T.C.
DOKUZ EYLÜL ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
YÖNETİM BİLİŞİM SİSTEMLERİ ANABİLİM DALI
YÖNETİM BİLİŞİM SİSTEMLERİ PROGRAMI
YÜKSEK LİSANS TEZİ

FATURA YÖNETİMİ UYGULAMASINDA FATURALARIN MAKİNE ÖĞRENMESİ İLE SINIFLANDIRILMASI VE BİR UYGULAMA

Bilgehan AVCI

DANIŞMAN
Doç. Dr. Can AYDIN

2023

T.C.
DOKUZ EYLÜL ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
YÖNETİM BİLİŞİM SİSTEMLERİ ANABİLİM DALI
YÖNETİM BİLİŞİM SİSTEMLERİ PROGRAMI
YÜKSEK LİSANS TEZİ

FATURA YÖNETİMİ UYGULAMASINDA FATURALARIN MAKİNE ÖĞRENMESİ İLE SINIFLANDIRILMASI VE BİR UYGULAMA

Bilgehan AVCI

DANIŞMAN
Doç. Dr. Can AYDIN

İZMİR – 2023

TEZ ONAY SAYFASI

YEMİN METNİ

Yüksek Lisans Tezi olarak sunduğum “Fatura Yönetimi Uygulamasında Faturaların Makine Öğrenmesi İle Sınıflandırılması Ve Bir Uygulama” adlı çalışmanın, tarafımdan, akademik kurallara ve etik değerlere uygun olarak yazıldığını ve yararlandığım eserlerin kaynakçada gösterilenlerden oluştuğunu, bunlara atıf yapılarak yararlanılmış olduğunu belirtir ve bunu onurumla doğrularım.

Bilgehan AVCI

...../…/2023

ÖZET

Yüksek Lisans Tezi

Fatura Yönetimi Uygulamasında Faturaların Makine Öğrenmesi ile Sınıflandırılması ve Bir Uygulama

Bilgehan AVCI

Dokuz Eylül Üniversitesi
Sosyal Bilimler Enstitüsü
Yönetim Bilişim Sistemleri Anabilim Dalı
Yönetim Bilişim Sistemleri Programı

Teknolojinin ulaşılabilir bir hale gelmesi ve yaygınlaşması şirketlerin büyük miktarlardaki veriyi daha düşük maliyetlerle, daha kolay, düzenli ve kullanılabilir bir şekilde depolayabilmelerine olanak sağlamaktadır. Depolanan verinin işlenmesi ve analiz edilerek karar verme süreçlerine dahil edilmesi de operasyonel olarak şirketlere büyük kolaylıklar sağlayabilmektedir.

Bu çalışmada; aktif olarak kullanılmakta olan ve çok sayıda kullanıcısı olan bir muhasebe uygulamasında kullanıcılar tarafından gerçekleştirilmekte olan faturaların sınıflandırılması işlemini otomatik hale getirecek bir karar destek sistemi geliştirilmesi amaçlanmıştır. Mevcut durumda sisteme yüklenen faturalar ilk olarak sınıflandırmadan sorumlu kullanıcılar tarafından incelenerek hangi sınıfa ait olduğu belirlenmektedir. Daha sonra belirlenen sınıfın kullanıcılarına iletilmektedir. Bu sınıflandırma işleminin kullanıcılar tarafından yapılması hem iş gücü kaybına hem de özellikle iş yoğunluğunun yüksek olduğu dönemlerde gecikmelere yol açmaktadır. Bu doğrultuda uygulamanın veri tabanında kayıtlı yaklaşık 450 bin adet faturanın verileri düzenlenerek makine öğrenmesi teknikleri ile bir model eğitilmiştir. Daha sonra eğitilen bu model kullanılarak fatura sınıfını otomatik tespit eden bir uygulama geliştirilmiştir.

Anahtar kelimeler: Makine Öğrenmesi, Sınıflandırma, Python, Python Flask, Karar Destek Sistemi.

ABSTRACT

Master's Thesis

Classification of Invoices with Machine Learning in Invoice Management Application and an Application

Bilgehan AVCI

Dokuz Eylül University

Graduate School of Social Sciences

Department of Management Information Systems

Management Information Systems Program

The accessibility and proliferation of technology enable companies to store large amounts of data at lower costs, in a more straightforward, organized, and usable manner. Processing and analyzing the stored data and incorporating it into decision-making processes also provide significant operational advantages for companies.

In this study; it is aimed to develop a decision support system that will automate the classification process of invoices carried out by users in an accounting application that is actively used and has many users. In the current situation, the invoices uploaded to the system are first examined by the users responsible for classification and the class they belong to is determined. It is then forwarded to the users of the specified class. Performing this classification process by users causes both labor loss and delays, especially during periods of high workload. In this direction, a model was trained with machine learning techniques by editing the data of approximately 450 thousand invoices registered in the application's database. Using this trained model, an application was developed that automatically detects the invoice class.

Keywords: Machine Learning, Classification, Python, Python Flask, Decision Support System.

FATURA YÖNETİMİ UYGULAMASINDA FATURALARIN MAKİNE ÖĞRENMESİ İLE SINIFLANDIRILMASI VE BİR UYGULAMA

İÇİNDEKİLER

TEZ ONAY SAYFASI ii
YEMİN METNİ iii
ÖZET iv
ABSTRACT v
İÇİNDEKİLER vi
KISALTMALAR viii
ŞEKİLLER LİSTESİ ix

GİRİŞ 1

BİRİNCİ BÖLÜM
İŞLETME YÖNETİMİNDE BİLGİ TEKNOLOJİLERİNİN KULLANIMI

1.1. PROBLEMİN TANIMI 3
1.2. ARAŞTIRMA SORULARI 6
1.3. LİTERATÜR ÇALIŞMASI 6

İKİNCİ BÖLÜM
KARAR DESTEK SİSTEMLERİ, ELEKTRONİK FATURA KAVRAMI VE FATURALARIN SINIFLANDIRILMASINDA UYGULANABİLECEK YÖNTEMLER

2.1. KARAR DESTEK SİSTEMLERİ 9
2.2. ELEKTRONİK FATURA KAVRAMI 10
2.3. KURAL TABANLI KODLAMA YÖNTEMİ 11
2.4. MAKİNE ÖĞRENMESİ YÖNTEMİ 13
2.4.1. Makine Öğrenmesi Algoritmaları 17

ÜÇÜNCÜ BÖLÜM
ARAŞTIRMA YÖNTEMİ VE SİSTEM GELİŞTİRME YAŞAM DÖNGÜSÜ

3.1. ARAŞTIRMA YÖNTEMİ 23
3.2. SİSTEM GELİŞTİRME YAŞAM DÖNGÜSÜ 24

DÖRDÜNCÜ BÖLÜM
UYGULAMA PROJE YÖNETİMİ VE GELİŞTİRİLEN ARAYÜZÜN BÖLÜMLERİ

4.1. VERİLERİN TOPLANMASI VE ANLAŞILMASI AŞAMASI 30
4.2. VERİ ÖN İŞLEME AŞAMASI 31
4.3. SAYISAL OLMAYAN VERİLERİN NUMARALANDIRILMASI VE VERİLERİN DