, B., Landau, S., & Leese, M. (2001). *Cluster Analysis*. Arnold, 4th ed.
- Eyüboğlu, S. (2001). *Müşteri İlişkileri Yönetimi ve Veri Madenciliği Uygulamaları*. Detay Yayıncılık.
- Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From Data Mining to Knowledge Discovery in Databases. *AI Magazine*, 17(3), 37.
- Ferrell, O., Hartline, M., & Hochstein, B. W. (2021). *Marketing Strategy* (8 b.). Cengage Learning.
- Few, S. (2006). *Information Dashboard Design* (1 b.). O'Reilly Media.

- Fürnkranz , J., Gamberger , D., & Lavrač, N. (2012). Foundations of Rule Learning. Springer.
- Gartner, J. (2022). Business Intelligence and Analytics. Gartner Research, 28(4), 110-130.
- Giudici, P. (2000). Applied Data Mining Practical Machine Learning Tools and Techniques with Java Implementations. San Fransisco: Morgan Kaufmann Publishers.
- Goldenberg, B. (2008). CRM in Real Time: Empowering Customer Relationships. Information Today, Inc.
- Goldenberg, B. (2008). CRM in Real Time: Empowering Customer Relationships. CyberAge Books.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.
- Gordon, A. (1999). Classification, Chapman and Hall. 2nd ed.
- Greenberg , P. (2009). CRM at the Speed of Light: Social CRM Strategies, Tools, and Techniques for Engaging Your Customers (4 b.). McGraw-Hill.
- Grönroos, C. (1994). From Marketing Mix to Relationship Marketing: Towards a Paradigm Shift in Marketing. 32(2), 4-20. Management Decision.
- Guazzelli , A., Lin, W.-C., & Jena , T. (2010). PMML in Action: Unleashing the Power of Open Standards for Data Mining and Predictive Analytics. CreateSpace Independent Publishing Platform.
- Gulliver , S., Joshi, U., & Michell, V. (2013). Adapted customer relationship management implementation framework: facilitating value creation in nursing homes. Total Quality Management and Business Excellence, 24(9-10), 991-1003.
- Gülsoy , N. (2019). Kredi Skorumlama Süreçlerinde Veri Madenciliği Ve Bankacılık Sektöründe Bir Uygulama Hazırlayan. Yüksek Lisans Tezi. Kayseri: Erciyes Üniversitesi Fen Bilimleri Enstitüsü.
- Gültekin , N. (2021). Veri Madenciliği Teknikleri İle Kredi Hedef Kitlesi Stratejilerinin Belirlenmesi. İstanbul: Bahçeşehir Üniversitesi Lisansüstü Eğitim Enstitüsü.
- Güvençli, M. F. (2024). Kredi Taksit Ödemesini Aksatmış Müşterilerden Kötü Niyetli Olmayanların Makine Öğrenmesi Yöntemleri İle Tespiti. Mersin: Mersin Üniversitesi Fen Bilimleri Enstitüsü.
- Halkidi, M., Batistakis, Y., & Vazirgiannis, M. (2001). On Clustering Validation Techniques. Journal of Intelligent Information Systems, 17(2-3), 107-145.
- Han , J., Kamber, M., & Pei, J. (2011). Data Mining: Concepts and Techniques. Morgan Kaufmann, 3rd ed.

