T.C. İSTANBUL TİCARET ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

LOJİSTİK SEKTÖRÜNDE MAKİNE ÖĞRENMESİ MODELLERİ YARDIMI İLE UYGUN PAKET TİPİNİN SEÇİLMESİ

Yasin Can KILIÇKAP

Danışman
Doç. Dr. Berk AYVAZ

İkinci Tez Danışmanı
Dr. Öğretim Üyesi Emine Elif NEBATİ

YÜKSEK LİSANS TEZİ
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI
İSTANBUL - 2023

KABUL VE ONAY SAYFASI

Yasin Can KILIÇKAP tarafından hazırlanan **"Lojistik Sektöründe Makine Öğrenmesi Modelleri Yardımı ile Uygun Paket Tipinin Seçilmesi"** adlı tez çalışması 23/06/2023 tarihinde aşağıdaki jüri üyeleri önünde başarı ile savunularak, İstanbul Ticaret Üniversitesi Fen Bilimleri Enstitüsü **Endüstri Mühendisliği Anabilim Dalı**'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Danışman	**Doç. Dr. Berk AYVAZ** İstanbul Ticaret Üniversitesi	.....
Jüri Üyesi	**Prof. Dr. Sibkat KAÇTIOĞLU** İstanbul Ticaret Üniversitesi	.....
Jüri Üyesi	**Doç. Dr. Ali Osman KUŞAKÇI** İbn Haldun Üniversitesi	.....

Onay Tarihi : **16.06.2023**

İstanbul Ticaret Üniversitesi, Fen Bilimleri Enstitüsünün 16.06.2023 tarih ve 2023/389 numaralı Yönetim Kurulu Kararının 2. maddesi gereğince, ders yüklerini ve tez yükümlülüğünü yerine getirdiği belirlenen "Yasin Can KILIÇKAP" adlı öğrencinin mezun olmasına oy birliği ile karar verilmiştir.

Prof. Dr. Doğan KAYA
Enstitü Müdürü

AKADEMİK VE ETİK KURALLARA UYGUNLUK BEYANI

İstanbul Ticaret Üniversitesi, Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmasında,

- tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- ve bu tezin herhangi bir bölümünü bu üniversitede veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.

16.06.2023

Yasin Can KILIÇKAP

İÇİNDEKİLER

Sayfa

İÇİNDEKİLER i
ÖZET ii
ABSTRACT iii
TEŞEKKÜR iv
ŞEKİLLER v
ÇİZELGELER vi
SİMGELER VE KISALTMALAR vii
1. GİRİŞ 1
2. LİTERATÜR ÖZETİ 6
3. METODOLOJİ 13
4. VERİ SETİ, UYGULAMALAR VE SONUÇLAR 21
4.1 Veri Seti 21
4.2 Makine Öğrenmesi Modellerinin Uyğulanması 27
4.3 Yönetim Etkisi 53
5. SONUÇ VE ÖNERİLER 55
KAYNAKLAR 57
EKLER 61
EK A. Python Kodları 61
ÖZGEÇMİŞ 69

ÖZET

Yüksek Lisans Tezi

LOJİSTİK SEKTÖRÜNDE MAKİNE ÖĞRENMESİ MODELLERİ YARDIMI İLE UYGUN PAKET TİPİNİN SEÇİLMESİ

Yasin Can KILIÇKAP

İstanbul Ticaret Üniversitesi
Fen Bilimleri Enstitüsü
Endüstri Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Berk AYVAZ

İkinci Tez Danışmanı: Dr. Öğretim Üyesi Emine Elif NEBATİ
2023, 69 sayfa

Bu çalışmada, lojistik sektörlerinde maliyet ve verimlilik açısından önemli bir yere sahip olan paketleme sürecindeki siparişler için uygun paket tipinin makine öğrenmesi modelleri yardımı ile belirlenmesi problemi üzerinde durulmuştur. Siparişlerin içerisinde çok fazla boşluk bulunduracak büyük paketlerde gitmesi ya da küçük paketlere tam sıoğmadan gönderilmesi marka imajını zedelemekteddir ve son kullanıcının siparişten memnun kalmamasına sebebiyet verebilmektedir.

Lojistik Regresyon, Destek Vektör Makineleri, Karar Ağaçları, K-En Yakın Komşu Sınıflandırıcısı, Rassal Orman Sınıflandırıcısı, Gaussian Naive Bayes Sınıflandırıcıs ve XGBoost Sınıflandırıcısı modelleri çapraz geçerleme yapılmadan ve çapraz geçerleme yapılarak uygulanmıştır.

Belirtilen modeller uygulandıktan sonra en doğru ve güvenilir olan modelin 10 kat çapraz geçerlemenin yapıldığı XGBoost Sınıflandırıcısı olduğu gösterilmiştir. Bu model sonucunda siaprişlerin yaklaşık olarak %83'ü doğru bir şekilde paketlenebilmektedir. Sürekli değişen personeller sebebiyle bilgi birikiminin tam olarak oluşamaması ve süre kaybının minimuma indirilmesi sayesinde bu modelin lojistik sektörlerinde uygulanabilir olduğu gösterilmiştir.

Anahtar Kelimeler: Konteynır yükleme problemi, lojistik şirketleri, makine öğrenmesi, paketleme, palet yükleme problemi.

ABSTRACT

M.Sc. Thesis

SELECTING APPROPRIATE TYPE OF PACKAGE WITH MACHINE LEARNING MODELS IN LOGISTIC COMPANIES

Yasin Can KILIÇKAP

İstanbul Commerce University
Graduate School of Applied and Natural Sciences
Department of Industrial Engineering

Supervisor: Assoc. Prof. Dr. Berk AYVAZ

Second Supervisor: Assist. Prof. Dr. Emine Elif NEBATİ
2023, 69 pages

In this study, the problem of selecting the appropriate package type for the orders in the packaging process, which has an important role in terms of cost and efficiency in the logistics sectors, is emphasized with the help of machine learning models. Sending orders in large packages that will contain too much space or sending them in small packages without fitting them completely damages the brand image and may cause the end user to be dissatisfied with the order.

Logistic Regression, Support Vector Machines, Decision Trees, K-Nearest Neighbor Classifier, Random Forest Classifier, Gaussian Naive Bayes Classifier and XGBoost Classifier models were applied without cross-validation and cross-validation.

After applying the specified models, it has been shown that the most accurate and reliable model is the XGBoost Classifier with 10-fold cross validation. As a result of this model, approximately 83% of orders can be packed correctly. It has been shown that this model is applicable in the logistics sectors, thanks to the fact that the knowledge cannot be fully formed due to the constantly changing personnel and the loss of time is minimized.

Keywords: Container loading problem, logistic companies, machine learning, packaging, pallet loading problem.

TEŞEKKÜR

Bu araştırma için beni yönlendiren, karşılaştığım zorlukları bilgi ve tecrübesi ile aşmamda yardımcı olan değerli Danışman Hocam Doç. Dr. Berk AYVAZ'a teşekkürlerimi sunarım.

Tezimin imalat aşamasındaki desteklerinde dolayı Arvato Bertelsmann Supply Chain Solutions şirketine teşekkür ederim.

Tezimin her aşamasında beni yalnız bırakmayan aileme sonsuz sevgi ve saygılarımı sunarım.

Yasin Can KILIÇKAP
İSTANBUL, 2023

ŞEKİLLER

Sayfa

Şekil 1.1. Farklı lojistik firma entegrasyonları 2
Şekil 3.1. Karar ağacı stratejik yapısı 14
Şekil 3.2. Farklı kümeye ait verilerin ayrıştırılması 15
Şekil 3.3. Hard margin 15
Şekil 3.4. K en yakın komşu sınıflandırıcısı K=3 ve K=5 16
Şekil 3.5. Sigmoid fonksiyonu 18
Şekil 3.6. Rassal orman sınıflandırıcısı 19
Şekil 4.1. Örnek sayısının sınıf değişkenlerine göre dağılım 28
Şekil 4.2. Korelasyon değerleri 30
Şekil 4.3. Farklı sınıflar için ürünlerin toplam uzunluğunun dağılımı 31
Şekil 4.4. Sınıf içi kareli hata değerleri 35
Şekil 4.5. Özniteliklerin önem dereceleri 37
Şekil 4.6. Karar ağacı dönüm noktaları 39
Şekil 4.7. Özniteliklerin önem dereceleri 43
Şekil 4.8. Özniteliklerin önem dereceleri 45
Şekil 4.9. Modellerin karşılaştırılması (3 kat çapraz geçerleme) 48
Şekil 4.10. Modellerin karşılaştırılması (5 kat çapraz geçerleme) 49
Şekil 4.10. Modellerin karşılaştırılması (10 kat çapraz geçerleme) 50
Şekil 4.12. Doğru/yanlış – pozitif/negatif tablosu 51
Şekil 4.13. ROC Eğrisi ve AUC değerleri 52
Şekil A.1. Gerekli kütüphanelerin eklenmesi 61
Şekil A.2. Excelin tanımlanması ve sütun kontrolleri 62
Şekil A.3. Excelde yer alan veri tiplerinin kontrolü 62
Şekil A.4. Sınıf değişkenlerinin dağılımı 62
Şekil A.5. Her bir öznitelik için istatistiki değerler 62
Şekil A.6. Isı haritasının çıkartılması 62
Şekil A.7. Korelasyon değerlerinin hesaplanması 62
Şekil A.8. En yüksek korelasyon değerine sahip 15 öz nitelik 62
Şekil A.9. Her bir öz niteliğin dağılım grafiği 63
Şekil A.10. Eğitme ve test verilerinin belirlenmesi 63
Şekil A.11. Standardizasyon ve Normalizasyon 63
Şekil A.12. Lojistik regresyon modelinin uygulanması 64
Şekil A.13. İlk 5 örneğin sınıf değerinin tahmin edilmesi 64
Şekil A.14. K-en yakın komşu yöntemi modelinin uygulanması 65
Şekil A.15. Sınıf içi kareli hata değerlerinin hesaplanması 65
Şekil A.16. Karar ağacı modelinin uygulanması 66
Şekil A.17. Öz niteliklerin önem derecesinin belirlenmesi 66
Şekil A.18. Hata matrisinin oluşturulması 66
Şekil A.19. Karar ağacının çizilmesi 66
Şekil A.20. Destek vektör makineleri modelinin uyğulanması 67
Şekil A.21. Gaussian naïve bayes modelinin uyğulanması 67
Şekil A.22. Rassal orman modelinin uyğulanması 67
Şekil A.23. XGBoost modelinin uyğulanması 67
Şekil A.24. 10 kat çapraz geçerleme 67
Şekil A.25. ROC ve AUC değerleri 68

ÇİZELGELER

Sayfa

Çizelge 4.1. Sipariş ve ürün adedi dağılımları 23
Çizelge 4.2. Paket tiplerine göre sipariş dağılımları 23
Çizelge 4.3. Örnek sipariş ürün ebatları 25
Çizelge 4.4. Paket tiplerine göre sipariş dağılımları 25
Çizelge 4.5. Ebat kaynaklı aykırı veriler 25
Çizelge 4.6. Paket tipi kaynaklı aykırı veriler 26
Çizelge 4.7. Paket tipi kaynaklı aykırı veriler 26
Çizelge 4.8. Özniteliklerde ve sınıf değerinde yer alan boş değer sayısı 27
Çizelge 4.9. Özniteliklerin ve sınıf değişkeninin veri tipi 28
Çizelge 4.10. Farklı sınıf değişkenlerine göre istatistiki değerler 29
Çizelge 4.11. Özniteliklerin sınıf değişkenine etkisi 31
Çizelge 4.12. Lojistik regresyon model sonuçları 33
Çizelge 4.13. Farklı örnekler için sınıf oranları 33
Çizelge 4.14. K-en yakın komşu model sonuçları 34
Çizelge 4.15. Sınıf içi kareli hata toplamları 35
Çizelge 4.16. Farklı örnekler için sınıf oranları 36
Çizelge 4.17. Karar ağacı model doğruluk oranı 36
Çizelge 4.18. 5 farklı örnek için sınıf oranları 37
Çizelge 4.19. Hata matrisi 38
Çizelge 4.20. Destek vektör makineleri model sonuçları 40
Çizelge 4.21. 5 farklı örnek için sınıf oranları 40
Çizelge 4.22. Naive bayes model sonuçları 41
Çizelge 4.23. 5 farklı örnek için sınıf oranları 41
Çizelge 4.24. Rassal orman model sonuçları 42
Çizelge 4.25. 5 farklı örnek için sınıf oranları 42
Çizelge 4.26. Hata matrisi 43
Çizelge 4.27. XGBoost model sonuçları 44
Çizelge 4.28. 5 farklı örnek için sınıf oranları 44
Çizelge 4.29. Hata matrisi 46
Çizelge 4.30. Modellerin karşılaştırılması 47
Çizelge 4.31. Modellerin karşılaştırılması (3 kat çapraz geçerleme) 48
Çizelge 4.31. Modellerin karşılaştırılması (5 kat çapraz geçerleme) 49
Çizelge 4.31. Modellerin karşılaştırılması (10 kat çapraz geçerleme) 50
Çizelge 4.34. Sınıf değişkenleri ve AUC değerleri 52

SİMGELER VE KISALTMALAR

AUC	Eğrinin altında kalan alan
CLP	Konteynır yükleme problemi
CPU	İşlemci
GPU	Grafik işlemci ünitesi
HALNS	Hibrit uyarlanabilir geniş komşuluk araması
HCLP	Hetetoren konteynır yükleme problemi
MIP	Tamsayılı programlama problemi
MÖ	Makine öğrenmesi
PL	Parti lojistik
PLP	Palet yükleme problemi
RCH	Yapıcı buluşsal yöntem
RFID	Radyo frekansı ile tanımlama teknolojisi
ROC	İşlem karakteristik eğrisi

1. GİRİŞ

Günümüz dünyasında bünyesinde lojistik süreçleri barındıran şirketlerin ihtiyaçları giderek artmaktadır. Bu ihtiyaçlar bazen şirket içerisindeki yenilikler ile giderilebilirken bazen de farklı şirketlerden destek alınarak giderilmektedir. Lojistik şirketleri, bu ihtiyaçların giderilmesinde destek alınan en önemli şirketlerdendir. İhtiyaçların seviyelerine göre lojistik şirketleri ile farklı seviyelerde bağlantılar kurularak ve onların bilgi birikiminden yararlanılarak süreç akışında yer alan lojistik faaliyetlerinin üreticiden son kullanıcıya kadar optimize edilmesi ve verimlilik artırılması amaçlanır (Kotlars ve Skribans, 2016).

Şirketlerin ihtiyaçlarının artarken farklılaşması da ek bir zorluk yaratır. Kimi şirketler sadece ürünleri son kullanıcıya ulaştırma yani transfer sürecinde desteğe ihtiyaç duyarken, kimileri de lojistik bilgilerinin yeterli olmamasından dolayı bütün süreçlerin iyileştirilmesinde desteğe ihtiyaç duyarlar. Bu sebeple de bu şirketlerin lojistik şirketleri ile farklı seviyelerdeki entegrasyonların kurulması en uygun çözüm olarak görülmektedir. (Sheikh ve Rana, 2014).

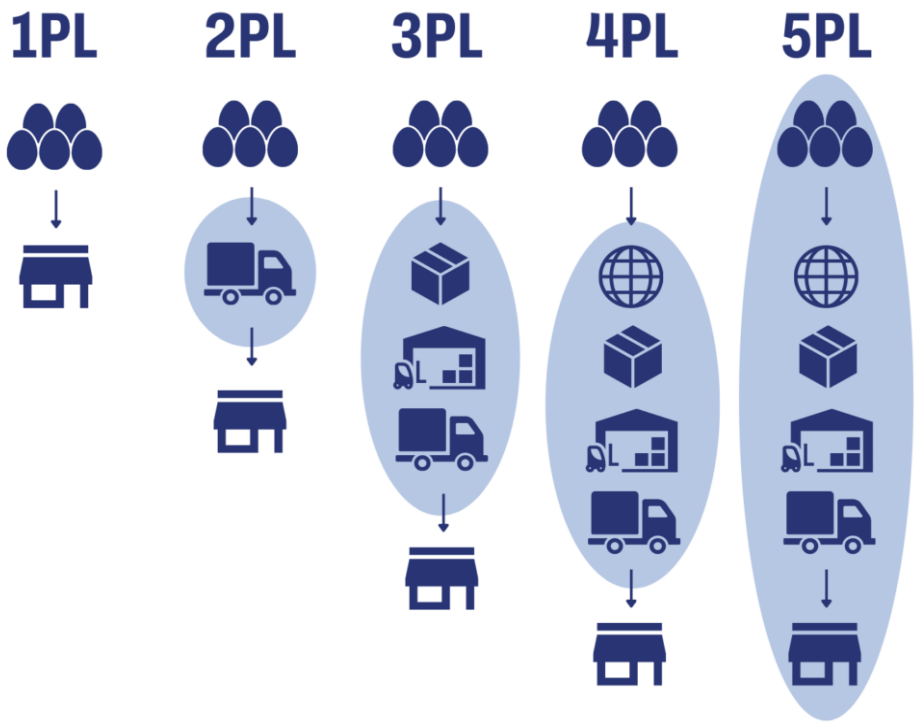
Günümüzde lojistik şirketleri verdikleri hizmet çeşitliliğine ve yetkinliklerine göre 5 farklı başlık altında incelenmektedir.

- Birinci Parti Lojistik Şirketleri (1PL): Belirli bir konumdan belirli bir konuma ürün gönderen şirketlerdir. Bir üreticinin ürettiği ürünleri doğrudan mağazasına taşımasını örnek olarak verebiliriz.
- İkinci Parti Lojistik Şirketleri (2PL): Ürünlerin taşınması sürecini gerçekleştiren ve kargo araçlarına sahip olan şirketlerdir. Kargo firmaları örnek olarak verilebilir.
- Üçüncü Parti Lojistik Şirketleri (3PL): Bazı şirketler lojistik süreçlerinin çok büyük bir bölümünün ya da hepsinin farklı bir tedarikçi tarafından yapılmasını ister. Bu tedarikçi firmalar üçüncü parti lojistik şirketleridir. Genellikle depolarda yapılan elleçleme süreçlerini de içerisinde barındırır. Örneğin bir B firması ürünlerini A firması aracılığıyla

mağazalara gönderirken paketleme konusunda da destek alıyorsa A firması üçüncü parti lojistik şirketi olarak değerlendirilir.

- Dördüncü Parti Lojistik Şirketleri (4PL): Burada yer alan şirketler sadece lojistik süreçler ile değil aynı zamanda bazı stratejik kararların verilmesinde de rol sahibidir. Örneğin A firması B firmasının üretim planlarını da yönetiyorsa A firması dördüncü parti lojistik şirketi olarak değerlendirilir.
- Beşinci Parti Lojistik Şirketleri (5PL): Bu şirketlerde makine ögrenmeleri, yapay zeka, radyo frekansı tanımlama (RFID) gibi çözümler ön plandadır. Genellikle süreçlerin içerisinde otomasyonlar yer almaktadır.

Şekil 1.1'de görüldüğü gibi birinci parti lojistik şirketlerinden beşinci parti lojistik şirketlerine ilerledikçe sorumluluğun arttığı, yapılan işlerin çeşitliliğinin arttığı ve üretim sürecinin de üreticiden çıkarak lojistik şirketine kaymaktadır.



Şekil 1.1 Farklı lojistik firma entegrasyonları

Lojistik şirketleri yukarıda belirtilen farklı görevleri yerine getirirken başarılı olmayı ve karlı olmayı amaçlar. Bu şirketleri başarılı ve karlı olmaları için çok fazla sayıda etmen bulunmaktadır. Bu etmenler genel anlamda verimlilik değerleri ve maliyet olmak üzere iki ana başlık altında değerlendirilir. Verimlilik değerleri süreç akışında yer alan her bir sürecin verimliliğini kapsar ve olabildiğince maksimum yapılması amaçlanır. Maliyet ana başlığı altında ise 4 ana kategori yer almaktadır.

- Tesis maliyeti
- İşgücü maliyeti
- Sermaye maliyeti
- Operasyon maliyeti

Yukarıda yer alan operasyon maliyeti ise kendi içerisinde alt başlıklara ayrılabilir.

- Stoklama maliyeti
- Toplama maliyeti
- Paketleme maliyeti
- Dağıtım maliyeti

Her bir maliyet kalemi lojistik şirketleri için çok önemlidir ve olabildiğince minimum değere getirilmesi amaçlanır. Paketleme ve Dağıtım maliyeti yer alan maliyetler arasında önemli bir yere sahiptir. Elleçleme işlemlerinin yapıldığı lojistik şirketlerinde her bir ürün ya da sipariş için uygun koli ya da paket tipinin seçilmesi gerekmektedir. Gereğinden büyük hacimli koliler seçilmesi dağıtım maliyetini doğrudan etkileyecektir. Dağıtım aşamasında ürünün hacminden bağımsız olarak o ürünün içine yerleştirildiği kolinin hacim değerine bakılır ve maliyet oluşturulur. Bu sebeple maliyet değerini arttırmamak için uygun koli tipi seçilmesi oldukça önemlidir.