- Hand, D., & Yu, K. (2007). Idiot's Bayes: Not So Stupid after All? *International Statistical Review*, 69(3), 385 - 398.
- Harvard Business Review. (2019). Organizational Agility and Data Utilization. *Harvard Business Review*, 97(3), 45-58.
- Hassani, H., Huang, X., & Silva, E. (2018). Digitalisation and Big Data Mining in Banking. *Big Data and Cognitive Computing*, 2(18).
- Hosmer Jr., D., Lemeshow, S., & Sturdivant, R. X. (2013). *Applied Logistic Regression* (3 b.). Wiley.
- Hsieh, N. C. (2012). Creating a Behavioral Scoring Model for Credit Card Customers of a Bank Using Neural Networks and Association Rules. *Journal of Financial Services*.
- Hung, S. Y., & Wang, Y. (2014). Developing a Customer Profitability Management System for a Telecommunications Company with Data Mining Techniques. *Telecommunications Management Journal*.
- Hwang, H., Jung, T., & Suh, E. (2011). Developing a Model to Analyze Customer Lifespan with Logistic Regression, Neural Networks, and Decision Tree Techniques. *Customer Relationship Management Journal*.
- Inmon, W. H. (2000). *Building the Unstructured Data Warehouse*. Technics Publications.
- İşbilir, H. (2023). Kredi Tahsis Sürecinde Makine Öğrenmesi Yöntemlerinin Tahmin Performanslarının Karşılaştırılmasına Yönelik Bir Uygulama. Ankara: Çankaya Üniversitesi Fen Bilimleri Enstitüsü.
- Jain, A. (2010). Data Clustering: 50 Years Beyond K-Means. *Pattern Recognition Letters*, 31(8), 651-666.
- Jain, A., & Dubes, R. (1988). *Algorithms for Clustering Data*, Prentice Hall. Englewood Cliffs.
- Jain, A., Murty, M., & Flynn, P. (1999). Data Clustering: A Review. *ACM Computing Surveys*, 31(3), 264-323.
- Johnson, L., Lee, M., & Kim, S. (2023). *Living in the Data Age*. Information Age Publishing.
- Kallkota, R. (2001). *e-Business 2.0: Roadmap for Success*. Addison-Wesley.
- Kallkota, R., Robinson, M., & Tapscott, D. (2000). *e-Business 2.0: Roadmap for Success* (2 b.). Addison-Wesley Professional.
- Kantardzic, M. (2011). *Data Mining Concepts Models, Methods and Algorithms* (2 b.). Wiley-IEEE Press.

- Kaufman, L., & Rousseeuw, P. (1990). *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley.
- Kaufmann, M. (2020). *Data Mining and Knowledge Discovery*. Morgan Kaufmann Publishers.
- Khajvand, M., & Tarokh, M. J. (2019). Segmenting Valuable Customers with Clustering Algorithms. *Journal of Business Research*.
- Kotler, P., & Keller, K. L. (2015). *Marketing Management* (1 b.). Pearson Education.
- Kotler, P. (2009). *Marketing Management*. Pearson Education.
- Kotler, P. (2016). *Marketing Management* (15 b.). Pearson Education.
- Kotler, P., Keller, K. L., & Chern, A. (2021). *Marketing Management* (16 b.). Pearson Education Limited.
- Kotsiantis, S., Kanellopoulos, D., & Pintelas, P. (2006). Data Preprocessing for Supervised Learning. *International Journal of Computer Science*, 1(1), 1306-4428.
- Kumar, V., & Reinartz, W. (2018). *Customer Relationship Management: Concept, Strategy, and Tools* (3 b.). Springer.
- Kumar, V., Steinbach, M., & Tan, P. (2005). *Introduction to Data Mining*. Addison Wesley.
- Kutner, M., Nachtsheim, C., & Neter, J. (2004). *Applied Linear Regression Models* (4 b.). McGraw Hill.
- Kwon, H., & Park, J. (2020). *Databases and Big Data*. TechBooks.
- Last, M., & Kandel, A. (2013). Implementing Production Planning Applications in the Semiconductor Industry with Decision Tree Algorithms. *Semiconductor Review*.
- Lazarov, V., & Capota, M. (2007). *Churn Prediction*. Business, Computer Science.
- Levin, N., & Zahavi, J. (2001). Predictive Modeling Using Segmentation. *Journal of Interactive Marketing*, 15(2), 2-22.
- Lindgreen, A., & Antioco, M. (2005). Customer Relationship Management: The Case of a European Bank. *Journal of the Academy of Marketing Science*, 33(3), 273-291.
- Linoff, G. S., & Berry, M. J. (2011). *Data Mining Techniques for Marketing, Sales, and Customer Relationship Management* (3 b.). Wiley.
- Linoff, G., & Berry, M. J. (1999). *Mastering Data Mining: The Art and Science of Customer Relationship Management* (1 b.). John Wiley & Sons.
- Liu, B., Shue, L., & Lee, C. (2016). Enhancing Customer Relationships in the IPS Industry with Attribute-Oriented Induction. *Telecommunications Review*.