Bazı ürün tipleri spesifik özellikler taşır ve paketleme işlemlerinde bu özelliklerin de göz önünde bulundurulması gerekir. Ürün tiplerindeki ve özelliklerindeki farklılıkların artması bu ürünlerin paketlenmesi sürecindeki uygulanan eforu da arttırmaktadır (ElMaraghy, 2013). Her ürünün kendine özgü

bir en, boy ve yükseklik değeri bulunurken kırılabilir olup olmaması ya da akıcı bir madde olup olmaması gibi farklı özellikler de bulunmaktadır. Bu farklılıkları göz önünde bulundurup uygun koli tipini seçmek zor ve zaman alıcı bir süreç oluşturmaktadır. Bazen siparişlerde birden fazla ürün olduğu zaman her bir ürün için bütün özellikleri dikkat almak gerektiği düşünüldüğünde süreç daha da zorlaşmaktadır.

Bu problem genellikle matematiksel modeller ile çözülmektedir. Literatürde bu problemler konteynır yükleme problemi (CLP) ya da palet yükleme problemi (PLP) olarak karşımıza çıkmaktadır. CLP ve PLP üç boyutlu bir paketleme problemidir (Gajda vd., 2020). Bu problemlerde ürünler bulunmaktadır ve bu ürünler bir palete ya da konteynıra bazı kısıtlar sağlanarak yüklenmektedir. Bazı durumlarda bütün ürünler paketlenirken bazı durumlarda ise bazı ürünler açıkta kalmaktadır. Bazı ürünlerin açıkta kaldığı durumlarda maksimize edilecek amaç fonksiyonu belirlenir. Bu amaç fonksiyonu kar, paketlene ürün sayısı, paketlenen hacim gibi farklı kriterler olabilmektedir.

CLP dikdörtgensel ürünler ile ilgilenmektedir. Bu yaklaşım gerçek hayatta küre, silindir gibi ürünlerin olduğu bir ortamda modelleri verimsiz kılabilmektedir. Bunların sonucunda sadece geometrik özelliklerin incelenmesi yerine; ağırlık, kırılganlık gibi farklı faktörler de incelenmelidir (Bortfeldt ve Wäscher, 2013).
Matematiksel modellerin yanı sıra farklı makine ögrenmesi (MÖ) yöntemleri de bu problemlerin çözümü için kullanılmaktadır. Son yıllarda fazla sayıda firma bu yaklaşımlara yönelmektedir. MÖ girdi verilerini kullanıp analiz eden ve belirli bir aralıkta sonuçları tahmin eden algoritmalar kullanır. Denetimli ve denetimsiz olmak üzere iki farklı MÖ tipi bulunmaktadır. Denetimli öğrenmede veriler için etiketler ya da sınıf değerleri bulunurken denetimsiz ögrenme de ise etiketler bulunmamaktadır. Denetimsiz ögrenme yöntemleri genellikle kümeleme problemleri için kullanılmaktadır (Kotsiantis vd., 2006).

Bu yüksek lisans tezinde ise bir üçüncü parti lojistik şirketinin paketleme verileri incelenmiş ve farklı MÖ yöntemleri uygulanmıştır. Uygulana makine ögrenmesi yöntemleri sırasıyla şöyledir.

- Lojistik Regresyon
- Karar Ağaçları
- Destek Vektör Makineleri
- K-en Yakın Komşu Algoritması
- Naive Bayes Sınıflandırıcıları
- Rassal Orman Sınıflandırıcısı
- XGBoost Sınıflandırıcı

2. LİTERATÜR ÖZETİ

Uygun koli tipinin ya da paket tipinin belirlenmesi için farklı çalışmalar yapılmış ve bu çalışmalar farklı şirketlerde uygulanmıştır. Bazı çalışmalar matematiksel modelleri kullanarak bir çözüm yolu geliştirirken bazı çalışmalar ise makine öğrenmesine ya da yapay zekaya başvurmaktadır. Matematiksel modeller gerçek dünyadaki bir problemin modele ya da bir matematiksel probleme dönüşmüş halidir (Voskogluo, 2020). Daha farklı bir söylemle ise gerçek dünyadaki problemler karmaşık süreçlere sahiptir ve bu problemlerin hızlı ve kolay bir yoldan çözülmesi zordur. Matematiksel modellerin sayesinde bazı parametreler ve denklemler ile gerçek dünyadaki problemler basitleştirilir ve çözüme ulaşmaya daha uygun hale getirilir. Bu sayede optimum sonuç bulmak mümkün hale gelmiş olur. Matematiksel modellerin olumsuz tarafı ise kabul edilen bazı varsayımların gerçek dünya da kolayca gerçekleşememesi ya da varsayılamamasıdır.

Uygun koli tipinin ya da paket tipinin belirlenmesi genellikle PLP ve CLP olarak ele alınmaktadır. Bu iki problem türü için genel fikir ise bu problemlerin optimum sonucunun bulunmasının çok zaman aldığı hatta bazı durumlarda mümkün olmadığıdır. Bu sebeple birçok araştırmacı bu problemlerin çözümü için stokastik yöntemlere ve doğrusal olmayan modellere yönelmişlerdir (Lau vd., 2006). Stokastik yöntemler ve doğrusal olmayan modeller zaman ve maliyet açısından farklı etkilere sahiptir. Stokastik yöntemler optimal sonuca yakın sonuçlar verirken maliyet açısından pahalıdırlar. Doğrusal olmayan modeller ise optimum sonucu uzun sürede verirken aynı zamanda da pahalıdırlar.

Yapılan çalışmaların birçoğu matematiksel modellere dayanmaktadır. Sayısal ve pratik olan bazı kısıtlamalara dayalı PLP çalışması yapılmıştır (Gajda vd., 2020). Ele alınan kısıtlar ürünlerin ağırlığı, ağırlığın palet üzerindeki dağılımı, palet dengesi, yükleme önceliği, ürünlerin yerleşim planları, ürünlerin istiflenmesi, ürünlerin tehlikeli ürün sınıfına girip girmediği ve sağlamlıktır. Bu çalışmada buluşsal bir yöntem kullanılmıştır. kullanılan bilişsel yöntem Rastgele

Yapıcı buluşsal yöntem (RCH) olarak bilinmektedir. Kullanılan buluşsal yöntem ile birlikte saniyeler içerisinde bir palet için çözüm üretilmektedir.

Yapılan bir diğer farklı çalışma da matematiksel model kullanılarak PLP çalışması yapılmıştır (Gzara vd., 2020). Geliştirilen matematiksel modelin uygulanması sonucu yaklaşık 60 saniye içerisinde bir palet için sonuç alındığı gözlemlenmiştir. Geliştirilen matematiksel modelde katman tabanlı sütun oluşturma yaklaşımı bulunmaktadır ve farklı kısıtlat ile bu yaklaşım geliştirilmiştir. Ele alınan bazı kısıtlar şu şekildedir.

- Dikey destek
- Küçülen destek yüzeyleri
- Farklı şekillere sahip öğeler
- Ağırlık sınırlaması
- Planogram dizileme

Var olan kısıtlardan en önemlilerinden bir tanesi Planogram dizilemedir. Bu kısıt ürünlerin palet üzerine yerleşimini ele alır. Ürünler palet üzerine yerleştirilirken yükleme ve boşaltmada harcanan eforun minimum seviyeye indirilmesi amacına hizmet eder. Aynı şekilde bir paletin doluluk oranını da maksimum seviyeye getirmeye çalışırken depolarda ya da farklı yerlerde stok maliyetinin de yüksek seviyelere çıkmasını engeller. Geliştirilen matematiksel model, öğe aralığını ideal şekilde ayarlayabilmek ve minimum %70 oranında kullanılabilirlik sağlamak amacıyla ikinci dereceden koni programını önerir. Bu çalışmada çeşitli öğe formları ve küçültülmüş destek yüzeyleri de ele alındı. Bu çalışma bu kısıtların ele alındığı ilk çalışmadır.

PLP ve CLP çözümleri için dal ve sınır yöntemleri de kullanılmaktadır. Dal ve sınır yöntemi kullanılarak geliştirilen bir matematiksel modelde tek bir konteynırı yüklemek ele alınmıştır (Martello vd., 2000). Farklı ürün sayıları için çözümler incelendiğinde ürün sayısı 90'dan fazla ise idealden uzak çözümler ile karşılaşılırken ürün sayısının 90'dan az olduğu durumlarda ise ideal çözümlere yakın çözümler elde edildiği fark edilmiştir.

Bir farklı çalışmada ise Mixed Integer Programming (MIP) olarak bilinen karışık tam sayılı programlama yöntemi kullanılmıştır (Paquay vd., 2018). Çalışmada farklı şekillere ve hacim değerlerine sahip ürünler ele alınarak 3 boyutlu bir yaklaşım ile kullanılmayan alanı minimum seviyeye indirme amaçlanmıştır. Bu çalışma bir havayolu kargo şirketi ile yapılmış uçaklara yüklenen paletlerin verimli bir şekilde doldurulması ve yerleştirilmesi ele alınmıştır. Çalışma iki aşamadan oluşmaktadır ve iki aşama da uyarlanmış yapıcı sezgisel metot içermektedir. İlk aşama da aynı tip ürünler ele alınmış ve kutuların kararlılığı, duyarlılığı, yerleşim konumları, ağırlık kapasitesi gibi kısıtlar ele alınarak paketleme modeli geliştirilmiştir. İkinci aşamada ise aynı model farklı ürünlerin olduğu bir paketleme sistemi için geliştirilmiştir. Ürünlerin ağırlıklarının dağılımına bakılmıştır. Geliştirilen algoritma bir palet için birkaç dakikada çözüm sunabilmektedir. Sunulan çözümde doluluk oranı gibi farklı yaklaşımlar da ele alınmıştır.

Bir farklı çalışmada large-scale heterogenous Container Loading problem (HCLP) olarak bilinen büyük ölçekli heterojen konteynır yükleme problemi ele alınmıştır (Li vd., 2022). Konteynır yükleme probleminin en basit versiyonu tek bir konteynırın olduğu versiyondur. Bu versiyonun bir üst versiyonu ise birden fazla aynı tipteki konteynırın olduğu versiyondur. En üst versiyon ise birden fazla sayıda farklı türde konteynırın olduğu problemdir. Bu çalışmada kullanılmayan alanı minimum seviyede tutacak, tüm ürünlerin yerleşmesini sağlayacak konteynır sayısını ve ürünlerin yerleşim planını belirlemek amaçlanmıştır. Hibrit uyarlanabilir geniş komşuluk araması olarak bilinen HALNS metodu çözüm için kullanılmıştır. her aşamada konteynırlar yüklenir ve tekrardan boşaltılır. Bu sayede sürekli farklı ihtimaller gözlemlenir ve en uygun çözüm yöntemi sunulur. Ağırlık ve süspansiyon kısıtları geliştirilen modeldeki en önemli kısıtlardandır.

Palet optimizasyon sisteminin önerildiği matematiksel modele dayalı bir çalışmada paletlere yerleştirilmesi gereken kolilerin model sonucuna göre optimum sonuç verdiği ve çözümün birkaç dakikada verildiği gözlemlenmiştir (Dowsland, 1987).

Geliştirilen farklı bir matematiksel modelde malzeme taşıma sistemlerinin kısıtları ele alınmıştır (Chen vd., 1991). Geliştirilen matematiksel model iki tabanlı tam sayılı bir formülasyon içermektedir. Geliştirilen modelde kutuların boyutlarının tam sayı olması gerekmemektedir.

Geliştirilen farklı bir matematiksel modelde ise dağıtıcı palet yükleme problemi ele alınmış ve sezgisel bir algoritma geliştirilmiştir (Terno vd., 2000). Çalışmada 2 boyutlu bir çözüm geliştirilip katman hesabı ile 3 boyuta çıkarılması ele alınmıştır.

Dağıtıcı palet yükleme probleminin ele alındığı farklı bir matematiksel modelde ise optimize bir yüklemeye ek olarak doğal dengenin de korunması ve sürekli hale getirilmesi ele alınmıştır (Bischoff vd., 1995).

Geliştirilen farklı bir matematiksel modelde ise PLP 2 boyutta incelenmiş ve sezgisel bir algoritma geliştirilmiştir (Young-Gun ve Kang, 2001). Çalışmada 5 bloklu yapıların çözümlerine hızlıca ulaşılabilinmektedir.

Bir farklı çalışmada dikdörtgen şeklindeki küçük ürünleri L şeklindeki kutulara yerleştirme üzerine odaklanılmıştır (Lins vd., 2003). Yapılan tahminlere göre küçük dikdörtgen cimler büyük dikdörtgen cicimlerin üst tarafında kolayca yerleştirilebilmektedir. Eğer cisimler L şeklinde ise sonuçlar daha doğru gelmektedir.

Dikdörtgen şeklindeki küçük ürünleri L şeklindeki kutulara yerleştirmenin olumsuz tarafları ve süreçte yarattığı verimsizlikler farklı bir çalışmada ele alınmıştır (Birgin vd., 2005). Ele alınan en büyük verimsizlik sürecin süresidir. Farklı bir model ile süre ideal değere getirilmiştir. Bu durumda modelin kullanılabilirliği artmıştır ancak hafıza gereksinimi de artmıştır.

Sezgisel bir algoritma olan genetik algoritma kullanılarak maliyet açısından daha düşük seviyede olan model geliştirilmiştir (Yiğit ve Aydemir, 2018). Bu çalışmada farklı kutular konteynıra yüklenirken kutuların döndürülebilir olması

engellenmiştir. Çalışmadaki genel amaç ise kullanılmayan alanı minimum seviyeye indirmek ve yerleştirilen kutu sayınını maksimum seviyeye çıkarmaktır.

Bir farklı çalışmada ise imalat sektörü üzerinde durulmuş ve sezgisel bir algoritma kullanılmıştır (Sheng vd., 2017). Geliştirilen algoritma siparişlerin süresini dikkate almaktadır. Süresi geçen siparişler bekletilirken süresi geçmeyen siparişler ise konteynıra yüklenmektedir. Model 2 aşamadan oluşmaktadır. İlk olarak benzetimli tavlama algoritması kullanılırken sonrasında ise ağaç grafiği arama algoritması kullanılmıştır.

Bir farklı çalışmada ise karınca kolonisi algoritmasını temel alan iki farklı algoritma geliştirilmiş ve konteynır yükleme problemi için önerilmiştir (Dereli ve Daş, 2010). Önerilen algoritmalar KKS1 ve KKS2 olarak geçmektedir. Sonuçlar karşılaştırıldığı zaman KKS2 daha İYİ sonuçlara sahip olduğu gözlemlenmiştir. Geliştirilen 2 algoritmanın parametreleri faktöriyel tasarım mantığı ile belirlenmiştir.

Konteynır yükleme problemi için yapılan farklı bir çalışmada ise yeni bir genetik algoritma sunulmuştur (Gehring ve Bortfeldth, 1997). 2 aşamadan oluşan algoritmada ilk olarak bir kutu kulesi üretilmesi amaçlanırken ikinci aşamada ise kutu kuleleri optimize bir şekilde düzenlenmektedir.

Genetik algoritmanın kullanıldığı farklı bir çalışmada ise yüklenmesi gereken ürünlerin sayısının maksimum seviyeye getirilmesi amaçlanmıştır (Erdem, 2014). 'En iyi sağ kalanlar' metodu ile ilerlenerek farklı nesiller üretilmiş ve bu nesillerin birbiri ile uyumu kontrol edilerek ideal çözüm sunulmuştur.

Farklı bir çalışmada ise tek bir konteynırın bulunduğu bir konteynır yükleme probleminde paralel genetik algoritma kullanılmıştır (Gehring ve Bortfeldth, 2002). birbirinden bağımsız popülasyonlar bir dizi işleme tabi tutulur ve en başarılı bireyler değiştirilir. En sonda en başarılı birey ideal çözüm olarak sunulur.

Farklı bir konteynır yükleme problemi çalışmasında ise yerleştirilecek ürünlerin sayısı ve ebatları ile bu ürünlerin yerleştirileceği konteynırın ebatları bilindiği bir senaryoda ürünlerin yerleştirilebilecekleri en iyi pozisyonları belirleyen ve tüm ürünleri yerleştiren bir matematiksel model geliştirilmiştir (George ve Robinson, 1980).

Matematiksel modellerin yanı sıra farklı makine öğrenmesi modelleri ve yapay zeka algoritmaları da PLP ve CLP çözümleri için kullanılmıştır.Yapılan bir çalışmada doluluk oranının bu problem için önemli olduğu ve bu kriterin makine öğrenmesi ile ideal bir şekilde kıyaslanabileceği belirtilmiştir (Knoll vd., 2019). Geliştirilen makine öğrenmesi modeli paket tipi seçimi sürecini standartlaştırmaktadır ve doluluk oranını her bir paket tipi için hesaplamaktadır. Toplamda 6 farklı kategori ve 13 adet öz nitelik belirlenmiştir. Sonuç değişkeni olarak da paket tipi ve doluluk oranı kullanılmıştır. belirlenen 6 farklı kategori ve 13 adet öz nitelik aşağıdaki gibidir.

- Boyut
    - Uzunluk
    - Genişlik
    - Yükseklik
    - Hacim
- Şekil
    - Uzunluk/Genişlik
    - Yükseklik/Uzunluk
    - Yükseklik/Genişlik
- Ağırlık
    - Ağırlık
    - Ağırlık/Hacim
- Kalite ve hassaslık
    - Kalite indeksi
    - Elektrostatik değer
- Fiyat
    - Ürün fiyatı

- Tehlikeli ürün
    - Ürünün tehlikeli olup olmadığı

Belirlenmiş olan 6 kategori ve 13 adet öz nitelik kullanılarak ilk önce paket tipi sınıflandırılması modeli kullanılmıştır. Belirlenen paket tipinden sonra ise doluluk oranı hesaplaması yapan tahminleme modeli kullanılmıştır. tahminleme modelinin girdileri paket tipi sınıflandırma modelinin çıktılarıdır.

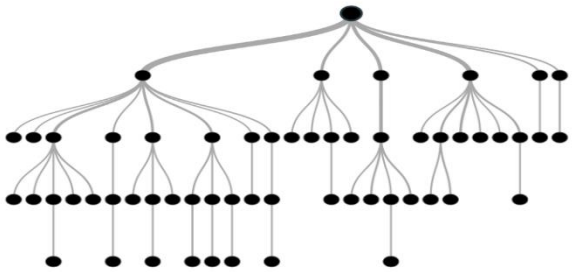
Bir diğer yapılan çalışmada ise PLP çözümüne makine öğrenmesi modelleriyle yaklaşılarak optimize sonuç elde edilmeye çalışılmıştır (Aylak vd., 2021). Yapılan çalışmada lojistik şirketlerinin maliyet düşürme konusunda sıklıkla çalışmalar yaptıkları ve paketleme sürecinde makine öğrenmesi metotları ile maliyetlerin azaltılabileceği belirtilmiştir. Çalışmada sınıflandırma metodu kullanılmıştır. geliştirilen makine öğrenmesi modeli iki aşamadan oluşmaktadır. Birinci aşamada bir paletin her bir yatay katmanı için yerleştirilebilecek maksimum sayıdaki koli sayısı belirlenmiştir. İkinci aşamada ise bir palette kaç yatay katman olması gerektiği ağırlık, uzunluk dayanıklılık gibi kısıtlar çerçevesinde belirlenmiştir. Kullanılan makine öğrenmesi modelleri ise destek vektör makineleri, K en yakın komşu sınıflandırıcısı, Rassal orman sınıflandırıcısı, karar ağaçları ve yapay sinir ağlarıdır. Paletler kullanılan makine öğrenmesi modelleri ile sınıflandırılmıştır ve sonrasında sınıflandırmanın doğruluğu kontrol edildiğinde %89 oranında doğruluk gözlemlenmiştir. Paletlerin doluluk oranı ise yaklaşık %85 civarındadır. Çalışmada 10 kat çapraz geçerleme kullanılmıştır ve öznitelikler olarak uzunluk, genişlik, yükseklik ve talep ele alınmıştır.

3. METODOLOJİ

Bu tez çalışmasında lojistik sektörlerinde elleçleme süreçlerinden bir tanesi olan paketleme sürecinde uygun ve doğru koli seçimi için 7 farklı makine öğrenmesi algoritması kullanılmıştır. bazı algoritmalar için farklı hiper parametreler kullanılarak model çeşitliliği arttırılmıştır. kullanılan 6 farklı makine öğrenmesi algoritması şunlardır.

1. Karar Ağaçları
2. Destek Vektör Makineleri
3. K-en Yakın Komşu Algoritması
4. Naive Bayes Sınıflandırıcıları
5. Lojistik Regresyon
6. Rassal Orman Sınıflandırıcısı
7. XGBoost Sınıflandırıcı

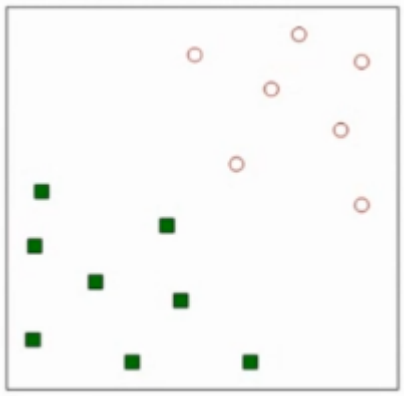
Karar ağaçları: Karar ağaçları algoritması gözetimli algoritmalardan bir tanesidir ve genellikle sınıflandırma problemleri için kullanılır (Bansal vd., 2022). Sınıflandırma problemlerine ek olarak tahminleme problemlerinde de sıklıkla kullanılırlar. Ağaç tabanlı algoritmalar ve modeller genellikle yüksek doğruluk oranları ile bilinirler. Karar ağaçları sadece doğrusal veri setlerini haritalamazlar. Doğrusal olmayan veri setlerinde de haritalama işlemini genellikle başarılı bir şekilde yaparlar. Karar ağaçları en yukarıdan en aşağıya inen bir stratejik yapıya sahiptir (Kantardzic, 2011). Karar ağaçlarında farklı kavramlar bulunmaktadır. Örneğin, yapraklar bir sınıflandırmayı ya da kararı temsil ederken, kök ise en iyi belirleyiciyi temsil etmektedir. Karar ağaçlarının genel bir görünümü Şekil 3.1'deki gibidir.