- Liu, C., & Shih, Y. (2015). Developing Customer Segmentation Models with AHP and Association Rule Techniques. *Decision Support Systems Journal*.
- Long, J. S. (1997). *Regression Models for Categorical and Limited Dependent Variables* (1 b.). Sage Publications.
- MacQueen, J. (1967). Some Methods for Classification and Analysis of Multivariate Observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 281-297.
- McKinney, W. (2010). Data Structures for Statistical Computing in Python. *Python in Science Conference*, (s. 51-56).
- Menard, S. W. (2002). *Applied Logistic Regression Analysis* (2 b.). SAGE Publications.
- Metsis, V., Androutsopoulos, I., & Paliouras, G. (2006). Spam filtering with naive Bayes-which naive Bayes? *CEAS*, 17, 28-69.
- Miyan, M. (2017). Applications of Data Mining in Banking Sector. *International Journal of Advanced Research in Computer Science*, 8(1), 108-114.
- Moin, K. I., & Ahmed, Q. B. (2012). Use of Data Mining in Banking. *International Journal of Engineering Research and Applications*, 2(2), 738-742.
- Montgomery, D. C., & Runger, G. C. (2018). *Applied Statistics and Probability for Engineers* (Cilt 7). Wiley.
- Montgomery, D. C., Peck, E. A., & Vining, G. G. (2012). *Introduction to Linear Regression Analysis* (5 b.). John Wiley & Sons Inc.
- Mosadegh, M. J., & Behboudi, M. (2011). Using Social Network Paradigm for Developing a Conceptual Framework in CRM. *Australian Journal of Business and Management Research*, 1(4).
- Ngai, E., Xiu, L., & Chau, D. (2009). Application of Data Mining Techniques in Customer Relationship Management: A Literature Review and Classification. *Expert Systems with Applications*, 36(2), 2592-2602.
- Nguyen, B., & Mutum, D. S. (2024). Applications of Data Mining in CRM. *Journal of Marketing Analytics*.
- Odabaş, Y. (2005). *Satışta ve Pazarlamada MİY* (3 b.). Sistem Yayıncılık.
- Odabaşı, Y. (2015). *Satışta ve Pazarlamada Müşteri İlişkileri Yönetimi*. Aura Kitapları.
- Oğuzlar, A. (2004). CART analizi ile hanehalkı işgücü anketi sonuçlarının özetlenmesi. *Atatürk Üniversitesi İktisadi ve İdari Bilimler Dergisi*, 18(3-4), 79-90.
- Okur, H. (2020). *Makine Öğrenmesi Yöntemiyle Kredi Risk Tahmini*. Ankara: Gazi Üniversitesi Fen Bilimleri Enstitüsü.

- O'Reilly, T. (2021). *Data Management in the Digital World*. O'Reilly.
- Ozgener, S., & İraz, R. (2006). Customer Relationship Management in Small-Medium Enterprises: The case of Turkish Tourism Industry. *Tourism Management*, 27(6), 1356-1363.
- Özdamar, D. (2004). *Bankacılık Sektöründe Müşteri İlişkileri Yönetimi ve Bir Uygulama*. Gazi Üniversitesi, Sosyal Bilimler Enstitüsü, İşletme Anabilim Dalı, Üretim Yönetimi ve Pazarlama Bilim dalı.
- Özekes, S. (2003). Veri Madenciliği Modelleri ve Uygulama Alanları. *İstanbul Ticaret Üniversitesi Dergisi*, 2(3), 65-82.
- Park, H., & Jun, C. (2009). A Simple and Fast Algorithm for K-medoids Clustering. *Expert Systems with Applications*, 36(2), 3336-3341.
- Parker, R., & Lee, C. (2023). *An Examination of Data Analytics and Knowledge Discovery*. Academic Press.
- Parvatiyar , A., & Sheth, J. (2001). Customer Relationship Management: Emerging Practice, Process, and Discipline. *Journal of Economic and Social Research*, 3(2), 1-34.
- Payne, A. (2005). *Handbook of CRM: Achieving Excellence in Customer Management*. Butterworth-Heinemann.
- Payne, A. (2005). *Handbook of CRM: Achieving Excellence in Customer Management* (1 b.). Routledge.
- Payne, A., & Frow, P. (2006). Customer Relationship Management: From Strategy to Implementation. 22(1), 135-168.
- Payne, A., & Frow, P. (2013). *Strategic Customer Management: Integrating Relationship Marketing and CRM*. Cambridge University Press.
- Peppers, D., & Rogers, M. (1993). *The One to One Future: Building Relationships One Customer at a Time* (1 b.). Currency Doubleday.
- Peppers, D., & Rogers, M. (1997). *The One to One Future: Building Relationships One Customer at a Time*. Currency Doubleday.
- Peppers, D., & Rogers, M. (2008). *Rules to Break and Laws to Follow: How Your Business Can Beat the Crisis of Short-Termism* (1 b.). Wiley.
- Peppers, D., & Rogers, M. (2011). *Managing Customer Relationships: A Strategic Framework* (2 b.). John Wiley & Sons Inc.
- Peppers, D., Rogers, M., & Dorf, B. (1999). Is Your Company Ready For One To One Marketing? *Harvard Business Review*.