Şekil 3.1 Karar ağacı stratejik yapısı

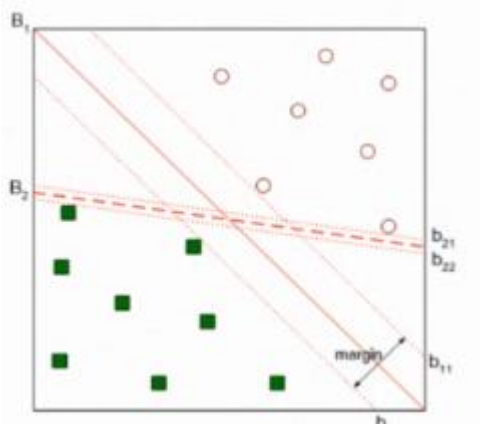
Karar ağaçlarını bazı avantajları ve dezavantajları bulunmaktadır. Anlaşılması ve yorumlanması diğer modellere göre daha kolaydır. Bu kolaylığa karşılık bazı durumlarda verinin net bir şekilde anlaşılması engelleyen karmaşık ağaçlar da üretilebilmektedir. Karar ağaçlarında genellikle parametrik bir yöntem bulunmamaktadır. Bu sebeple de verinin bulunduğu uzay hakkında bir yaklaşıma ihtiyaç duymamaktadır. Dezavantajlarından bir tanesi de aşırı öğrenme ya da ezbere öğrenme durumu yaşanabilmektedir. Bu durumu engellemek için de genellikle budama işlemine başvurulmaktadır.

Destek vektör makineleri: Destek vektör makineleri gözetimli algoritmalardan bir tanesidir ve genellikle sınıflandırma ve tahminleme problemleri için kullanılır (Bansal vd., 2022). Tahminleme ve sınıflandırma problemlerinin dışında fotoğraf ayrıştırma yazı ayrıştırma gibi farklı alanlarda da kullanılmaktadır. Bu algoritmalar karmaşık olan ama veri miktarı bakımından orta seviyede olan veri setleri için ideal yöntemlerden bir tanesidir. Destek vektör makinelerinde farklı Kernel metotları bulunmaktadır. Bu metotlar “Radial”, “Linear” ve “Poly” olarak geçmektedir. İdeal Kernel metodu için 3 farklı metot denenmeli ve doğruluk oranı kontrol edilmelidir. Destek Vektör Makinelerinde Şekil 3.2’de ter alan iki farklı kümeye ait olan noktaları ayırmak için bir doğru çizmek gerekir. Çizilen doğru iki kümeyi de olabildiğince en doğru şekilde ayırabilmelidir.



Şekil 3.2 Farklı kümeye ait verilerin ayrıştırılması

Şekil 3.3'te yer alan b_{11} ve b_{12} arasındaki mesafe ya da marjin değerini en büyükleyen hiper düzlem ideal hiper düzlem olur. 2 boyutta doğru, 3 boyutta düzlem n boyutta ise hiper düzlem tercih edilir. Figür 4'te yer alan b_{11} ve b_{12} arasında herhangi bir veri olmadığı için bu durum hard margin olarak ifade edilir. Eğer herhangi bir veri b_{11} ve b_{12} arasında kalıyor olsaydı bu durumda ise soft margin olarak ifade edilmiş olacaktır.



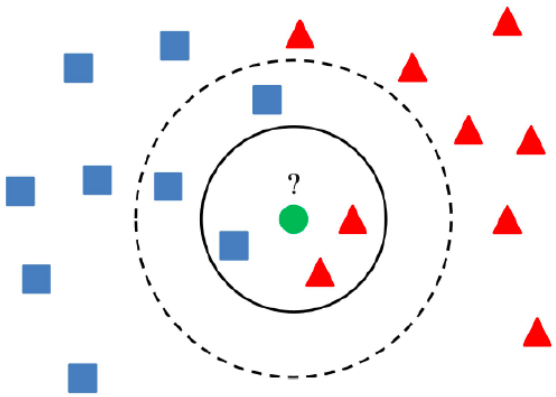
Şekil 3.3 Hard margin

K-en yakın komşu yöntemi: K-en yakın komşu yöntemi gözetimli algoritmalardan bir tanesidir ve genellikle sınıflandırma ve tahminleme problemleri için kullanılır (Bansal vd., 2022). Algoritma da yer alan "K" değeri, veri setindeki bir noktanın komşu sayısını ifade eder. Örneğin K=5 değeri bir noktanın ait olacağı sınıfı tahmin etmek için o noktaya en yakın 5 adet komşu noktasına bakılıp karar verilmesi gerektiğini belirtir. Bir noktanın ait olacağı

sınıfı doğru belirleyebilmek için "K" değerinin doğru seçilmesi çok önemlidir. "K" değerinin 1 ya da 2 gibi küçük değerler seçilmesi gürültülü sonuçlara sebep olur. Yüksek "K" değerleri ise modelin çalışması için çok fazla zamana ihtiyaç duyar ve modelin aykırı değerlere duyarlı olmasına sebep olur. 1'den 20'ye kadar her bir "K" değeri için model çalıştırıldığında "K" değeri yükseldikçe doğruluk değerinin arttığı görülür. Bunun en önemli sebeplerinden birisi de modelin ezberlenmesidir. İdeal "K" değeri genellikle 5 olarak ya da veri setindeki öz nitelik sayısı olan n sayısının karekökü yani $\sqrt{n}$ olarak seçilir. K-en yakın komşu yönteminde noktaların birbirine yakın olma durumu uzaklık değerleri ile hesaplanır. Kullanılan mesafe hesaplamalarından bazıları şu şekildedir.

1. Euclidean
2. Manhattan
3. Chebyshev
4. Hamming
5. Minkowski

Şekil 3.4'te yer alan ve yeşil ile belirtilen noktanın sınıfına karar verebilmek için "K" değeri 3 seçilirse içte yer alan çemberin içindeki 3 noktaya göre karar verilir. Eğer "K" değeri 5 seçilmiş olsaydı o zaman dışta yer alan kesikli çemberin içinde yer alan 5 noktaya bakılarak karar verilir.



Şekil 3.4 K en yakın komşu sınıflandırıcısı K=3 ve K=5

Naive bayes: Naive bayes sınıflandırıcısı tahmin etme problemlerinde kullanılan basit bir algoritmadır (Lewis, 1998). Bu algoritma istatistik temellidir (Hand ve Yu, 2001). Naive bayes algoritmasında her bir özniteliğin diğer özniteliklerden bağımsız olduğu varsayımı bulunmaktadır (Lee vd., 2011). Bu sebeple bu algoritmanın performansı genellikle lojistik regresyondan daha iyidir. Genellikle yüksek boyutlu veri setlerinde daha doğru sonuçlar verebilmektedir. Sınıf etiketlerinin olduğu bir veri seti verildiği zaman Gaussian Naive Bayes algoritması her bir test verisi için her sınıfın son olasılığı Bayes teoremini kullanarak hesaplar ve test verisini en yüksek orana sahip sınıfa atar. Kullanılan Bayes teoremi aşağıdaki formüldeki gibidir.

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$$

P(A|B): B durumunun doğru olduğu bilindiğinde A durumunun olma olasılığı
P(B|A): A durumunun doğru olduğu bilindiğinde B durumunun olma olasılığı
P(A): A durumunun olma olasılığı
P(B): B durumunun olma olasılığı

Gerçek hayatta her bir özellik hayatın bir noktasında farklı bir özellik ile bağlantılı olduğu için bu algoritmanın varsaydığı özniteliklerin birbirinden bağımsız olması bu algoritmanın olumsuz taraflarından bir tanesidir. Bu varsayımdan dolayı değişkenler arası ilişkilerde modellenemez. Naive Bayes sınıflandırıcısının 3 farklı türü bulunmaktadır. Bu türler aşağıdaki gibidir.

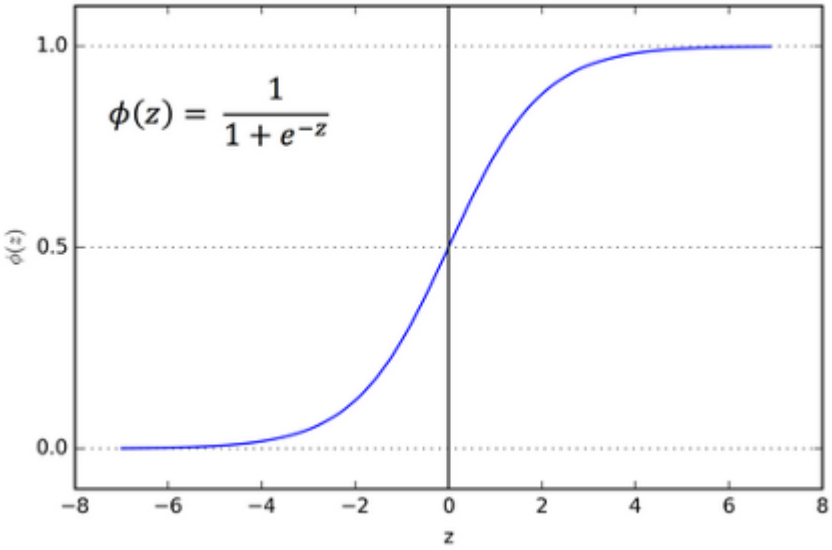
1. Gaussian naive bayes
2. Multinominal naive bayes
3. Bernoulli naive bayes

Belirtilen 3 farklı sınıflandırıcı türü veri tipinin özelliklerine göre tercih edilebilir.

Lojistik regresyon: Lojistik regresyon genellikle sınıflandırma problemleri için kullanılan algoritmalardan bir tanesidir. Bu algoritma kategorik verileri ve

sayısal verileri sınıflandırmada kullanılabilmektedir. En az sayıda değişkeni kullanılarak bağımsız ve bağımlı değişkenler arasında var olan ilişkileri açıklar. Bağımlı değişkenin sürekli olması gerektiği şeklinde bir gereklilik bulanmamaktadır. Bağımlı değişken iki ya da daha fazla nitel değere sahip ise lojistik regresyon kullanılabilir (Albert ve Lesaffre, 1986; Aldrich ve Nelson, 1984; Lemeshow ve Hosmer, 2000).

Lojistik regresyonda sigmoid (lojistik) fonksiyonu kullanılır. Bu fonksiyon Şekil 3.5'te da görüldüğü gibi S şeklinde bir görüntüye sahiptir.



Şekil 3.5 Sigmoid fonksiyonu

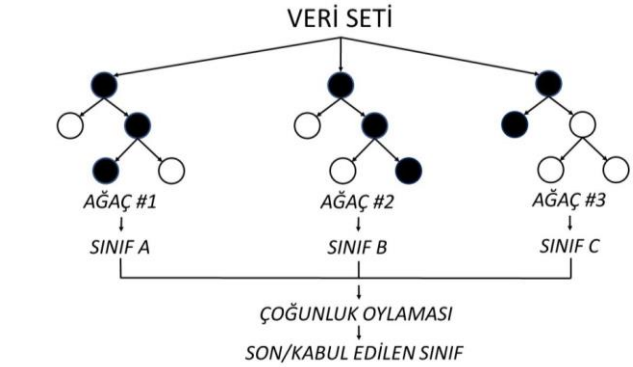
Öznitelik sayısı gözlem sayısından fazla ise lojistik regresyon kullanılmamalıdır. Eğer kullanılırsa aşırı öğrenme gerçekleşebilir. Ek olarak lojistik regresyon ile veriler arasında ayrım yapılabilmesi için veri setinin doğrusal olarak ayrılabilmesi gerekmektedir.

Rassal orman sınıflandırıcısı: Rassal orman sınıflandırıcısı Leo Breiman tarafından geliştirilen makine öğrenmesi algoritmalarından bir tanesidir. Bu algoritma her bir karar ağacının birden çok karar ağacında farklı bir gözlem örneği üzerinde eğitilerek çeşitli modeller oluşturulmasına ve sınıflandırmalar oluşturulmasına olanak tanımaktadır (Akar ve Güngör, 2012). Rassal Orman

Sınıflandırıcısında eğitme farklı veri grupları üzerinde uygulanır ve bunu sayesinde varyans değeri ve aşırı ögrenme durumunun gözlenme ihtimali düşürülmektedir. Bu durum karar ağaçlarında yer alan en büyük problemlerden bir tanesidir.

Rassal orman sınıflandırıcısında özniteliklerin ne kadar önemli olduğu ya da sonuç değişkenine ne kadar etki ettiği sonucuna ulaşılabilinmektedir. Bir özniteliğin başka bir öznitelikten daha önemli olması demek daha önemli olan özniteliğin bağımlı değişkendeki varyansın açıklanmasına o kadar fazla katkıda bulunması demektir.

Bun algoritmanın bazı olumsuz tarafları bulunmaktadır. Eğitme hızlı bir şeklide gerçekleşirken tahmin ise yavaş bir şekilde gerçekleşmektedir. Tahminlemenin daha doğru bir değer alabilmesi için yüksek sayıda ağaç gerekmektedir. Bu durumda geçen zaman artmaktadır. Verilerin seyrekliğine bakıldığında eğer veriler çok seyrek değilse sonuçların çok güvenilir olmadığı gözlemlenmektedir. Şekil 3.6'da de görüldüğü gibi Rassal Orman Sınıflandırıcısı farklı sayıda karar ağaçları kullanılarak doğru sınıf değerine ulaşılmaya çalışır.



Şekil 3.6 Rassal orman sınıflandırıcısı

XGBoost algoritması: Extreme gradient boosting algoritması olarak da bilinir. Hızlı ve güçlü bir algoritma olması ile bilinmektedir (Chen ve Guestrin, 2016). Bu algoritmada çok sayıda küçük öğrenme modelleri ya da öğrenme modeli ağaçları oluşturularak analizin yapılacağı veri seti için uygulanması en doğru model oluşturulması amaçlanmaktadır. Oluşturulan öğrenme modeli

ağaçlarından her bir tanesi önceki öğrenme modeli ağaçlarının tahminlerine dayanır ve veri setine en iyi şekilde uyum sağlamaya çalışır. Bu uyum ile beraber diğer Gradient Boosting yöntemlerine göre performans olarak daha iyi seviyedir (Nokeri, 2021).

XGBoost algoritmasının diğer bir önemli özelliği ise birden fazla CPU/GPU üzerinde paralel işlem yapabilme yeteneğidir. Bir diğer önemli özelliği ise veri madenciliği işlemleri ve problemleri için öneli olan özellikleri belirleme, haya tahminlerinin olabildiğince minimum seviyelere çekmesidir.

XGBoost algoritması öğrenme modeli ağaçlarının sahip olduğu farklı parametreleri otomatik bir şekilde optimize edebilmektedir. Bu sayede modelin performansını iyileştirmektedir.

4. VERİ SETİ, UYGULAMALAR VE SONUÇLAR

4.1 Veri Seti

Lojistik sektörlerinde özellikle elleçleme süreçlerinin uygulandığı 3. Parti lojistik şirketlerinde son kullanıcılar ya da firmalar tarafından verilen siparişlerin paketlenmesinde kullanılması gereken ideal ebatlardaki paket tipinin belirlenmesi için makine öğrenmesi algoritmalarının kullanılmasından özniteliklerin belirlenmesi ve veri setinin oluşturulması gerekmektedir. Veri seti oluşturulurken bir sipariş için belirlenebilecek her türlü özniteliğin belirlenmesi gerekir. Belirlenen özniteliklerden bazıları problem için öne arz ederken bazıları da modellerin doğruluğu negatif yönde etkileyebilmektedir. Oluşturulan veri setinden uygun olmayan özniteliklerin çıkartılmasına boyut indirgeme denmektedir. Öznitelik çıkartılması yani boyut indirgeme yerine öznitelik seçimi en çok kullanılan yöntemdir (Dubey ve Bhujade, 2021). Doğru öznitelik seçimi için ayrılan süre ile öznitelik çıkarmak için geçen süre karşılaştırıldığında, doğru öznitelik seçimi için geçen süre daha kısa sürmektedir. Öznitelik seçimi bir optimizasyon problemidir (Cai vd.,2018).

Öznitelik seçimi için farklı sınıflandırmalar mevcuttur. Bu sınıflandırmalar şu şekildedir.

- Denetimli
- Denetimsiz
- Sarmalayıcılar
- Filtreler
- Özellik sıralaması
- Altküme seçimi
- Korelasyon
- Bilgi kazanımı
- Mesafe değerine dayalı yaklaşımlar

Öznitelik seçiminin genel amacı gereksizliği, gereğinden fazla olmayı (Redundancy) ortadan kaldırmak ve uygunluğu, ilgili olma durumunu

(Relevancy) arttırmaktır. Korelasyon ve bilgi kazanımı gibi farklı metrikler kullanılarak veri setinin gereksizliği ve ilgili olma durumu kolayca açıklanabilir. Doğru öznitelik seçimi sonrası veri setindeki gereksizlik olabildiğince minimum seviyeye gelirken, ilgili olma durumu ise olabildiğince maksimum seviyeye gelmiş olacaktır. Bu işlemler sonrası değerli ve güçlü bir eğitme modeli geliştirilebiliyor, doğruluk değeri yüksek model sonuçları elde ediliyor olacaktır (Shen, 2021).

Uygun ve doğru paket tipinin belirlenmesi için bazı öznitelikler oldukça önemlidir. Bazı çalışmalarda bazı paket karakteristiklerinin oldukça önemli olduğu belirtilmiştir (Rosenthal, 2016). Bu karakteristik özellikler şunlardır.

- Hacim
- Tip (Standart ya da özel paket tipleri)
- Malzeme (Plastik ya da metal)

Yukarıda yer alan özelliklere ek olarak ebatlar ve ağırlık faktörü de paket tipi belirlenmesi için önem arz etmektedir (Boeckle, 1994; Schulz,2014).

Yapılan bu araştırmalar sonrasında bir 3.parti lojistik şirketinin paketleme verileri ile çalışmalar yapılmıştır. Ele alınan paketleme verisinde toplamda 196,059 adet sipariş bulunmaktadır. Bu siparişler içerisinde bir ya da birden fazla ürün bulundurabilmektedir. 196,059 adet siparişin içerisinde bulunan toplam ürün sayısı ise 853,534 adettir. Bazı siparişler içerisinde sadece bir adet ürün bulundurabilirken bazı siparişler ise birden fazla ürün bulundurabilmektedir. İçerisinde bir adet ürün bulunduran siparişler tekli sipariş olarak isimlendirilirken, birden fazla ürün bulunduran siparişler ise çoklu sipariş olarak nitelendirilmektedir. Bu sipariş tiplerinin dağılımları Çizelge 4.1'de gösterilmiştir.

Çizelge 4.1. Sipariş ve ürün adedi dağılımları

	Sipariş Adedi	Sipariş Oranı	Ürün Adedi	Ürün Oranı
Tekli Sipariş	43,443	22,2%	43,443	5,1%
Çoklu Sipariş	152,616	77,8%	810,091	94,9%

Çoklu siparişlerin ortalama sepet adedi incelendiğinde bu değer 5.3 olarak görünmektedir. Bu değer şunu ifade etmektedir. İçerisinde birden fazla ürünün bulunduğu siparişlerde ortalama olarak bir siparişte 5.3 ürün bulunmaktadır.
Bazı siparişler bir pakete yerleştirilebilirken bazı siparişlerdeki ürünleri bir pakete yerleştirmek mümkün olmaya bilmektedir. Bu durumun sebepleri arasında ürün sayısı, ürünlerin ebatları, kullanılan paket tiplerinin ebatları yer almaktadır. Ele alınan 196,059 adet siparişi paketlemek için toplamda 196,400 adet paket kullanılmıştır.

Siparişler için farklı paket tipleri kullanılabilmektedir. Ürünlerin ebatları ve paketlerin ebatları kullanılacak koliyi direkt olarak etkilemektedir. Kullanılan farklı paket tipleri incelendiğinde Çizelge 4.2'de yer alan adetler ve dağılımlar gözlemlenmektedir.

Çizelge 4.2. Paket tiplerine göre sipariş dağılımları

	Sipariş Adedi	Sipariş Oranı	Paket Adedi	Paket Oranı
Naylon Paket	8.123	4,1%	8.156	4,2%
Paket Tipi 1	89.247	45,5%	89.249	45,4%
Paket Tipi 2	56.094	28,6%	56.094	28,6%
Paket Tipi 3	32.483	16,6%	32.483	16,5%
Paket Tipi 4	1.940	1,0%	1.952	1,0%
Paket Tipi 5	616	0,3%	689	0,4%
Paket Tipi 6	7.261	3,7%	7.277	3,7%
Özel Paket	459	0,2%	500	0,3%
Toplam	196.059	100%	196.400	100%

Bu araştırmalar ve analizler sonrası veri seti için 12 farklı öznitelik oluşturulmuştur. Belirlenen ve oluşturulan 12 farklı öznitelik aşağıdaki gibidir.

- Toplam ürün sayısı
- Toplam farklı ürün sayısı
- Ürünlerin toplam desi değeri

- Ürünlerin toplam uzunluğu
- Ürünlerin maksimum uzunluğu
- Ürünlerin toplam genişliği
- Ürünlerin maksimum genişliği
- Ürünlerin toplam yüksekliği
- Ürünlerin maksimum yüksekliği
- Ortalama yükseklik/genişlik
- Ortalama uzunluk/yükseklik
- Ortalama uzunluk/genişlik

Belirlenmiş olan 12 öznitelik bir sipariş için uygun paket tipinin belirlenmesinde oldukça önemlidirler. Toplam ürün sayısı bir siparişte yer alan ürün sayısını ifade eder. Toplam farklı ürün sayısı ise o siparişin içerisindeki farklı ürün sayısını ifade eder. Bir kişi 3 tane aynı üründen almış ise bu değer 1 olarak gelmektedir. Her bir ürünün uzunluk, genişlik ve yükseklik değeri bulunmaktadır. Bu değerler bir ürünün desi değerinin hesaplanmasına kullanılır. Ürün ebatlarının cm cinsinden alındığı bir durumda desi değeri aşağıdaki formül ile hesaplanır.

$$Desi = (Uzunluk * Genişlik * Yükseklik)/3000$$

Siparişin içerisinde yer alan ürünlerin yükseklik değerlerinin toplamı ürünlerin toplam yükseklik değerini, genişlik değerlerinin toplamı ürünlerin toplam genişlik değerini ve uzunluk değerlerinin toplamı ürünlerin toplam uzunluk değerini vermektedir. Bu değerlerin maksimum olanları ise ürünlerin maksimum yüksekliğini, maksimum h-genişliğini ve maksimum uzunluğu vermektedir. Ortalama yükseklik/genişlik, ortalama uzunluk/yükseklik ve ortalama uzunluk/genişlik değerleri ise ürünlerin şekilleri hakkında bilgi vermektedir.

Bir siparişte 3 adet A ürünü ve 2 adet B ürünü bulunduğunu varsayalım. Bu ürünlerin ebatları cm cinsinden Çizelge 4.3'te gösterildiği gibi olsun.

Çizelge 4.3. Örnek sipariş ürün ebatları

	Yükseklik	Genişlik	Uzunluk
A	10	15	20
B	5	10	15

Bu sipariş örneği için 12 farklı öznitelik değeri Çizelge 4.4'te gösterildiği gibidir.

Çizelge 4.4. Paket tiplerine göre sipariş dağılımları

Öznitelikler	Değerler
Toplam ürün sayısı	5
Toplam farklı ürün sayısı	2
Ürünlerin toplam desi değeri	3,5
Ürünlerin toplam uzunluğu	90
Ürünlerin maksimum uzunluğu	20
Ürünlerin toplam genişliği	65
Ürünlerin maksimum genişliği	15
Ürünlerin toplam yüksekliği	40
Ürünlerin maksimum yüksekliği	10
Ortalama yükseklik/genişlik	0,62
Ortalama uzunluk/yükseklik	2,25
Ortalama uzunluk/genişlik	1,38

Veri setinde yer alan ürünler incelendiği zaman bazı ürünlerin ebat değerleri yer almadığı fark edilmiştir. Bu ürün tiplerinin yer aldığı paket tipleri veri seti için aykırı veri olarak işaretlenmiş ve çalışmanın dışında tutulmuştur. Aykırı verilerin dağılımı Çizelge 4.5'te gösterildiği gibidir.

Çizelge 4.5. Ebat kaynaklı aykırı veriler

	Paket Adedi	Paket Oranı
Aykırı Veri	561	0,3%
Aykırı Olmayan Veri	195.839	99,7%
Toplam	196.400	100%

Tablo 2'de belirtilen paket tiplerinden özel paket ve naylon paket tipleri de araştırmanın dışında tutulmuştur. Bunun sebebi ise bu paket tipleri genellikler 10 adetten fazla siparişin bir koliye ya da büyük bir poşete yerleştirilerek firmalara ya da mağazalara gönderildiği siparişlerdir. Ürün sayısından, ürün

ebatlarından bağımsız olarak seçildiği için aykırı veri olarak ele alınıp çalışmanın dışında tutulmuştur. Bu veriler Çizelge 4.6'da kırmızı ile işaretlenerek gösterilmiştir

Çizelge 4.6. Paket tipi kaynaklı aykırı veriler

	Sipariş Adedi	Sipariş Oranı	Paket Adedi	Paket Oranı
Paket Tipi 1	89.247	45,5%	89.249	45,4%
Paket Tipi 2	56.094	28,6%	56.094	28,6%
Paket Tipi 3	32.483	16,6%	32.483	16,5%
Paket Tipi 4	1.940	1,0%	1.952	1,0%
Paket Tipi 5	616	0,3%	689	0,4%
Paket Tipi 6	7.261	3,7%	7.277	3,7%
Naylon Paket	8.123	4,1%	8.156	4,2%
Özel Paket	459	0,2%	500	0,3%
Toplam	196.059	100%	196.400	100%

Belirtilen aykırı veriler çıkarıldıktan sonra makine öğrenmesi modellerinin uygulanacağı veri setinin son halinin 4 farklı örnek paketi Çizelge 4.7'de gösterilmiştir. Örneğin paket 4'te yer alan 4 adet ürün bulunmaktadır ve bu ürünler aynı ürünlerdir. Bu ürünlerin toplam yükseklik değeri 19.2 cm değerindedir. Bu ürünlerim maksimum yükseklik değeri ise 4.8 cm'dir.

Çizelge 4.7. Paket tipi kaynaklı aykırı veriler

	Paket 1	Paket 2	Paket 3	Paket 4
Ürünlerin Toplam Yüksekliği	1.50	3.50	1.70	19.20
Ürünlerin Maksimum Yüksekliği	1.50	3.50	1.70	4.80
Ürünlerin Toplam Uzunluğu	17.70	10.50	8.90	67.20
Ürünlerin Maksimum Uzunluğu	17.70	10.50	8.90	16.80
Ürünlerin Toplam Genişliği	10.00	3.50	1.70	39.60
Ürünlerin Maksimum Genişliği	10.00	3.50	1.70	9.90
Ürünlerin Toplam Desi Değeri	0.09	0.04	0.01	1.06
Toplam Ürün Sayısı	1	1	1	4
Toplam Farklı Ürün Sayısı	1	1	1	1
Ortalama Yükseklik/Genişlik	0.15	1.00	1.00	0.48
Ortalama Uzunluk/Yükseklik	11.80	3.00	5.24	3.50
Ortalama Uzunluk/Genişlik	1.77	3.00	5.24	1.70
Paket tipi	1	1	1	3

4.2 Makine Öğrenmesi Modellerinin Uyğulanması

Veri setinin son haline makine öğrenmesi modelleri uygulanmadan önce verinin bazı özellikleri incelenmiştir. İlk olarak her bir öznitelik için kaç adet boş değer ya da Null değer olduğu incelenmiştir. Çizelge 4.8'de de görüldüğü gibi bütün öznitelikler için boş değer bulunmamaktadır.

Çizelge 4.8. Özniteliklerde ve sınıf değerinde yer alan boş değer sayısı

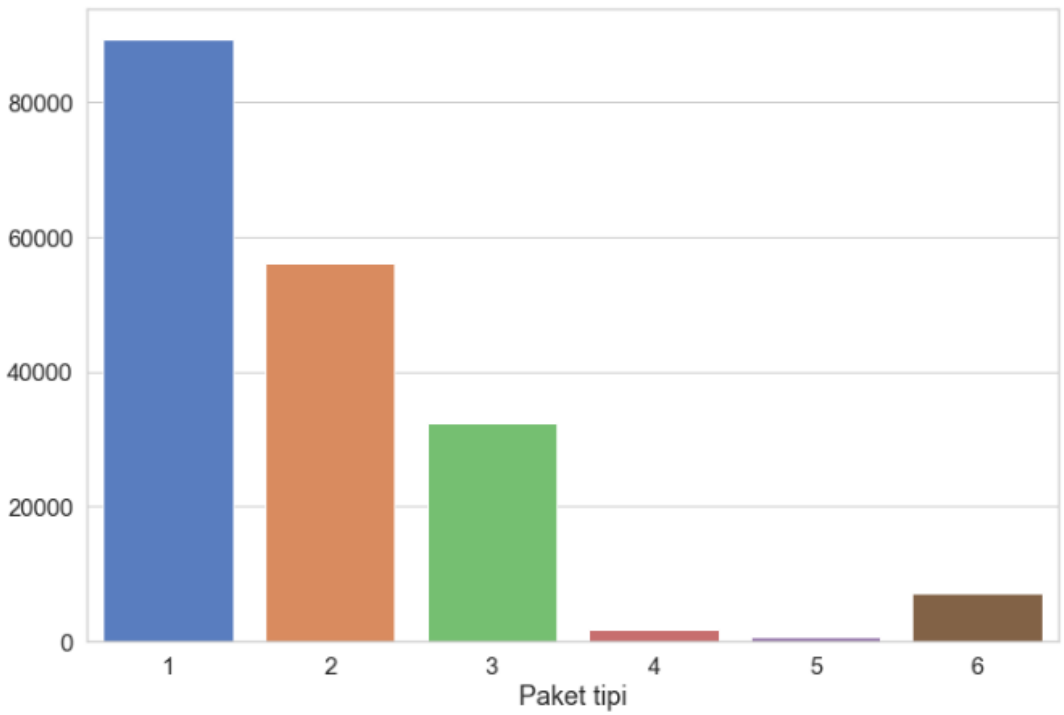
Öznitelikler	Değersiz Eleman Sayısı
Ürünlerin Toplam Yüksekliği	0
Ürünlerin Maksimum Yüksekliği	0
Ürünlerin Toplam Uzunluğu	0
Ürünlerin Maksimum Uzunluğu	0
Ürünlerin Toplam Genişliği	0
Ürünlerin Maksimum Genişliği	0
Ürünlerin Toplam Desi Değeri	0
Toplam Ürün Sayısı	0
Toplam Farklı Ürün Sayısı	0
Ortalama Yükseklik/Genişlik	0
Ortalama Uzunluk/Yükseklik	0
Ortalama Uzunluk/Genişlik	0
Paket Tipi	0

Her bir özniteliğin ve sınıf değişkenin veri tipi incelendiğinde ise Çizelge 4.9'da da görüldüğü gibi Toplam ürün sayısı, Toplam farklı ürün sayısı ve Paket tipi integer yani tam sayı iken diğer değişkenler float yani kesirli değerler içeren veri tipindedir.

Çizelge 4.9. Özniteliklerin ve sınıf değişkeninin veri tipi

Öznitelikler	Veri Tipi
Ürünlerin Toplam Yüksekliği	Float64
Ürünlerin Maksimum Yüksekliği	Float64
Ürünlerin Toplam Uzunluğu	Float64
Ürünlerin Maksimum Uzunluğu	Float64
Ürünlerin Toplam Genişliği	Float64
Ürünlerin Maksimum Genişliği	Float64
Ürünlerin Toplam Desi Değeri	Float64
Toplam Ürün Sayısı	Int
Toplam Farklı Ürün Sayısı	Int
Ortalama Yükseklik/Genişlik	Float64
Ortalama Uzunluk/Yükseklik	Float64
Ortalama Uzunluk/Genişlik	Float64
Ürünlerin Toplam Yüksekliği	Int

Her bir sınıf değişkeninde yer alan örnek sayısı dağılımı ise Şekil 4.1'de ki gibidir. En çok örnek paket tipi 1 de gözlemlenirken en az örnek ise paket tipi 5'te gözlemlenmiştir.



Şekil 4.1 Örnek sayısının sınıf değişkenlerine göre dağılım

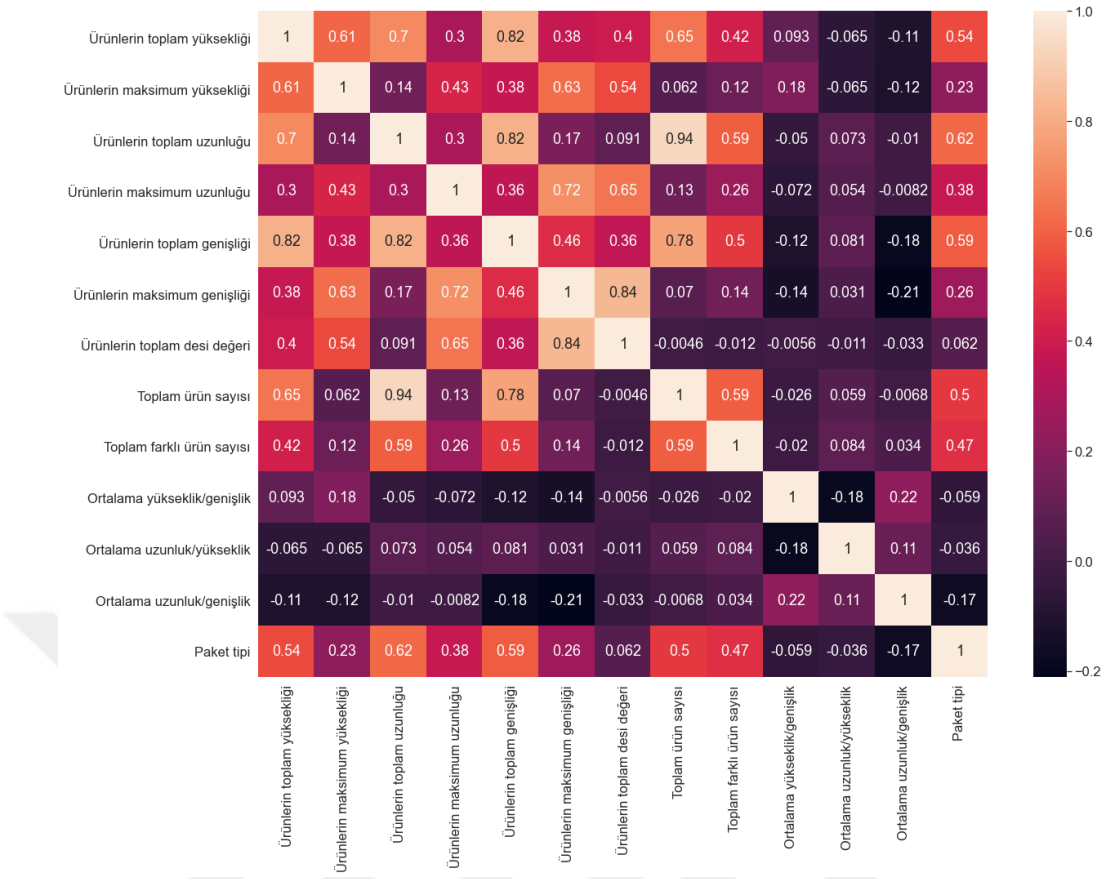
Her bir sınıf değişkeni için her bir özniteliğin ortalamasını, standart sapmasını, minimum değerini ve maksimum değerini incelediğimizde ise Çizelge 4.10'da ürünlerin toplam yüksekliği için sonuçları görmekteyiz.

Paket tipinin 1 olduğu örnekler için ürünlerin toplam yüksekliğinin ortalaması 7.09 cm, standart sapması ise 7.13 gelmektedir. Paket tipi değiştikçe bu değerlerin gözle görülebilir bir şekilde değiştiği gözlemlenmektedir. Bu durumda siparişin toplam yüksekliğine bakarak hangi paket tipine gireceği çok doğru bir yorum olmasa da kabaca söylenebilir.

Çizelge 4.10. Farklı sınıf değişkenlerine göre istatistiki değerler

Paket Tipi	Ürünlerin Toplam Yüksekliği Ortalama	Ürünlerin Toplam Yüksekliği Std	Ürünlerin Toplam Yüksekliği Min	Ürünlerin Toplam Yüksekliği Max
1	7.09	7.13	0.10	428.50
2	16.54	13.64	0.10	1680.00
3	35.66	29.68	0.30	1120.00
4	78.24	94.46	2.60	1960.00
5	161.32	148.54	5.30	2800.00
6	68.99	70.55	1.30	1680.00

Her bir özniteliğin diğer özniteliklerle olan ilişkisine ısı haritası yöntemi ile bakıp korelasyon değerlerini incelediğimiz zaman Şekil 4.2'de olan değerleri görmekteyiz.



Şekil 4.2 Korelasyon değerleri

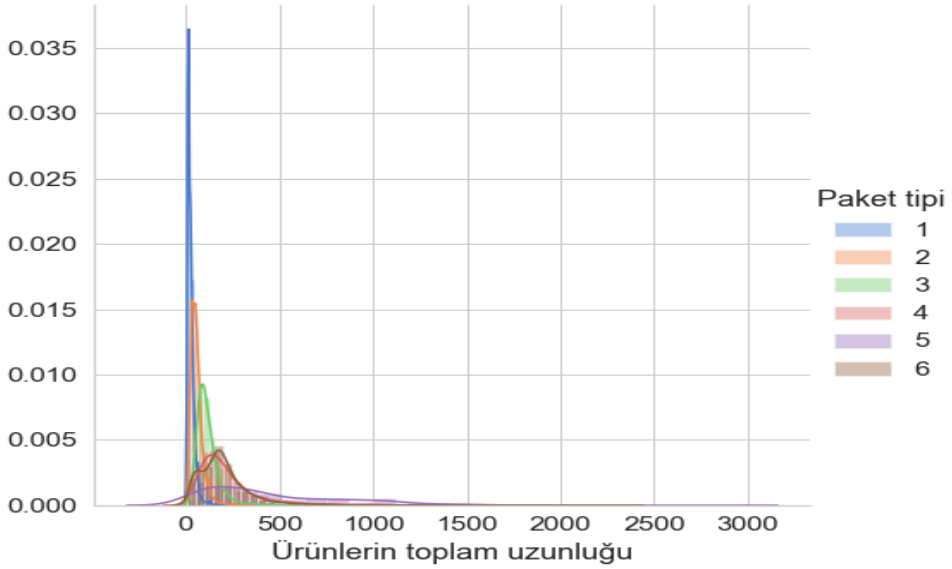
Korelasyon matrisi -1 ile 1 arasında değerler almaktadır. 1 değeri pozitif yönde yüksek korelasyon olduğunu ifade ederken, -1 değeri ise negatif yönde yüksek korelasyon olduğunu ifade eder. Korelasyon değerlerine baktığımız zaman en alt satırda yer alan Paket tipi sınıf değişkenine farklı özniteliklerin etkisi ile ilgili bilgilere ulaşılabilmektedir. Ürünlerin toplam uzunluğu 0.62 değeri ile sınıf değişkenine en çok katkıda bulunan öznitelik olduğu söylenebilir. İkinci sırada ise 0.59 değeri ile ürünlerin toplam genişliği gelmektedir. Özniteliklerin birbirleriyle olan ilişkilere baktığımız zaman ürünlerin toplam genişliği ile ürünlerin toplam yüksekliği arasında 0.82 değerinde korelasyon bulunmaktadır. Bu durumda bir örneğin toplam genişliği arttıkça toplam yüksekliğinin de artacağı söylenebilmektedir. Aynı şekilde ortalama uzunluk/genişlik ile ürünlerin maksimum genişliği arasında 0.21 değerinde negatif yönde korelasyon bulunmaktadır. Bu durumda bir örneğin maksimum genişlik değeri arttıkça ortalama uzunluk/genişlik değerinin azalacağı söylenebilmektedir. Eğer iki farklı özniteliğin korelasyon değeri 1 ya da -1 olsaydı ya da bu değerlere çok

yakın olsaydı bu 2 öznitelikten bir tanesinin çıkartılarak modellere başlanması gerekirdi. Figür 9'da yer alan değerlere göre ise herhangi bir özniteliği çıkarmaya gerek bulunmamaktadır. Her bir özniteliğin sınıf değişkenine etkisi ise Çizelge 4.11'de gösterilmiştir.

Çizelge 4.11. Özniteliklerin sınıf değişkenine etkisi

Öznitelikler	Sınıf Değişkenine Etkiler
Ürünlerin Toplam Uzunluğu	0.620733
Ürünlerin Toplam Genişliği	0.587865
Ürünlerin Toplam Yüksekliği	0.544145
Toplam Ürün Sayısı	0.495650
Toplam Farklı Ürün Sayısı	0.473098
Ürünlerin Maksimum Uzunluğu	0.383394
Ürünlerin Maksimum Genişliği	0.264461
Ürünlerin Maksimum Yüksekliği	0.228374
Ortalama Uzunluk/Genişlik	0.167055
Ürünlerin Toplam Desi Değeri	0.061700
Ortalama Yükseklik/Genişlik	0.058564
Ortalama Uzunluk/Yükseklik	0.036286

Sınıf değişkenine en çok etkisi olan ürünlerin toplam uzunluğu özniteliğinin farklı sınıf değerleri için dağılım grafiği Şekil 4.2'de gösterilmiştir.



Şekil 4.3 Farklı sınıf değerleri için ürünlerin toplam uzunluğunun dağılımı

Bu analizlerden sonra farklı makine öğrenmesi modellerinin uygulanabilmesi için veriler test ve eğitme olmak üzere ikiye ayrılmıştır. Verilerin %25'i test verisi, geri kalan %75'lik kısım ise eğitme verisi olarak tanımlanmıştır.