- Peyman, A., & Safanaz, H. (2020). Analyzing the Relationship between KM, CRM, and Data Mining. *Knowledge Management Review*.
- Porter, M. E., & Millar, V. E. (1985). How Information Gives You Competitive Advantage. 63(4), 149-160. *Harvard Business Review*.
- Quinlan, J. R. (1986). *Induction of Decision Trees*. Kluwer Academic Publishers, Boston - Manufactured in The Netherlands, 1, 81-106.
- Rajola, F. (2003). *Customer Relationship Management: A Strategic Approach*. Springer.
- Ranjan, J., & Bhatnagar, V. (2011). Role of knowledge management and analytical CRM in business: Data mining based framework. *Learning Organization*, 18(2), 131-148.
- Reichheld, F. F. (2001). *The Loyalty Effect: The Hidden Force Behind Growth, Profits, and Lasting Value*. Harvard Business Review Press.
- Reichheld, F. (2003). *The Loyalty Effect: The Hidden Force Behind Growth, Profits, and Lasting Value*. Harvard Business Review Press.
- Reinartz, W., & Kumar, V. (2002, July). The Mismanagement of Customer Loyalty. *Harvard Business Review*.
- Rish, I. (2001, August 4-6). An empirical study of the naive Bayes classifier. *IJCAI 2001 workshop on empirical methods in artificial intelligence*, 41-46.
- Rodpysh, R. (2022). The Impact of Data Mining on Customer Relationship Management. *Data Science Review*.
- Russell, S., & Norvig, P. (2009). *Artificial Intelligence: A Modern Approach* (3 b.). Pearson.
- Rust, R. T., Lemon, K. N., & Zeithaml, V. A. (2004). Return on Marketing: Using Customer Equity to Focus Marketing Strategy. 69(4).
- Sancak, S. (2008). Saldırı tespit sistemleri tekniklerinin karşılaştırılması. *Gebze İleri Teknoloji Enstitüsü, Sosyal Bilimler Enstitüsü*.
- Savaş, S., Topaloğlu, N., & Yılmaz, M. (2012). Veri madenciliği ve Türkiye'deki uygulama örnekleri. *İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi*, 21, 1-23.
- Schneider, D., Corbae, G., & Jensen, J. B. (2003). *Marketing 2.0 – Strategies for Closer Customer Relationship*. Springer.
- Schubert, E., Sander, J., Ester, M., & Kriegel, H. (2017). DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN. *ACM Transactions on Database Systems*, 42(3), 19.
- Seber, G. A., & Lee, A. J. (2003). *Linear Regression Analysis* (2 b.). Wiley.