Veriler test ve eğitme verisi olarak iki gruba ayrıldıktan sonra standardizasyon işlemi uygulanmıştır. Eğitme verisinden ortalama ve standart sapma seçilir ve standardize edilir. Test verisi içinde eğitme verisinden elde edilen ortalama ve standart sapma kullanılır ve standardize edilir. Adil bir standardize yapmak için bu işlem uygulanır.

Bu işlemlerden sonra test ve eğitme verisi olarak 2 gruba ayrılmış ve standardize edilmiş verilere farklı makine öğrenmesi modelleri uygulanmıştır.

Lojistik regresyon: ilk olarak lojistik regresyon modeli uygulanmıştır. Bu modelde farklı "C" hiper parametre değerleri denenerek optimum değer belirlenmeye çalışılmıştır. Yüksek C değeri modele eğitim verilerine yüksek ağırlık vermesini ve karmaşıklık cezasına daha düşük ağırlık vermesini söyler. Düşük C değeri ise karmaşıklık cezasına daha fazla ağırlık vermesini söyler. Temel olarak yüksek bir C değeri eğitim verilerine çok güvenebileceğimiz anlamına gelirken, düşük bir değer ise bu verilerin gerçek dünya verilerini tam olarak temsil edemeyebileceği anlamına gelmektedir. 0 ile 2 arasında 10 farklı C değerine göre Lojistik Regresyon modellerinin doğruluk sonuçları Çizelge 4.12'deki gösterilmiştir. Bu modellerde en yüksek doğruluk oranı %79 ile C değerinin 1.3 olduğu senaryoda gelmiştir.

Çizelge 4.12. Lojistik regresyon model sonuçları

Modeller	Doğruluk Değerleri
Lojistik Regresyon (C = 1.3)	0.790385
Lojistik Regresyon (C = 1.5)	0.790385
Lojistik Regresyon (C = 1.8)	0.790256
Lojistik Regresyon (C = 0.8)	0.790192
Lojistik Regresyon (C = 1.0)	0.790192
Lojistik Regresyon (C = 0.5)	0.79015
Lojistik Regresyon (C = 2.0)	0.79015
Lojistik Regresyon (C = 0.3)	0.789722
Lojistik Regresyon (C = 0.1)	0.788675

C değerinin 1.3 olduğu senaryoda ilk 5 örneğin hangi sınıf değişkenine hangi oranda gidebileceği Çizelge 4.13'te yer almaktadır. Bu sonuçlara göre Veri 1 %90.6 oranla paket tipi 1 ile gönderilecektir. Veri tipi 3 ise %75.8 oranında Paket tipi 2 ile gönderilecektir.

Çizelge 4.13. 5 farklı örnek için sınıf oranları

	Paket Tipi 1	Paket Tipi 2	Paket Tipi 3	Paket Tipi 4	Paket Tipi 5	Paket Tipi 6
Örnek 1	90.6%	9.2%	0.2%	0.0%	0.0%	0.0%
Örnek 2	96.1%	3.9%	0.0%	0.0%	0.0%	0.0%
Örnek 3	18.9%	75.8%	5.3%	0.0%	0.0%	0.1%
Örnek 4	36.8%	57.7%	5.4%	0.0%	0.0%	0.1%
Örnek 5	92.7%	7.1%	0.1%	0.0%	0.0%	0.0%

İlk 5 örnek için tahmin edilen sınıf değişkenlerine baktığımızda ise sırasıyla 1, 1, 2, 2, 1 gelmektedir. Bu değerlerde Çizelge 4.13'ü desteklemektedir.

K-en yakın komşu sınıflandırıcısı: K-en yakın komşu sınıflandırıcısında farklı K değerleri için modeller çalıştırılmıştır. Bir noktanın hangi sınıfa ait olacağını doğru tahmin edebilmek için o noktaya en yakın kaç komşuya bakılması gerektiğini K sayısı belirler. Bu sınıflandırma yönteminde 1'den 10'a kadar K sayıları denenmiştir. Örneğin K=7 değeri bir noktanın ait olacağı sınıfı tahmin

etmek için o noktaya en yakın 7 adet komşu noktasına bakılıp karar verilmesi gerektiğini belirtir. “K” değerinin 1 ya da 2 gibi küçük değerler seçilmesi gürültülü sonuçlara sebep olur. Yüksek “K” değerleri ise modelin çalışması için çok fazla zamana ihtiyaç duyar ve modelin aykırı değerlere duyarlı olmasına sebep olur. 1’den 10’a kadar her bir “K” değeri için model çalıştırıldığında sonuçlar Çizelge 4.14’teki gibi gelmektedir. “K” değeri yükseldikçe doğruluk değerinin arttığı görülür. Bunun en önemli sebeplerinden birisi de modelin ezberlenmesidir. İdeal “K” değeri genellikle 5 olarak ya da veri setindeki öz nitelik sayısı olan n sayısının karekökü yani $\sqrt{n}$ olarak seçilir. Bu durumda K değerinin 4 ya da 5 olduğu durumda doğruluk oranı yaklaşık olarak %80 gelmektedir.

Çizelge 4.14. K-en yakın komşu model sonuçları

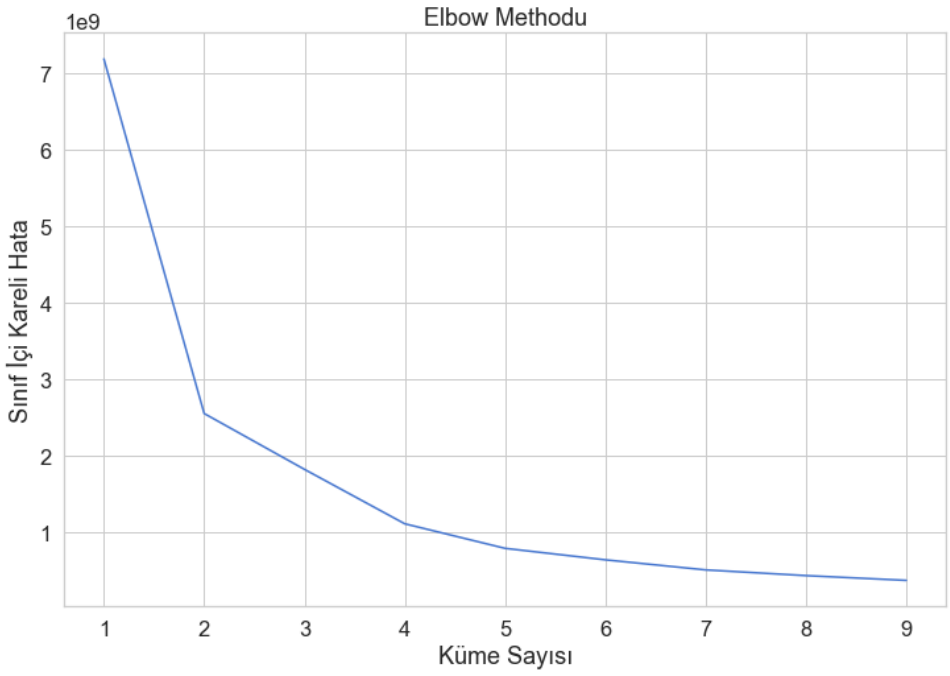
Modeller	Doğruluk Değerleri
K-en Yakın Komşu K=10	0.809145
K-en Yakın Komşu K=9	0.807863
K-en Yakın Komşu K=8	0.807372
K-en Yakın Komşu K=7	0.80515
K-en Yakın Komşu K=6	0.803291
K-en Yakın Komşu K=5	0.800427
K-en Yakın Komşu K=4	0.798098
K-en Yakın Komşu K=3	0.790128
K-en Yakın Komşu K=2	0.768526
K-en Yakın Komşu K=1	0.75391

Bu iki farklı K değerine ek olarak Elbow metodu ile k değerinin ne olması gerektiği de çıkarım olarak ulaşılabilir. Elbow metodunda her bir K değerine göre noktaların küme merkezine olan uzaklıklarının karelerinin toplamı hesaplanır. Bu değerler sınıf içi kareli hata olarak ifade edilir ve bunlar bir grafik üzerinde gösterilir. Grafiğe bakarak sınıf içi kareli hatalar arasındaki farkın azalmaya başladığı noktaya uygun K değeri olarak belirlenmiş olur. Bu noktaya dirsek noktası da denir. Çizelge 4.15’te farklı küme değerleri için sınıf içi kareli hata değerlerinin toplamı gösterilmiştir.

Çizelge 4.15. Sınıf içi kareli hata toplamları

Küme Sayısı	Sınıf İçi Kareli Hata Toplamı
Küme Sayısı = 1	7,189,266,747
Küme Sayısı = 2	2,548,692,465
Küme Sayısı = 3	1,817,679,195
Küme Sayısı = 4	1,105,526,018
Küme Sayısı = 5	784,053,666
Küme Sayısı = 6	633,674,873
Küme Sayısı = 7	501,918,449
Küme Sayısı = 8	427,208,885
Küme Sayısı = 9	364,870,518

Çizelge 4.15'te belirtilen değerlerin grafik üzerinde gösterimi Şekil 4.4'teki gibidir. Şekil 4.4'te de görüldüğü gibi K değerinin 4 ya da 5 olduğu noktada sınıf içi kareli hatalar arasındaki fark azalmaktadır.



Şekil 4.4 Sınıf içi kareli hata değerleri

K değerinin 5 olduğu senaryoda ilk 5 örneğin hangi sınıf değişkenine hangi oranda gidebileceği Çizelge 4.16'da yer almaktadır. Bu sonuçlara göre Veri 1

%100 oranla paket tipi 1 ile gönderilecektir. Veri tipi 3 ise %60 oranında Paket tipi 2 ile gönderilecektir.

Çizelge 4.16. Farklı örnekler için sınıf oranları

	Paket Tipi 1	Paket Tipi 2	Paket Tipi 3	Paket Tipi 4	Paket Tipi 5	Paket Tipi 6
Veri 1	100%	0%	0%	0%	0%	0%
Veri 2	100%	0%	0%	0%	0%	0%
Veri 3	40%	60%	0%	0%	0%	0%
Veri 4	20%	80%	0%	0%	0%	0%
Veri 5	100%	0%	0%	0%	0%	0%

İlk 5 örnek için tahmin edilen sınıf değişkenlerine baktığımızda ise sırasıyla 1, 1, 2, 2, 1 gelmektedir. Bu değerlerde Tablo 15'i desteklemektedir.

Karar ağaçları: Karar ağaçları makine öğrenmesi modelinde herhangi bir farklı parametre kullanılmamış, modelin varsayılan parametreleri üzerinden ilerlenmiştir. Bu model uygulandığı zaman doğruluk oranı Çizelge 4.17'de gösterildiği gibi %78 değerini almaktadır. Lojistik Regresyon ve K-en Yakın Komşu Sınıflandırıcısı ile karşılaştırıldığında doğruluk oranı bir miktar daha düşük gelmektedir.

Çizelge 4.17. Karar ağacı model doğruluk oranı

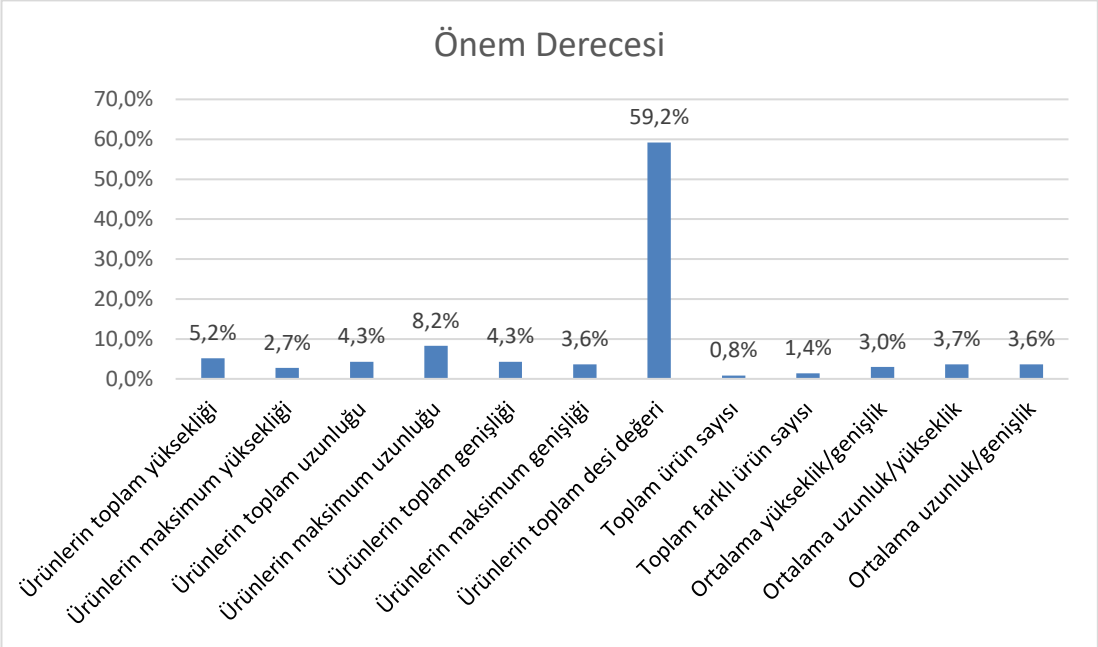
Modeller	**Doğruluk Değerleri**
Karar Ağacı	0.778611

Karar Ağacı modelinde ilk 5 örneğin hangi sınıf değişkenine hangi oranda gidebileceği Tablo 17'de yer almaktadır. Bu sonuçlara göre Veri 1 %98 oranla paket tipi 1'e, veri 2 %97 oranında paket tipi 1'e gönderilirken, diğer 3 veri ise %100 oranında paket tipi 1'e gönderilecektir. İlk 5 örnek için tahmin edilen sınıf değişkenlerine baktığımızda ise sırasıyla 1, 1, 1, 1, 1 gelmektedir. Bu değerlerde Çizelge 4.18'i desteklemektedir. Lojistik regresyonda ve K-en Yakın Komşu Sınıflandırıcısında veri 3 ve 4 paket tipi 1'e gönderilirken, Karar Ağacında ise bu veriler paket tipi 1'e gönderilmiştir.

Çizelge 4.18. *5* farklı örnek için sınıf oranları

	Paket Tipi 1	Paket Tipi 2	Paket Tipi 3	Paket Tipi 4	Paket Tipi 5	Paket Tipi 6
Veri 1	98%	2%	0%	0%	0%	0%
Veri 2	97%	3%	0%	0%	0%	0%
Veri 3	100%	0%	0%	0%	0%	0%
Veri 4	100%	0%	0%	0%	0%	0%
Veri 5	100%	0%	0%	0%	0%	0%

Karar Ağaçlarında özniteliklerin sınıf değişkenine ne oranda etki ettiği görülebilmektedir. Şekil 4.5'te de görüldüğü gibi bu veri setinde sınıf değişkenine en çok etki eden öznitelik ürünlerin toplam desi değeridir. Bu değişken sınıf değişkenine %60 oranında etki etmektedir.



Şekil 4.5 Özniteliklerin önem dereceleri

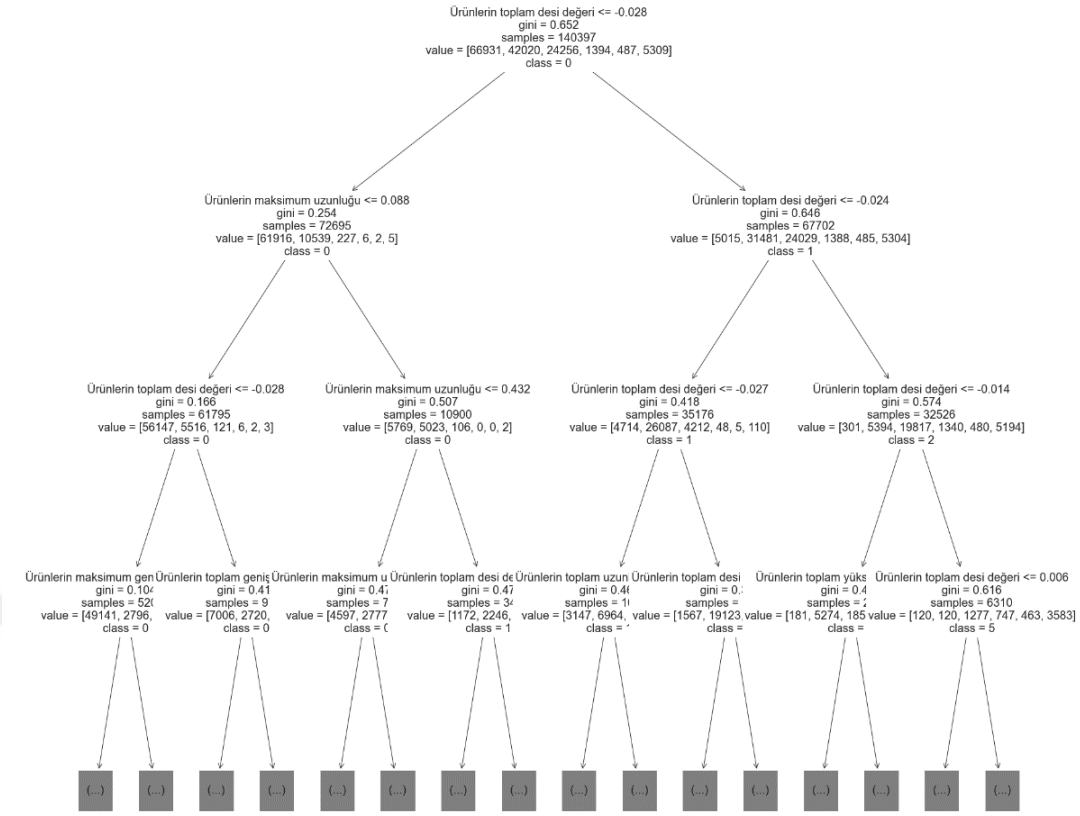
Bu modelin bir metrik göstergesi olan hata matrisi incelendiğinde Çizelge 4.19'daki sonuçlar gözlemlenmektedir. Hata matrisinde sütunlar tahmin edilen sınıfları gösterirken, kolonlar ise gerçek sınıfları göstermektedir. Örneğin 2,300 adet paket tipinin gerçek tipi 2 iken model tarafından 1 olarak tahmin edilmiştir. Bu tabloda doğru tahmin edilenlerin toplam adete oranı doğruluk oranı olan %78 değerini vermektedir. Paket tiplerinin numaraları büyüklükleri

ile ilgili bir fikir verseydi bu durumda doğru tahmine dilen paket tiplerinin bir sağı ve bir solundaki tahminlerde doğruluk oranına eklenebilirdi.

Çizelge 4.19. Hata matrisi

Hata Matrisi	Paket Tipi 1	Paket Tipi 2	Paket Tipi 3	Paket Tipi 4	Paket Tipi 5	Paket Tipi 6
Paket Tipi 1	19,883	2,300	110	7	3	8
Paket Tipi 2	2,668	9,618	1,664	31	6	20
Paket Tipi 3	113	1,858	5,520	196	5	394
Paket Tipi 4	3	26	171	107	37	121
Paket Tipi 5	1	3	8	34	84	32
Paket Tipi 6	4	27	440	149	24	1,125

Karar ağaçlarının önemli yapılarından bir tanesi de yapraklardır. Karar Ağacında kararlar verilirken bazı dönüm noktaları bulunmaktadır. Bu dönüm noktaları Şekil 4.6'da gösterilmiştir. Bu figürden de ürünlerin toplam desi değerinin en önemli karar verme özniteliği olduğu söylenebilmektedir.



Şekil 4.6 Karar ağacı dönüm noktaları

Destek vektör makineleri: Destek vektör makineleri vektör tabanlı bir algoritmadır. Her bir veri arasında vektörel uzaklıklar hesaplanarak sınıflandırma problemleri çözülür. Vektör tabanlı bir hesaplama yöntemi içerdiği için çalışma süresi diğer yöntemlere göre daha uzun sürebilmektedir. Bu algoritmalar karmaşık olan ama veri miktarı bakımından orta seviyede olan veri setleri için ideal yöntemlerden bir tanesidir. Destek Vektör Makinelerinde farklı Kernel metotları bulunmaktadır. Bu metotlar “Radial”, “Linear” ve “Poly” olarak geçmektedir. İdeal Kernel metodu için 3 farklı metot denenmeli ve doğruluk oranı kontrol edilmelidir. 3 farklı Kernel metoduna göre elde edilen doğruluk oranları incelendiğinde Çizelge 4.20’de görüldüğü gibi en yüksek doğruluk oranı 0.81 ile “Radial” metot ile elde edilmiştir. 2. Sırada 0.79 ile “Linear” gelirken en sonda ise 0.78 ile “Polynomial” metodu gelmiştir.

Çizelge 4.20. Destek vektör makineleri model sonuçları

Modeller	Doğruluk Değerleri
Destek Vektör Makineleri (Radial)	0.812372
Destek Vektör Makineleri (Linear)	0.794124
Destek Vektör Makineleri (Polynomial)	0.738889

Kernel metodunun "Radial" olduğu senaryoda ilk 5 örneğin hangi sınıf değişkenine hangi oranda gidebileceği Çizelge 4.21'de yer almaktadır. Bu sonuçlara göre Veri 1 %95.9 oranla paket tipi 1 ile gönderilecektir. Veri tipi 3 ise %87.3 oranında Paket tipi 2 ile gönderilecektir.

Çizelge 4.21. 5 farklı örnek için sınıf oranları

	Paket Tipi 1	Paket Tipi 2	Paket Tipi 3	Paket Tipi 4	Paket Tipi 5	Paket Tipi 6
Veri 1	95.9%	3.9%	0.1%	0.0%	0.0%	0.0%
Veri 2	92.5%	7.0%	0.4%	0.1%	0.0%	0.0%
Veri 3	10.1%	87.3%	2.3%	0.3%	0.0%	0.0%
Veri 4	43.5%	54.5%	1.8%	0.2%	0.0%	0.0%
Veri 5	95.8%	4.0%	0.1%	0.0%	0.0%	0.0%

İlk 5 örnek için tahmin edilen sınıf değişkenlerine baktığımızda ise sırasıyla 1, 1, 2, 2, 1 gelmektedir. Bu değerlerde Çizelge 4.21'i desteklemektedir.

Gaussian naive bayes sınıflandırıcıları: Bu algoritma istatistik temellidir. Gaussian Naive Bayes algoritmasında her bir özniteliğin diğer özniteliklerden bağımsız olduğu varsayımı bulunmaktadır. Genellikle yüksek boyutlu veri setlerinde daha doğru sonuçlar verebilmektedir. Sınıf etiketlerinin olduğu bir veri seti verildiği zaman Naive Bayes algoritmaları her bir test verisi için her sınıfın son olasılığı Bayes teoremini kullanarak hesaplar ve test verisini en yüksek orana sahip sınıfa atar. Gaussian naive bayes sınıflandırıcısının doğruluk değeri Çizelge 4.22'de gösterilmiştir. Doğruluk oranı diğer modellere göre 0.672799 değeri ile daha düşük gelmiştir. Özniteliklerin birbirinden tamamen bağımsız olmaması bu durumu açıklayan en önemli etmenlerden bir tanesidir.

Çizelge 4.22. Naive bayes model sonuçları

Modeller	Doğruluk Değerleri
Gaussian Naive Bayes	0.672799

Gaussian naive bayes metodunda ilk 5 örneğin hangi sınıf değişkenine hangi oranda gidebileceği Çizelge 4.23'te yer almaktadır. Bu sonuçlara göre Veri 1 %99.5 oranla paket tipi 1 ile gönderilecektir. Veri tipi 3 ise %59.9 oranında Paket tipi 2 ile gönderilecektir.

Çizelge 4.23. 5 farklı örnek için sınıf oranları

	Paket Tipi 1	Paket Tipi 2	Paket Tipi 3	Paket Tipi 4	Paket Tipi 5	Paket Tipi 6
Veri 1	99.5%	0.5%	0.0%	0.0%	0.0%	0.0%
Veri 2	100.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Veri 3	40.0%	59.9%	0.1%	0.0%	0.0%	0.0%
Veri 4	73.3%	26.7%	0.0%	0.0%	0.0%	0.0%
Veri 5	99.4%	0.6%	0.0%	0.0%	0.0%	0.0%

İlk 5 örnek için tahmin edilen sınıf değişkenlerine baktığımızda ise sırasıyla 1, 1, 2, 1, 1 gelmektedir. Bu değerlerde Çizelge 4.23'ü desteklemektedir.

Rassal orman sınıflandırıcısı: Bu algoritma her bir karar ağacının birden çok karar ağacında farklı bir gözlem örneği üzerinde eğitilerek çeşitli modeller oluşturulmasına ve sınıflandırmalar oluşturulmasına olanak tanımaktadır. Rassal Orman Sınıflandırıcısında eğitme farklı veri grupları üzerinde uygulanır ve bunu sayesinde varyans değeri ve aşırı ögrenme durumunun gözlenme ihtimali düşürülmektedir. Bu durum karar ağaçlarında yer alan en büyük problemlerden bir tanesidir. Bu modelin doğruluk oranı Çizelge 4.24'te gösterildiği gibi 0.8338 değerindedir. Bu değer önceki modellere göre daha yüksek doğruluk oranıdır.

Çizelge 4.24. Rassal orman model sonuçları

Modeller	Doğruluk Değerleri
Rassal Orman Sınıflandırıcısı	0.833846

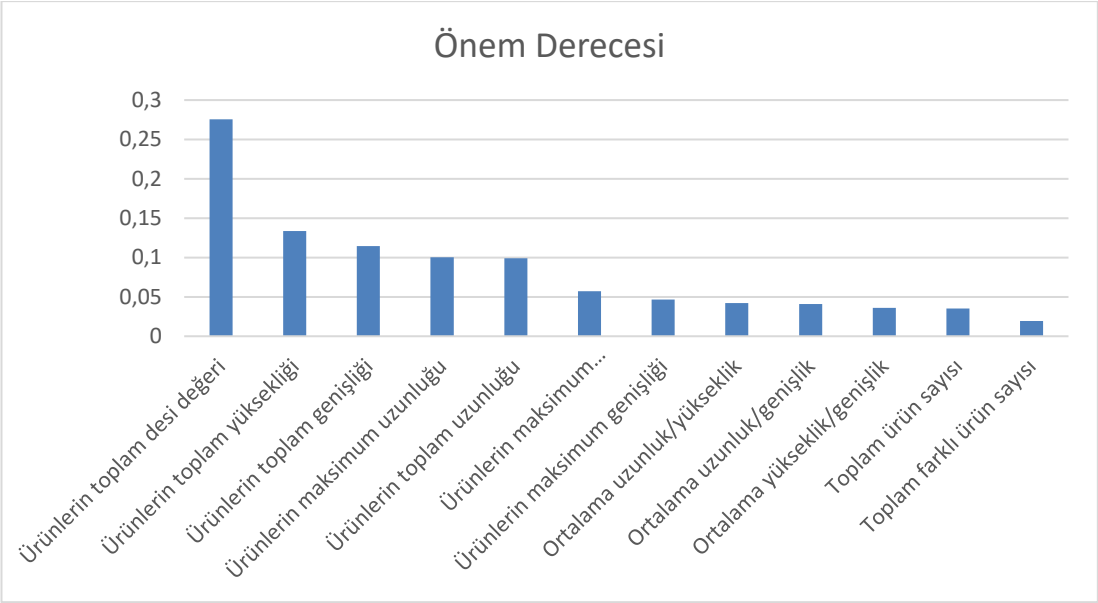
Rassal orman metodunda ilk 5 örneğin hangi sınıf değişkenine hangi oranda gidebileceği Çizelge 4.25'te yer almaktadır. Bu sonuçlara göre Veri 1 %98 oranla paket tipi 1 ile gönderilecektir. Veri tipi 3 ise %81 oranında Paket tipi 2 ile gönderilecektir.

Çizelge 4.25. 5 farklı örnek için sınıf oranları

	Paket Tipi 1	Paket Tipi 2	Paket Tipi 3	Paket Tipi 4	Paket Tipi 5	Paket Tipi 6
Veri 1	98%	2%	0%	0%	0%	0%
Veri 2	97%	3%	0%	0%	0%	0%
Veri 3	13%	81%	6%	0%	0%	0%
Veri 4	52%	48%	0%	0%	0%	0%
Veri 5	100%	0%	0%	0%	0%	0%

İlk 5 örnek için tahmin edilen sınıf değişkenlerine baktığımızda ise sırasıyla 1, 1, 2, 1, 1 gelmektedir. Bu değerlerde Çizelge 4.25'i desteklemektedir.

Rassal orman sınıflandırıcısında özniteliklerin sınıf değişkenine ne oranda etki ettiği görülebilmektedir. Şekil 4.7'de de görüldüğü gibi bu veri setinde sınıf değişkenine en çok etki eden öznitelik ürünlerin toplam desi değeridir. Bu değişken sınıf değişkenine %27 oranında etki etmektedir.



Şekil 4.7 Özniteliklerin önem dereceleri

Bu modelin bir metrik göstergesi olan hata matrisi incelendiğinde Çizelge 4.26'daki sonuçlar gözlemlenmektedir. Hata matrisinde sütunlar tahmin edilen sınıfları gösterirken, kolonlar ise gerçek sınıfları göstermektedir. Örneğin 1,793 adet paket tipinin gerçek tipi 2 iken model tarafından 1 olarak tahmin edilmiştir. Bu tabloda doğru tahmin edilenlerin toplam adete oranı doğruluk oranı olan %83 değerini vermektedir. Paket tiplerinin numaraları büyüklükleri ile ilgili bir fikir verseydi bu durumda doğru tahmin edilen paket tiplerinin bir sağı ve bir solundaki tahminlerde doğruluk oranına eklenebilirdi.

Çizelge 4.26. Hata matrisi

Hata Matrisi	Paket Tipi 1	Paket Tipi 2	Paket Tipi 3	Paket Tipi 4	Paket Tipi 5	Paket Tipi 6
Paket Tipi 1	20,485	1,793	48	4	2	9
Paket Tipi 2	2,115	10,721	1,163	-	3	5
Paket Tipi 3	38	1,439	6,337	11	1	260
Paket Tipi 4	2	14	191	69	35	154
Paket Tipi 5	1	2	3	33	92	31
Paket Tipi 6	-	12	391	36	10	1,320

XGBoost sınıflandırıcısı: Bu algoritmada çok sayıda küçük öğrenme modelleri ya da öğrenme modeli ağaçları oluşturularak analizin yapılacağı veri seti için uygulanması en doğru model oluşturulması amaçlanmaktadır. Oluşturulan öğrenme modeli ağaçlarından her bir tanesi önceki öğrenme modeli ağaçlarının tahminlerine dayanır ve veri setine en iyi şekilde uyum sağlamaya çalışır. Bu modelin doğruluk oranı Çizelge 4.27'de gösterildiği gibi 0.8338 değerini almaktadır. Bu değer bütün modellere göre daha yüksek doğruluk oranıdır.

Çizelge 4.27. XGBoost model sonuçları

Modeller	**Doğruluk Değerleri**
XGBoost Sınıflandırıcısı	0.838803

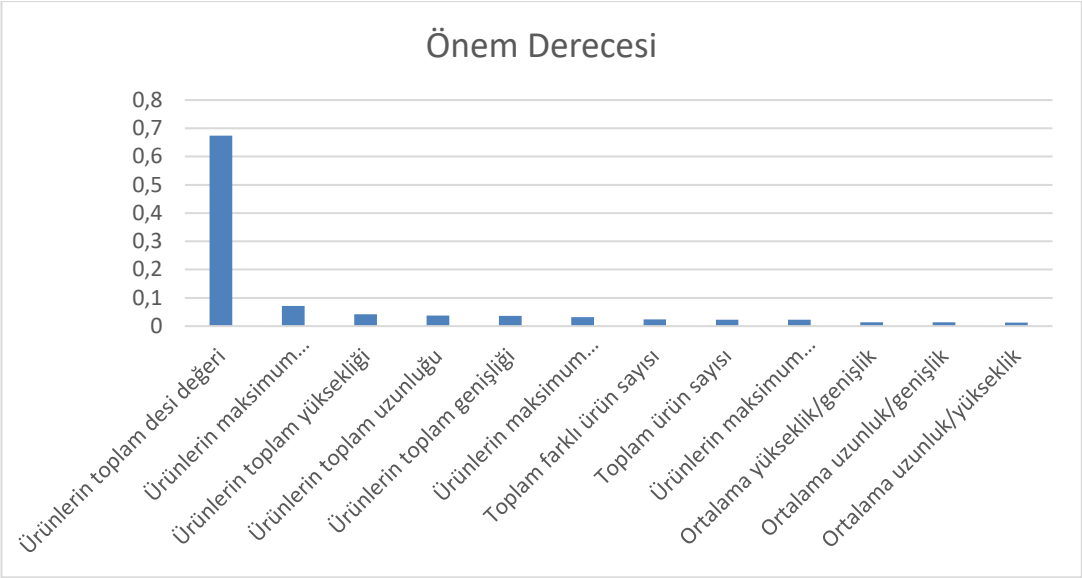
XGBoost metodunda ilk 5 örneğin hangi sınıf değişkenine hangi oranda gidebileceği Çizelge 4.28'de yer almaktadır. Bu sonuçlara göre Veri 1 %97.9 oranla paket tipi 1 ile gönderilecektir. Veri tipi 3 ise %94.6 oranında Paket tipi 2 ile gönderilecektir.

Çizelge 4.28. 5 farklı örnek için sınıf oranları

	Paket Tipi 1	Paket Tipi 2	Paket Tipi 3	Paket Tipi 4	Paket Tipi 5	Paket Tipi 6
Veri 1	97.9%	2.1%	0.0%	0.0%	0.0%	0.0%
Veri 2	98.6%	1.4%	0.0%	0.0%	0.0%	0.0%
Veri 3	3.6%	94.6%	1.8%	0.0%	0.0%	0.0%
Veri 4	62.2%	37.4%	0.4%	0.0%	0.0%	0.0%
Veri 5	97.7%	2.3%	0.0%	0.0%	0.0%	0.0%

İlk 5 örnek için tahmin edilen sınıf değişkenlerine baktığımızda ise sırasıyla 1, 1, 2, 1, 1 gelmektedir. Bu değerlerde Çizelge 28'i desteklemektedir.

XGBoost sınıflandırıcısında özniteliklerin sınıf değişkenine ne oranda etki ettiği görülebilmektedir. Şekil 4.8'de de görüldüğü gibi bu veri setinde sınıf değişkenine en çok etki eden öznitelik ürünlerin toplam desi değeridir. Bu değişken sınıf değişkenine %67 oranında etki etmektedir. Bu öznitelik diğer özniteliklere göre oldukça öne çıkmaktadır.



Şekil 4.8 Özniteliklerin önem dereceleri

Bu modelin bir metrik göstergesi olan hata matrisi incelendiğinde Çizelge 4.29'daki sonuçlar gözlemlenmektedir. Hata matrisinde sütunlar tahmin edilen sınıfları gösterirken, kolonlar ise gerçek sınıfları göstermektedir. Örneğin 1,642 adet paket tipinin gerçek tipi 2 iken model tarafından 1 olarak tahmin edilmiştir. Bu tabloda doğru tahmin edilenlerin toplam adete oranı doğruluk oranı olan %83 değerini vermektedir. Paket tiplerinin numaraları büyüklükleri ile ilgili bir fikir verseydi bu durumda doğru tahmin edilen paket tiplerinin bir sağı ve bir solundaki tahminlerde doğruluk oranına eklenebilirdi.

Çizelge 4.29. Hata matrisi

Hata Matrisi	Paket Tipi 1	Paket Tipi 2	Paket Tipi 3	Paket Tipi 4	Paket Tipi 5	Paket Tipi 6
Paket Tipi 1	20,607	1,642	50	3	2	7
Paket Tipi 2	2,096	10,791	1,113	-	3	4
Paket Tipi 3	38	1,423	6,352	1	-	272
Paket Tipi 4	2	14	190	75	29	155
Paket Tipi 5	1	2	4	28	97	30
Paket Tipi 6	-	8	377	38	12	1,334

7 farklı makine öğrenmesi modelinin uygulanması sonucu elde edilen doğruluk değerleri karşılaştırıldığı zaman Çizelge 4.30'da elde edilen sonuçlar gözlemlenmektedir. Sonuçlar incelendiği zaman doğruluk oranı en yüksek olan model 0.8388 ile XGBoost sınıflandırıcısı olurken, ikinci sırada ise 0.8338 doğruluk değeri ile Rassal orman sınıflandırıcısı gelmektedir. Destek vektör makinelerinden Kernel metodu "Radial" olan metot ise K-en yakın komşu sınıflandırıcılarına göre daha iyi sonuç verdiği gözlemlenmektedir. Bu modeller arasından doğruluk oranı en düşük olan model ise 0.6727 doğruluk değeri ile Gaussian naive bayes sınıflandırıcısıdır. Özniteliklerin birbirinden tamamen bağımsız olduğunu varsayması bu modelin doğruluk değerinin düşük çıkmasındaki en önemli sebeplerdendir.

Çizelge 4.30. Modellerin karşılaştırılması

Modeller	Doğruluk Değerleri
XGBoost Sınıflandırıcısı	0.838803
Rassal Orman Sınıflandırıcısı	0.833846
Destek Vektör Makineleri (Radial)	0.812372
K-en Yakın Komşu K=5	0.800427
K-en Yakın Komşu K=4	0.798098
Destek Vektör Makineleri (Linear)	0.794124
Lojistik Regresyon (C = 1.3)	0.790385
Lojistik Regresyon (C = 1.5)	0.790385
Karar Ağacı	0.778611
Destek Vektör Makineleri (Polynomial)	0.738889
Gaussian Naive Bayes	0.672799

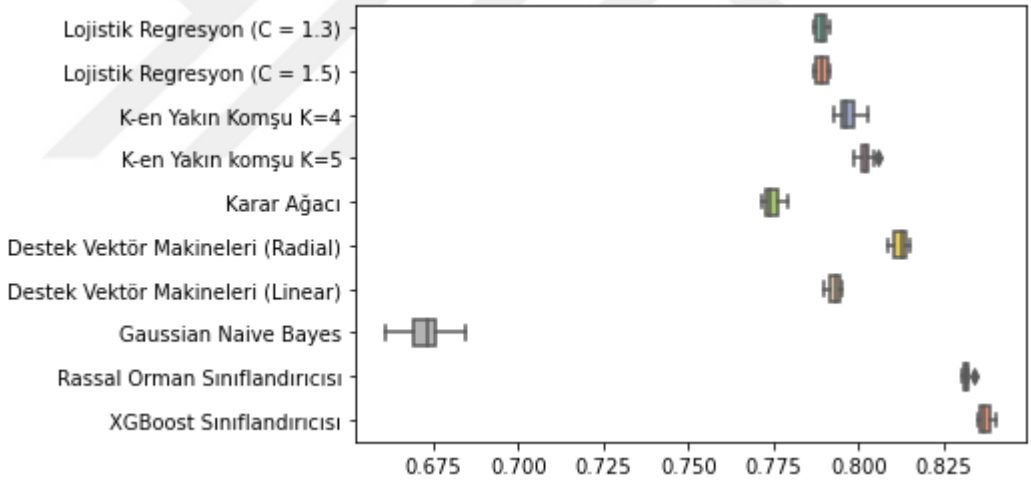
Çizelge 4.30'da elde edilen sonuçlar çapraz geçerleme yapılmadan elde edilen sonuçları göstermektedir. Çapraz geçerleme makine öğrenmesi modelleri için önem taşımaktadır. Çapraz geçerlemede bir istatiksel analizin ya da makine öğrenmesi modellerinin bağımsız bir veri setinde nasıl bir sonuç vereceğini gösteren bir model doğrulama tekniğidir. Veri seti belirlenen k sayısına göre eşit bölümlere ayrılır ve her bir bölümün eğitme ve test verileri için kullanılması sağlanır. Bu şekilde elde edilecek hata minimize edilmeye çalışılır. Bir modeli k sayısı kadar eğitmek ve test etmenin doğruluk değeri açısından faydası varken, eğitme süresi açısından ise zaman alıcı bir tarafı bulunmaktadır. Küçük ve orta boyutlu veri setlerinde bunu çapraz geçerleme uygulamak çok fazla zaman almaz iken büyük veri setlerinde hesaplama ve zaman açısından oldukça maliyetli olabilmektedir.

Çizelge 4.31'de 3 kat çapraz geçerleme yapıldığı senaryoda elde edilen sonuçlar yer almaktadır. Elde edilen sonuçlara göre XGBoost sınıflandırıcısı 0.8364 değeri ile en iyi doğruluk değerine sahip iken, ikinci sırada ise 0.8303 doğruluk değeri ile Rassal orman sınıflandırıcısı gelmektedir. En az doğruluk değerine sahip model ise 0.6653 değeri ile gaussian naive bayes sınıflandırıcısıdır. En az doğruluk değerine sahip kinci sıradaki model ise 0.7721 değeri ile Karar Ağacı modelidir. Şekil 4.9'da ise bu modellerin doğruluk değerlerine göre kutu grafiği

gösterimi yer almaktadır. Belirgin bir şekilde Gaussian naive bayes modeli diğer modellerden doğruluk değeri olarak ayrışmaktadır.