- Sen, A., & Sinha, A. P. (2011). IT Alignment Strategies for Customer Relationship Management. *Decision Support Systems*, 51(3), 609-619.
- Seyrek, İ. H., & Ata, H. A. (2010). Veri Zarflama Analizi ve Veri Madenciliği ile Mevduat Bankalarında Etkinlik Ölçümü. *BDDK Bankacılık ve Finansal Piyasalar Dergisi*, 4(2), 67-84.
- Shanks, G., & Bekmamedova, N. (2012). Achieving benefits with business analytics systems: an evolutionary process perspective. *Journal of Decision Systems*, 21(3), 231-244.
- Sharma, A. K. (2007). *Customer Relationship Management: Concepts and Application*. Paperback.
- Sheth, J. N., & Parvatiyar, A. (2000). *Handbook of Relationship Marketing*. Sage Publications.
- Siddiqi, J. I., Akhgar, B., & Wise, T. (2002). A framework for the implementation of a Customer Relationship Management strategy in retail sector. Department of Applied Computer Science, Sheffield Hallam University (UK).
- Siddiqi, J. I., Akhgar, B., & Wise, T. (2006). Framework for Implementation of CRM Strategy in Retail Sector. *The Journal of Database Marketing*.
- Silahtaroglu, G. (2008). *Kavram ve Algoritmalarıyla Temel Veri Madenciliği (2 b.)*. İstanbul: Papatya Yayıncılık.
- Skowron, J., Delrieu, B., Davis, K., & Brad, D. (2020). *CRM Success Strategies*.
- Smith, J., & Brown, K. (2022). *Big Data and Management*. Data Publishers.
- Sneath, P., & Sokal, R. (1973). *Numerical Taxonomy*. Freeman.
- Sokal, R., & Michener, C. (1958). A Statistical Method for Evaluating Systematic Relationships. 38, 1409-1438.
- Solomon, M. R. (2016). *Consumer Behavior: Buying, Having, and Being (12 b.)*. Pearson Education.
- Swift, R. (2000). *Accelerating Customer Relationships: Using CRM and Relationship Technologies (1 b.)*. Prentice Hall.
- Swift, R. (2001). *Accelerating Customer Relationships: Using CRM and Relationship Technologies*. Prentice Hall.
- Sykes, A. O. (1993). *An Introduction to Regression Analysis*. University of Chicago Law School:
https://chicagounbound.uchicago.edu/law_and_economics/?utm_source=chicagounbou

nd.uchicago.edu%2Flaw_and_economics%2F51&utm_medium=PDF&utm_campaign=PDFCoverPages adresinden alındı

Şahin, Y. (2024). Türk Bankacılık Sektöründe Kredi Riski Yönetimi Ve Riskin Ölçülmesi. Trabzon: Karadeniz Teknik Üniversitesi Sosyal Bilimler Enstitüsü.

Tan, P. N., Steinbach , M., Karpatne, A., & Kumar, V. (2018). Introduction to Data Mining (2 b.). Pearson.

Tan, P.-N., Steinbach, M., & Vipin , K. (2018). Introduction to Data Mining (2 b.). Pearson.

Turban, E., Sharda, R., Delen, D., & King, D. (2011). Business Intelligence: A Managerial Approach. Prentice Hall.

Tütüncü , T. E. (2022). İstatistik Bilim Dalı Makine Öğrenmesi Algoritmaları İle Kredi Temerrüt Riskini Tahmin Etme. Bursa: Bursa Uludağ Üniversitesi Sosyal Bilimler Enstitüsü .

Uçkan, C. G., Latif, H., & Çeltek, E. (2003). Turistik Ürün Oluşturma sürecinde Yeni bir yaklaşım MİY. Pazarlama Dünyası(4).

Wang , L., & Pang, P. (2021). Utilizing Apriori and Fuzzy Algorithms for Customer Insights. Journal of Data Analysis.

Wang, J., & Wu, Y. (2018). Investigating Customer Relationships in the Banking Sector with Decision Trees. Banking and Finance Journal.

Wang, Z., & Zhou, Y. (2023). Optimization of Customer Loyalty Programs with AI. AI and Business Journal.

Williams, S., & Adams, D. (2021). Knowledge Discovery and Data Mining. CRC Press.

Witten , I. H., Frank , E., & Hall , M. A. (2011). Data Mining: Practical Machine Learning Tools and Techniques (3 b.). Morgan Kaufmann.

Yapıcı , A. P., Özel , A., & Ayça , C. (2010). Oracle data miner ile kredi ödemeleri üzerine bir veri. Bitirme Ödevi.

Yıldız , H. (2019). Finans Sektöründe Veri Madenciliği Kredi Skorlama. İstanbul: Milli Savunma Üniversitesi Hızırhan Havaacılık Ve Uzay Teknolojileri Enstitüsü.

Zhang, H. (2004). The optimality of naive Bayes. Faculty of Computer Science University of New Brunswick, 1(2), 3.