Çizelge 4.31. Modellerin karşılaştırılması (3 kat çapraz geçerleme)

Modeller	Doğruluk Değerleri
XGBoost Sınıflandırıcısı	0.836493
Rassal Orman Sınıflandırıcısı	0.830366
Destek Vektör Makineleri (Radial)	0.811194
K-en Yakın Komşu K=5	0.798661
K-en Yakın Komşu K=4	0.794276
Destek Vektör Makineleri (Linear)	0.792865
Lojistik Regresyon (C = 1.3)	0.789313
Lojistik Regresyon (C = 1.5)	0.789131
Karar Ağacı	0.772144
Gaussian Naive Bayes	0.665369



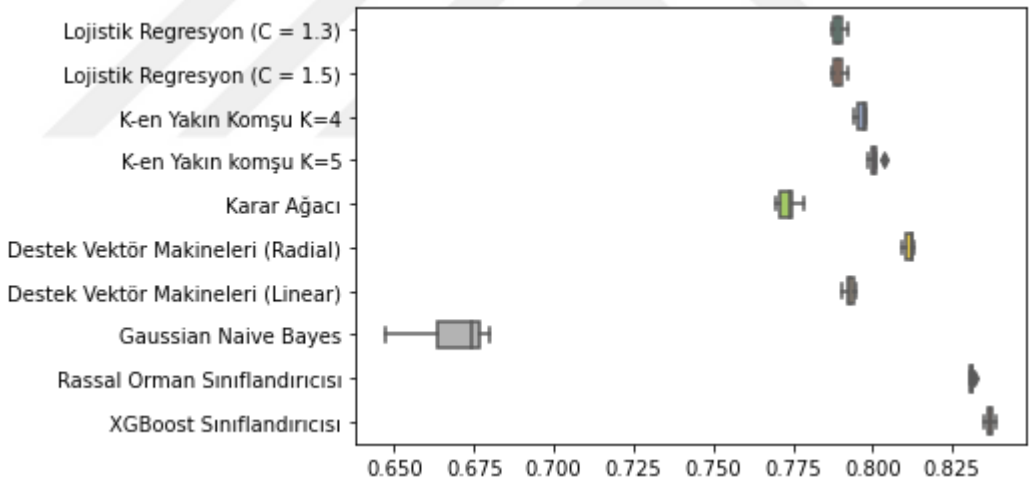
Şekil 4.9 Modellerin karşılaştırılması (3 Kat Çapraz Geçerleme)

Çizelge 4.32'de 5 kat çapraz geçerleme yapıldığı senaryoda elde edilen sonuçlar yer almaktadır. Elde edilen sonuçlara göre XGBoost sınıflandırıcısı 0.8368 değeri ile en iyi doğruluk değerine sahip iken, ikinci sırada ise 0.8310 doğruluk değeri ile Rassal orman sınıflandırıcısı gelmektedir. En az doğruluk değerine sahip model ise 0.6682 değeri ile Gaussian naive bayes Sınıflandırıcısıdır. En az doğruluk değerine sahip kinci sıradaki model ise 0.7738 değeri ile Karar ağacı modelidir. Şekil 4.10'da ise bu modellerin doğruluk değerlerine göre kutu grafiği

gösterimi yer almaktadır. Belirgin bir şekilde Gaussian naive bayes modeli diğer modellerden doğruluk değeri olarak ayrışmaktadır.

Çizelge 4.32. Modellerin karşılaştırılması (5 kat çapraz geçerleme)

Modeller	Doğruluk Değerleri
XGBoost Sınıflandırıcısı	0.836862
Rassal Orman Sınıflandırıcısı	0.831087
Destek Vektör Makineleri (Radial)	0.811466
K-en Yakın Komşu K=5	0.80067
K-en Yakın Komşu K=4	0.796428
Destek Vektör Makineleri (Linear)	0.792833
Lojistik Regresyon (C = 1.5)	0.789302
Lojistik Regresyon (C = 1.3)	0.78927
Karar Ağacı	0.773821
Gaussian Naive Bayes	0.668269



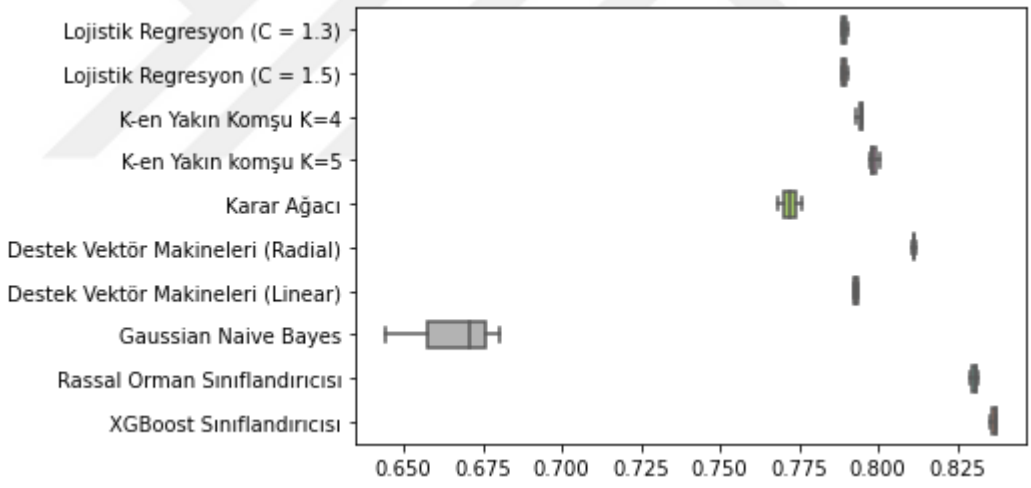
Şekil 4.10 Modellerin karşılaştırılması (5 kat çapraz geçerleme)

Çizelge 4.33'te 10 kat çapraz geçerleme yapıldığı senaryoda elde edilen sonuçlar yer almaktadır. Elde edilen sonuçlara göre XGBoost sınıflandırıcısı 0.8373 değeri ile en iyi doğruluk değerine sahip iken, ikinci sırada ise 0.8318 doğruluk değeri ile Rassal orman sınıflandırıcısı gelmektedir. En az doğruluk değerine sahip model ise 0.6726 değeri ile Gaussian naive bayes sınıflandırıcısıdır. En az doğruluk değerine sahip kinci sıradaki model ise 0.7747 değeri ile Karar ağacı modelidir. Şekil 4.11'de ise bu modellerin doğruluk değerlerine göre kutu grafiği

gösterimi yer almaktadır. Belirgin bir şekilde Gaussian naive bayes modeli diğer modellerden doğruluk değeri olarak ayrışmaktadır.

Çizelge 4.33. Modellerin karşılaştırılması (10 kat çapraz geçerleme)

Modeller	Doğruluk Değerleri
XGBoost Sınıflandırıcısı	0.837391
Rassal Orman Sınıflandırıcısı	0.831824
Destek Vektör Makineleri (Radial)	0.811931
K-en Yakın Komşu K=5	0.802059
K-en Yakın Komşu K=4	0.796845
Destek Vektör Makineleri (Linear)	0.792977
Lojistik Regresyon (C = 1.5)	0.789233
Lojistik Regresyon (C = 1.3)	0.789137
Karar Ağacı	0.774729
Gaussian Naive Bayes	0.672612



Şekil 4.11 Modellerin karşılaştırılması (10 kat çapraz geçerleme)

Elde edilen sonuçlar sonrası çapraz geçerleme yapılmadan önce ve sonra doğruluk oranı en yüksek olan makine öğrenmesi modeli XGBoost sınıflandırıcısı olmaktadır. XGBoost sınıflandırıcısı için ROC (Receiver operating characteristic) eğrisi ve AUC (Area under the curve) değerlerine bakmak modelin performansını ölçmek için kullanılan güçlü bir araçtır.

ROC eğrisinde, verilen bir sınıflandırma modelinin veri noktaları için tahmin edilen pozitif olasılıklarını gerçek pozitif oranları ile gösterilir. ROC eğrisi, verilerin doğru pozitif oranı (True positive rate) ve yanlış pozitif oranı (False positive rate) arasındaki ilişkisini görselleştirir. Doğru pozitif oranı, gerçek pozitif verilerin doğru tahmin edilen pozitif verilerin oranıdır. Yanlış pozitif oranı ise, gerçek negatif verilerin yanlış tahmin edilen pozitif verilerin oranıdır.
AUC Score, ROC eğrisinin altındaki alanı ölçer. AUC, 0 ile 1 arasında bir sayıdır ve modelin tahmin performansını ölçmek için kullanılır. Daha yüksek bir AUC skoru, daha iyi bir model performansını gösterir. 1 olan bir AUC skoru, tüm verilerin doğru tahmin edildiğini gösterir, 0 ise tüm verilerin yanlış tahmin edildiğini gösterir. ROC eğrisi ve AUC skoru, sınıflandırma modelinin güçlü ve zayıf yanlarını anlamaya yardımcı olur ve modelin tahmin performansının nasıl iyileştirileceğine dair fikirler sunar.

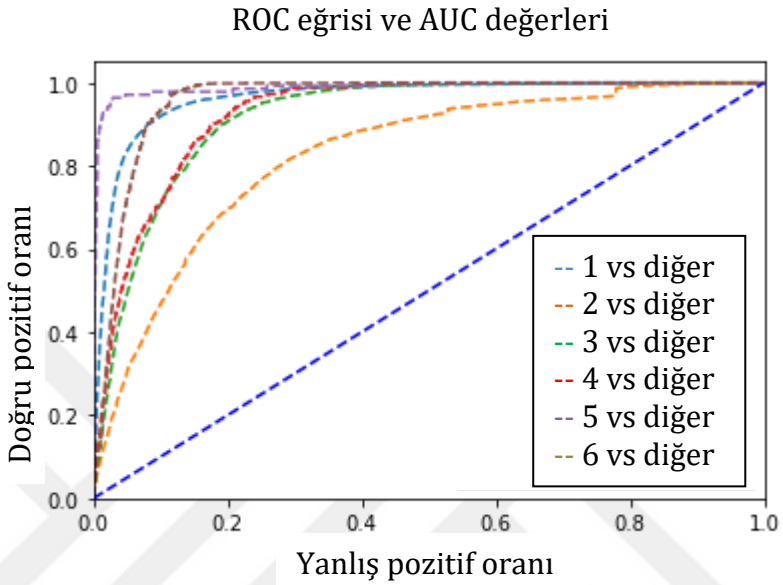
Doğru negatif, Yanlış pozitif, Yanlış negatif ve Doğru pozitif değerlerini içeren matrisin basit gösterimi Şekil 4.12'de gösterilmiştir. Doğru negatif değerleri negatif olarak tahmin edilen ve gerçekte de negatif olan değerleri göstermektedir. Yanlış pozitif değerleri pozitif olarak tahmin edilen ama gerçekte negatif olan değerleri göstermektedir. Yanlış negatif değerleri negatif olarak tahmin edilen ama gerçekte pozitif olan değerleri göstermektedir. Gerçek pozitif değerleri ise pozitif olarak tahmin edilen ve gerçekte de pozitif olan değerleri göstermektedir.



Şekil 4.12 Doğru/yanlış – pozitif/negatif tablosu

Şekil 4.12'de ise Doğru/yanlış – pozitif/negatif değerlerine göre elde edilen ROC eğrisi ve AUC değerleri görünmektedir. Veri setinde 6 farklı sınıf değişkeni

bulunduğu için “OneVsRestClassifier” yöntemi kullanılmıştır. Bu yöntemde her bir sınıf değişkeni için ilgili sınıf değişkeni dışındaki diğer sınıf değişkenleri yeni bir sınıf değişkeni olarak tanımlanır ve veri seti 2 sınıf değişkeni bulunduran bir sınıf değişkeni haline getirilir. Bu yöntem ile sınıf değişkeni sayısı kadar AUC değeri elde edilir.



Şekil 4.13 ROC eğrisi ve AUC değerleri

Çizelge 4.34’te her bir sınıf değişkeni için AUC değerleri yer almaktadır. En yüksek AUC değeri sınıf değişkeni 5 (Paket tipi 5) olanlar ve Diğer (Paket tipi 1, 2, 3, 4, 6) grubu için 0.99 ile elde edilmiştir. En düşük AUC değeri ise 0.83 değeri ile sınıf değişkeni 2 (Paket tipi 2) olanlar ve Diğer (Paket tipi 1, 3, 4, 5, 6) grubu için elde edilmiştir.

Çizelge 4.34. Sınıf değişkenleri ve AUC değerleri

Sınıf Değişkenleri	**AUC Değeri**
1 vs Diğer	0.96
2 vs Diğer	0.83
3 vs Diğer	0.92
4 vs Diğer	0.93
5 vs Diğer	0.99
6 vs Diğer	0.96

AUC değerlerinin ortalamasına baktığımızda 0.93 değeri, geometrik ortalamasına baktığımız zaman 0.93 değeri ve harmonik ortalamaya baktığımız zaman ise 0.92 değeri elde edilmektedir.

4.3 Yönetim Etkisi

Lojistik sektöründe özellikle de 3.parti lojistik şirketlerinde paketleme süreçleri çıkış süreçlerinin maliyet açısından ve verimlilik açısından önemli bir bölümünü kapsamaktadır. Paketleme sürecinde yanlış paket tiplerinin seçilmesi sürecin verimlilik değerinin düşmesine sebep olurken maliyetleri de olumsuz etkilemektedir. Genel anlamda yanlış paket tipinin kullanılması sonucu karşılaşılan olumsuz yöndeki etkiler şöyledir.

- Verimlilik değerinin düşmesi
- Maliyetlerin artması
- Marka imajının zarar görmesi

Süreç olarak bakıldığı zaman kullanılması gereken paket tipinden daha küçük bir paket seçildiği zaman ürünlerin o pakete sığmayacağı fark edilecek ve ürünler küçük paketten çıkartılarak daha büyük bir pakete yerleştirilecektir. Bu süreçte paket tipini seçme, ürünlerin barkodunu okutarak pakete yerleştirme gibi süreçlerin tekrarlanması gerekecektir. Bu durum paketlenme süresini arttırmış olacaktır. Artan paketleme süresi sadece paketleme süresinin verimliliğini değil aynı zaman da genel verimliliği de düşürecektir.

Düşen verimlilik değerine ek olarak dağıtım maliyetleri de artacaktır. Bazı lojistik şirketleri dağıtım maliyetini de kendisi karşılamaktadır. İdealden büyük bir paket tipi dağıtım şirketine verildiğinde paketin hacim değeri ile orantılı dağıtım maliyeti oluşacaktır. Kullanılmaması gereken büyük hacimli paketlerin sıklıkla kullanılması da dağıtım maliyetini gözle görülür bir şekilde artıracaktır.
İdealden uzak büyük paketlerin kullanılması sonucunda karşılaşılan diğer bir kötü etki ise ürünlerin ait olduğu markanın marka değerinin düşmesidir. Siparişler büyük paketlerde son kullanıcıya iletildiği zaman, ürünler dağıtım sürecinde paket içerisinde sıklıkla hareket edecek boş alana sahip olacak ve

paketin kenarlarına çarparak hasar görecektir. Son kullanıcı ürünleri hasarlı olarak teslim aldığında ise iade sürecine başvuracak ve markanın iade oranları artacaktır. Aynı zamanda bu kullanıcıların şikayet etmeleri, ya da farklı platformlarda bu durumu yorum olarak belirtmeleri markanın imajını kötü etkileyecektir.

Lojistik sektörlerindeki elleçleme süreçlerinde yer alan personeller sıklıkla değişebilmektedir. Süreçlerin tam anlamıyla öğrenilmeden personellerin kaybedilmesi ve yerlerine yeni kişilerin getirilmesi sonucunda öğrenme eğrisi de tam anlamıyla oluşmayacaktır. Bu kişilerin bir bilgi, birikimini oluşması sonrası farklı bir süreçte çalışmaya başlaması da olasıdır.

Verimlilik değerinin düşmesi, maliyetlerin artması, marka imajının zarar görmesi ve çalışanların bilgi birikiminin ideal seviyeye gelmeden farklı bir sürece geçmesi ya da işten ayrılması gibi sebeplerden ötürü bir sipariş için hangi paket tipinin kullanılması gerektiği kararının personele bırakılmaması gerekmektedir. Makine öğrenmesi modelleri ile geçmişte bu siparişler için hangi paket tiplerinin kullanıldığı ile ilgili bir model sonucunda yukarıda belirtilmiş riskler minimuma indirilmiş olacaktır.

5. SONUÇ VE ÖNERİLER

Lojistik sektörlerinde farklı verimlilik ve maliyet kalemleri bulunmaktadır. Bu kalemler giriş ve çıkış süreçleri olmak üzere ikiye ayrılmaktadır. Çıkış süreçlerinde yer alan paketleme ve dağıtım süreçleri verimlilik değerlerini ve maliyet değerlerini etkileyen önemli 2 süreçtir. Özellikle paketleme sürecinin verimliliğini arttırmak ve maliyetini düşürmek için çok fazla sayıda çalışma yapılmıştır. Bu çalışmaların büyük bir çoğunluğunda bu problem palet yükleme problemi ya da konteynır yükleme problemi olarak ele alınmıştır. Bu problemlerin çözümü için sıklıkla farklı matematiksel modeller ve denklemler kullanılmıştır. matematiksel modeller kadar olmasa makine öğrenmesi ve yapay zeka çalışmaları da bu problemler için geliştirilmiş ve uygulanmıştır.

Bu tez çalışmasında ise Türkiye'de yer alan bir lojistik firmasının paketleme verileri incelenmiştir. İncelenen veri seti 853.534 adet ürünü içermektedir. Bu ürünler toplamda 196.059 adet siparişe aittir. Bu siparişleri paketlemek için ise toplamda 196.400 adet paket kullanılmıştır. kullanılan paket tipleri 8 tanedir ama 2 tanesinin kullanılma amacı farklılık gösterdiği için bu paket tiplerinin kullanıldığı siparişler çalışmadan çıkartılmış ve 6 farklı paket tipi ile ilerlenmiştir.

Paketleme verilerinin olduğu veri setine farklı makine öğrenmesi modelleri uygulanmıştır. Uygulanan makine öğrenmesi modelleri şunlardır.

- XGBoost Sınıflandırıcısı
- Rassal Orman Sınıflandırıcısı
- Destek Vektör Makineleri
- K-En Yakın Komşu Sınıflandırıcısı
- Lojistik Regresyon
- Karar Ağaçları Sınıflandırma Yöntemi
- Gaussian Naive Bayes Sınıflandırıcısı

İlk olarak yukarıda yer alan modeller herhangi bir çapraz geçerleme yapılmadan uygulanılmıştır. Ardından ise 3 kat, 5 kat ve 10 kat olmak üzere 3 farklı çapraz

geçerleme yapılarak her bir model tekrardan uygulanılmıştır. Çapraz geçerleme yapılmadığı durumda en yüksek doğruluk oranına götüren model 0.8388 ile XGBoost Sınıflandırıcı olmuştur. İkinci sırada ise 0.8338 ile Rassal Orman Sınıflandırıcısı gelmektedir. Doğruluk oranı en düşük olan model ise 0.6727 ile Gaussian Naive Bayes Sınıflandırıcısı olmuştur.

Çapraz geçerleme yapıldığı durumda ise en güvenilir sonuç 10 kat çapraz geçerleme yapıldığı durumda elde edilmektedir. Bu durumda en yüksek doğruluk oranını veren model 0.8373 ile XGBoost Sınıflandırıcı olmuştur. Çapraz geçerleme yapılmadı senaryoya göre doğruluk oranı çok az da olsa düşmüştür ama modelin güvenilirliği ise artmıştır. Çapraz geçerleme yapıldığı durumda en yüksek doğruluk oranına sahip ikinci model ise 0.8318 ile Rassal Orman Sınıflandırıcısıdır. En son sırada ise Gaussian Naive Bayes Sınıflandırıcısı gelmektedir.

Bu çalışmanın sonrasında ise gelecekteki çalışmalarda parametre optimizasyonu üzerinde çalışılması planlanmaktadır. Modeller için farklı parametrelerin kullanılması doğruluk değerini ve modelin güvenilirliğini etkilemektedir ve bu sebeple makine öğrenmesi modelleri için önemli bir yer tutmaktadır. Ek olarak Elbow metodu ile paket tipi sayısının 6 yerine 4'e indirilmesi için çalışmalar detaylandırılacaktır. Bu kararı marka sahibi vereceği için ilerdeki çalışmalarda paket tipi sayısının 4'e indirmenin sağlayacağı faydalar netleştirilecek ve firmaya sunulacaktır.

KAYNAKLAR

Akar, Ö., Güngör, O., 2012. Classification of Multispectral Images Ssing Random Forest Algorithm. Journal of Geodesy and Geoinformation, 1(2), 105-112. doi:10.9733/jgg.241212.1.

Albert, A., Lesaffre, E., 1986. Multiple Group Logistic Discrimination. Statistical Methods of Discrimination and Classification, 209-224. doi:10.1016/b978-0-08-034000-5.50009-5,

Aylak, B. L., İnce, M., Oral, O., Süer, G., Almasarwah, N., Singh, M., Salah, B., 2021. Application of Machine Learning Methods for Pallet Loading Problem. Applied Sciences, 11(18), 8304. doi:10.3390/app11188304.

Bansal, M., Goyal, A., Choudhary, A., 2022. A comparative Analysis of K-Nearest Neighbor, Genetic, Support Vector Machine, Decision Tree, and Long Short Term Memory Algorithms in Machine Learning. Decision Analytics Journal, 3, 100071. doi:10.1016/j.dajour.2022.100071.

Birgin, E. G., Morabito, R., Nishihara, F. H., 2005. A Note on an L-Approach for Solving the Manufacturer's Pallet Loading Problem. Journal of the Operational Research Society, 56(12), 1448-1451. doi:10.1057/palgrave.jors.2601960.

Bischoff, E., Janetz, F., Ratcliff, M., 1995. Loading Pallets with non-Identical Items. European Journal of Operational Research, 84(3), 681-692. doi:10.1016/0377-2217(95)00031-k.

Boeckle, U., 1994. Bewertung Von Verpackungssystemen. Modelle Von Verpackungssystemen, 183-209. doi:10.1007/978-3-663-08766-3_5.

Bortfeldt, A., & Wäscher, G., 2013. Constraints in Container Loading – a State-of-the-Art Review. European Journal of Operational Research, 229(1), 1-20. doi:10.1016/j.ejor.2012.12.006.

Cai, J., Luo, J., Wang, S., Yang, S., 2018. Feature Selection in Machine Learning: A New Perspective. Neurocomputing, 300, 70-79. doi:10.1016/j.neucom.2017.11.077.

Chen, C. S., Sarın, S., Ram, B., 1991. The Pallet Packing Problem for non-Uniform Box Sizes. International Journal of Production Research, 29(10), 1963-1968. doi:10.1080/00207549108948061.

Chen, T., Guestrin, C., 2016. XGBoost. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. doi:10.1145/2939672.2939785.

Dereli, T., Daş, G.S., 2010. Konteyner Yükleme Problemleri için Karınca Koloni Optimizasyonu Yaklaşımı. Gazi Üniversitesi Mühendislik –Mimarlık Fakültesi Dergisi, 25(4), 881-894.

Dowsland, K. A., 1987. An Exact Algorithm for the Pallet Loading Problem. European Journal of Operational Research, 31(1), 78-84. doi:10.1016/0377-2217(87)90140-8.

Dubey, G. P., Bhujade, D. R., 2021. Optimal Feature Selection for Machine Learning Based Intrusion Detection System by Exploiting Attribute Dependence. Materials Today: Proceedings, 47, 6325-6331. doi:10.1016/j.matpr.2021.04.643.

ElMaraghy, H., Schuh, G., ElMaraghy, W., Piller, F., Schönsleben, P., Tseng, M., Bernard, A., 2013. Product Variety Management. CIRP Annals, 62(2), 629-652. doi:10.1016/j.cirp.2013.05.007.

Erdem, H. A., 2014. Solving Container Loading Problem with Genetic Algorithm. 2014 IEEE 15th International Symposium on Computational Intelligence and Informatics (CINTI). doi:10.1109/cinti.2014.7028707.

Aldrich, J., 1984. Estimation of Probit and Logit Models for Dichotomous Dependent Variables. Linear Probability, Logit, and Probit Models, 31-47. doi:10.4135/9781412984744.n2

G, Y., Kang, M., 2001. A Fast Algorithm for Two-Dimensional Pallet Loading Problems of Large Size. European Journal of Operational Research, 134(1), 193-202. doi:10.1016/s0377-2217(00)00249-6.

Gajda, M., Trivella, A., Mansini, R., Pisinger, D., 2020. An Optimization Approach for a Complex Real-Life Container Loading Problem. SSRN Electronic Journal. doi:10.2139/ssrn.3740046.

Gehring, H., Bortfeldt, A., 1997. A Genetic Algorithm for Solving the Container Loading Problem. International Transactions in Operational Research, 4(5-6), 401-418. doi:10.1111/j.1475-3995.1997.tb00095.x.

Gehring, H., Bortfeldt, A., 2002. A Parallel Genetic Algorithm for Solving the Container Loading Problem. International Transactions in Operational Research, 9(4), 497-511. doi:10.1111/1475-3995.00369.

George, J., Robinson, D., 1980. A Heuristic for Packing Boxes into a Container. Computers & Operations Research, 7(3), 147-156. doi:10.1016/0305-0548(80)90001-5.

Gzara, F., Elhedhli, S., Yildiz, B. C., 2020. The Pallet Loading Problem: Three-Dimensional Bin Packing with Practical Constraints. European Journal of Operational Research, 287(3), 1062-1074. doi:10.1016/j.ejor.2020.04.053.

Hand, D. J., Yu, K., 2001. Idiot's Bayes? Not so Stupid After All? International Statistical Review, 69(3), 385-398. doi:10.1111/j.1751-5823.2001.tb00465.x.

Hosmer, D. W., Lemeshow, S., 2000. Applied Logistic Regression. doi:10.1002/0471722146.

Kantardzic, M., 2011. Data Mining. doi:10.1002/9781118029145.

Knoll, D., Neumeier, D., Prüglmeier, M., Reinhart, G., 2019. An Automated Packaging Planning Approach Using Machine Learning. Procedia CIRP, 81, 576-581. doi:10.1016/j.procir.2019.03.158.

Kotlars, A., Skribans, V., 2016. Third Party Logistics Companies' Activities and Selection Methods. 9th International Scientific Conference "Business and Management 2016". doi:10.3846/bm.2016.09.

Kotsiantis, S. B., Zaharakis, I. D., Pintelas, P. E., 2006. Machine Learning: A Review of Classification and Combining Techniques. Artificial Intelligence Review, 26(3), 159-190. doi:10.1007/s10462-007-9052-3.

Lau, H., Chan, T., Tsui, W., Ho, G., Choy, K., 2009. An AI Approach for Optimizing Multi-Pallet Loading Operations. Expert Systems with Applications, 36(3), 4296-4312. doi:10.1016/j.eswa.2008.03.024.

Lee, C., Gutierrez, F., Dou, D., 2011. Calculating Feature Weights in Naive Bayes with Kullback-Leibler Measure. 2011 IEEE 11th International Conference on Data Mining. doi:10.1109/icdm.2011.29.

Lewis, D. D., 1998. Naive (Bayes) at Forty: The Independence Assumption in Information Retrieval. Machine Learning: ECML-98, 4-15. doi:10.1007/bfb0026666.

Li, Y., Chen, M., Huo, J., 2022. A Hybrid Adaptive Large Neighborhood Search Algorithm for the Large-Scale Heterogeneous Container Loading Problem. Expert Systems with Applications, 189, 115909. doi:10.1016/j.eswa.2021.115909.

Lins, L., Lins, S., Morabito, R., 2003. N L-Approach for Packing (ℓ, w)-Rectangles into Rectangular and L-Shaped Pieces. Journal of the Operational Research Society, 54(7), 777-789. doi:10.1057/palgrave.jors.2601553.

Martello, S., Pisinger, D., Vigo, D., 2000. The Three-Dimensional Bin Packing Problem. Operations Research, 48(2), 256-267. doi:10.1287/opre.48.2.256.12386.

Nokeri, T. C., 2021. Tree Modeling and Gradient Boosting with Scikit-Learn, XGBoost, PySpark, and H2O. Data Science Solutions with Python, 59-74. doi:10.1007/978-1-4842-7762-1_6.

Paquay, C., Limbourg, S., Schyns, M., 2018. A Tailored Two-Phase Constructive Heuristic for the Three-Dimensional Multiple Bin Size Bin Packing Problem with Transportation Constraints. European Journal of Operational Research, 267(1), 52-64. doi:10.1016/j.ejor.2017.11.010.

Rosenthal, A., 2016. Entwicklung Modularer Ladungsträgerkonzepte. Ganzheitliche Bewertung Modularer Ladungsträgerkonzepte, 31-53. doi:10.1007/978-3-658-15676-3_3.

Schulz, M., 2014. Die Logistikintegrierte Produktentwicklung und Ihre Organisatorische Umsetzung in der Deutschen Automobilindustrie. Logistikintegrierte Produktentwicklung, 45-81. doi:10.1007/978-3-658-04927-0_4.

Sheikh, Z., Rana, S., 2014. The Role of Logistics Service Providers in Supply Chain Performance Management: A Comprehensive Literature Review. International Journal of Academic Research in Business and Social Sciences, 4(5). doi:10.6007/ijarbss/v4-i5/886.

Shen, P., Ding, X., Ren, W., Liu, S., 2021. A Stable Feature Selection Method Based on Relevancy and Redundancy. Journal of Physics: Conference Series, 1732(1), 012023. doi:10.1088/1742-6596/1732/1/012023.

Sheng, L., Xiuqin, S., Changjian, C., Hongxia, Z., Dayong, S., Feiyue, W., 2017. Heuristic Algorithm for the Container Loading Problem with Multiple Constraints. Computers & Industrial Engineering, 108, 149-164. doi:10.1016/j.cie.2017.04.021.

Terno, J., Scheithauer, G., Sommerweiß, U., Riehme, J., 2000. An Efficient Approach for the Multi-Pallet Loading Problem. European Journal of Operational Research, 123(2), 372-381. doi:10.1016/s0377-2217(99)00263-5.

Voskoglou, M. G., 2020. Problem Solving and Mathematical Modelling. doi:10.20944/preprints202012.0672.v1.

Yiğit, T., Aydemir, M., 2018. Paket Döndürmeksİzİn Genetİk Algoritma Kullanarak Konteyner Yükleme Problemi Optimizasyonu. Mühendislik Bilimleri Ve Tasarım Dergisi, 21-28. doi:10.21923/jesd.318853.

EKLER

EK A. Python Kodları

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import itertools
import matplotlib.gridspec as gridspec
import seaborn as sns
sns.set(style='whitegrid',palette='muted',font_scale=1.5)
from pylab import rcParams
rcParams['figure.figsize']=12,8
from sklearn import svm
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.model_selection import KFold
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn import metrics
import warnings
warnings.filterwarnings('ignore')
from sklearn.metrics import roc_curve, auc
from sklearn.cluster import KMeans
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
from sklearn import tree
from matplotlib import pyplot as plt
from xgboost import XGBClassifier
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
from sklearn.preprocessing import StandardScaler
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import *
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import MinMaxScaler
from sklearn.multiclass import OneVsRestClassifier
from sklearn.preprocessing import label_binarize
from collections import OrderedDict
```

Şekil A.1. Gerekli kütüphanelerin eklenmesi

```
Paket=pd.read_csv('AnaTabloPythonTR.csv')
Paket.isnull().sum()
```

Şekil A.2. Excelin tanımlanması ve sütun kontrolleri

```
Paket.describe()
Paket.info()
```

Şekil A.3. Excelde yer alan veri tiplerinin kontrolü

```
sns.countplot(x='Paket tipi',data=Paket)
plt.show()
```

Şekil A.4. Sınıf değişkenlerinin dağılımı

```
grouped = Paket.groupby('Paket tipi').agg({'Ürünlerin toplam yüksekliği':['mean','std',min,max],'Ürünlerin maksimum yüksekliği':[
})
grouped.columns=["_".join(x) for x in grouped.columns.ravel()]
grouped
```

Şekil A.5. Her bir öznitelik için istatistiki değerler

```
X = Paket.drop('Paket tipi', axis=1)
y = Paket['Paket tipi']
fig, ax = plt.subplots(figsize=(20, 15))
sns.heatmap(X.corr(), annot=True)
plt.show()
```

Şekil A.6. Isı haritasının çıkartılması

```
cor = X.corr()
cor
```

Şekil A.7. Korelasyon değerlerinin hesaplanması

```
Paket.corr()["Paket tipi"].abs().nlargest(15)
```

Şekil A.8. En yüksek korelasyon değerine sahip 15 öz nitelik

```
sns.FacetGrid(Paket,hue="Paket tipi",size=7).map(sns.distplot,"Ürünlerin toplam uzunluğu").add_legend()
sns.FacetGrid(Paket,hue="Paket tipi",size=7).map(sns.distplot,"Ürünlerin toplam genişliği").add_legend()
sns.FacetGrid(Paket,hue="Paket tipi",size=7).map(sns.distplot,"Ürünlerin toplam yüksekliği").add_legend()
sns.FacetGrid(Paket,hue="Paket tipi",size=7).map(sns.distplot,"Toplam ürün sayısı").add_legend()
sns.FacetGrid(Paket,hue="Paket tipi",size=7).map(sns.distplot,"Toplam farklı ürün sayısı").add_legend()
sns.FacetGrid(Paket,hue="Paket tipi",size=7).map(sns.distplot,"Ürünlerin maksimum uzunluğu").add_legend()
sns.FacetGrid(Paket,hue="Paket tipi",size=7).map(sns.distplot,"Ürünlerin maksimum genişliği").add_legend()
sns.FacetGrid(Paket,hue="Paket tipi",size=7).map(sns.distplot,"Ürünlerin maksimum yüksekliği").add_legend()
sns.FacetGrid(Paket,hue="Paket tipi",size=7).map(sns.distplot,"Ortalama uzunluk/genişlik").add_legend()
sns.FacetGrid(Paket,hue="Paket tipi",size=7).map(sns.distplot,"Ürünlerin toplam desi değeri").add_legend()
sns.FacetGrid(Paket,hue="Paket tipi",size=7).map(sns.distplot,"Ortalama yükseklik/genişlik").add_legend()
sns.FacetGrid(Paket,hue="Paket tipi",size=7).map(sns.distplot,"Ortalama uzunluk/yükseklik").add_legend()

plt.savefig("chart.png")
```

Şekil A.9. Her bir öz niteliğin dağılım grafiği

```
Attrition_Flag=dataset['Package Type']
data=dataset[dataset.columns[:12]]

#%25 test verisi %75 eğitme verisi
train,test=train_test_split(dataset,test_size=0.25,random_state=0,stratify=dataset['Package Type'])

#eğitme verisi
train_X=train[train.columns[:12]]
train_Y=train['Package Type']

#test verisi
test_X=test[test.columns[:12]]
test_Y=test['Package Type']
```

Şekil A.10. Eğitme ve test verilerinin belirlenmesi

```
features=train_X.columns.values

for feature in features:
    mean, std=train_X[feature].mean(),train_X[feature].std()
    train_X.loc[:,feature]=(train_X[feature]-mean)/std

    test_X.loc[:,feature]=(test_X[feature]-mean)/std
```

Şekil A.11. Standardizasyon ve normalizasyon

```
Accuracy=[]
classifiers=['Logistic Regression c=0.1',
             'Logistic Regression c=0.3',
             'Logistic Regression c=0.5',
             'Logistic Regression c=0.8',
             'Logistic Regression c=1.0',
             'Logistic Regression c=1.3',
             'Logistic Regression c=1.5',
             'Logistic Regression c=1.8',
             'Logistic Regression c=2.0']
models=[LogisticRegression(C=0.1),
        LogisticRegression(C=0.3),
        LogisticRegression(C=0.5),
        LogisticRegression(C=0.8),
        LogisticRegression(C=1.0),
        LogisticRegression(C=1.3),
        LogisticRegression(C=1.5),
        LogisticRegression(C=1.8),
        LogisticRegression(C=2.0)]

for i in models:
    model = i
    model.fit(train_X,train_Y)
    prediction=model.predict(test_X)
    #prediction değerlewrini gerçek y değerleri ile karşılaştırıyoruz.
    Accuracy.append(metrics.accuracy_score(prediction,test_Y))

models_dataframe=pd.DataFrame(Accuracy,index=classifiers)
models_dataframe.columns=['Accuracy']
models_dataframe.sort_values(['Accuracy'],ascending=[0])
```

Şekil A.12. Lojistik regresyon modelinin uygulanması

```
model_lr.predict_proba(test_X[:5])
model_lr.predict(test_X[:5])
```

Şekil A.13. İlk 5 örneğin sınıf değerinin tahmin edilmesi

```
Accuracy=[]
classifiers=['K-Nearest Neighbors n=1',
             'K-Nearest Neighbors n=2',
             'K-Nearest Neighbors n=3',
             'K-Nearest Neighbors n=4',
             'K-Nearest Neighbors n=5',
             'K-Nearest Neighbors n=6',
             'K-Nearest Neighbors n=7',
             'K-Nearest Neighbors n=8',
             'K-Nearest Neighbors n=9',
             'K-Nearest Neighbors n=10',]
models=[KNeighborsClassifier(n_neighbors=1),
        KNeighborsClassifier(n_neighbors=2),
        KNeighborsClassifier(n_neighbors=3),
        KNeighborsClassifier(n_neighbors=4),
        KNeighborsClassifier(n_neighbors=5),
        KNeighborsClassifier(n_neighbors=6),
        KNeighborsClassifier(n_neighbors=7),
        KNeighborsClassifier(n_neighbors=8),
        KNeighborsClassifier(n_neighbors=9),
        KNeighborsClassifier(n_neighbors=10)]
for i in models:
    model = i
    model.fit(train_X,train_Y)
    prediction=model.predict(test_X)
    #prediction değerlewrini gerçek y değerleri ile karşılaştırıyoruz.
    Accuracy.append(metrics.accuracy_score(prediction,test_Y))

models_dataframe=pd.DataFrame(Accuracy,index=classifiers)
models_dataframe.columns=['Accuracy']
models_dataframe.sort_values(['Accuracy'],ascending=[0])
```

Şekil A.14. K-en yakın komşu yöntemi modelinin uygulanması

```
wcss=[]
for i in range(1,10):
    kmeans=KMeans(n_clusters=i, init='k-means++',n_init=10,max_iter=500,random_state=0)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)
print(wcss)

plt.plot(range(1,10),wcss)
plt.title('Elbow Methodu')
plt.xlabel('Küme Sayısı')
plt.ylabel('Sınıf İçi Kareli Hata')
```

Şekil A.15. Sınıf içi kareli hata değerlerinin hesaplanması

```
Accuracy=[]
classifiers=['Karar Agacı']
models=[DecisionTreeClassifier()]
for i in models:
    model = i
    model.fit(train_X,train_Y)
    prediction=model.predict(test_X)
    #prediction değerlewrini gerçek y değerleri ile karşılaştırıyoruz.
    Accuracy.append(metrics.accuracy_score(prediction,test_Y))

models_dataframe=pd.DataFrame(Accuracy,index=classifiers)
models_dataframe.columns=['Accuracy']
models_dataframe.sort_values(['Accuracy'],ascending=[0])
```

Şekil A.16. Karar ağacı modelinin uygulanması

```
fea_imp.plot(kind='barh',x='col',y='imp',figsize=(10,10),legend=None)
plt.title('Karar Ağacı Öznitelik Önemleri')
plt.ylabel('Öznitelikler')
plt.xlabel('Önemler')
```

Şekil A.17. Öz niteliklerin önem derecesinin belirlenmesi

```
from sklearn.metrics import confusion_matrix
predictions=model_dt.predict(test_X)
confusion_matrix(test_Y, predictions)
```

Şekil A.18. Hata matrisinin oluşturulması

```
from sklearn import tree
from matplotlib import pyplot as plt

fig=plt.figure(figsize=(30,30))
_=tree.plot_tree(model_dt,max_depth=2,feature_names=feature_names,class_names=('0','1','2','3','4','5','6'),fontsize=20)
```

Şekil A.19. Karar ağacının çizilmesi

```
Accuracy=[]
classifiers=['linear Svm',
             'Radial Svm',
             'polynomial Svm']
models=[svm.SVC(kernel='linear'),
        svm.SVC(kernel='rbf'),
        svm.SVC(kernel='poly')]
for i in models:
    model = i
    model.fit(train_X,train_Y)
    prediction=model.predict(test_X)
    #prediction değerlewrini gerçek y değerleri ile karşılaştırıyoruz.
    Accuracy.append(metrics.accuracy_score(prediction,test_Y))

models_dataframe=pd.DataFrame(Accuracy,index=classifiers)
models_dataframe.columns=['Accuracy']
models_dataframe.sort_values(['Accuracy'],ascending=[0])
```

Şekil A.20. Destek vektör makineleri modelinin uyğulanması

```
model_gnb = GaussianNB()
```

```
model_gnb.fit(train_X, train_Y)
```

Şekil A.21. Gaussian naïve bayes modelinin uyğulanması

```
model_rf = RandomForestClassifier(n_estimators=100,random_state=0)
```

```
model_rf.fit(train_X, train_Y)
```

Şekil A.22. Rassal orman modelinin uyğulanması

```
from xgboost import XGBClassifier
```

```
model_xgb = XGBClassifier()
```

Şekil A.23. XGBoost modelinin uyğulanması

```
#10 kat çapraz geçerleme
kfold=KFold(n_splits=10)
```

Şekil A.24. 10 kat çapraz geçerleme

```
XG.fit(X_train_norm,y_train)
y_pred =XG.predict(X_test_norm)
pred_prob = XG.predict_proba(X_test_norm)

y_test_binarized=label_binarize(y_test,classes=np.unique(y_test))

# roc curve for classes
fpr = {}
tpr = {}
thresh ={}
roc_auc = dict()

n_class = classes.shape[0]

for i in range(n_class):
    fpr[i], tpr[i], thresh[i] = roc_curve(y_test_binarized[:,i], pred_prob[:,i])
    roc_auc[i] = auc(fpr[i], tpr[i])

    # plotting
    plt.plot(fpr[i], tpr[i], linestyle='--',
             label='%s vs Diğer (AUC=%0.2f)'%(classes[i],roc_auc[i]))

plt.plot([0,1],[0,1],'b--')
plt.xlim([0,1])
plt.ylim([0,1.05])
plt.title('ROC Eğrisi ve AUC Değerleri')
plt.xlabel('Yanlış Pozitif Oranı')
plt.ylabel('Doğru Pozitif Oranı')
plt.legend(loc='lower right')
plt.show()
```

Şekil A.25. ROC ve AUC değerleri

ÖZGEÇMİŞ

Adı Soyadı : Yasin Can KILIÇKAP

Eğitim Durumu

Lise : Özel Ergani Seçenek Uğur Lisesi, 2016

Lisans : Orta Doğu Teknik Üniversitesi, MühendislikFakültesi, Endüstri Mühendisliği Bölümü

Yüksek Lisans : İstanbul Ticaret Üniversitesi,
Fen Bilimleri Enstitüsü, Endüstri Mühendisliği Anabilim Dalı

Yayınları

Kılıçkap, Y. C., Ayvaz, B., 2022. Selecting Appropriate Type of Package with Machine Learning Models in Logistic Companies. Southeast Europe Journal of Soft Computing, 11(2), 40–48.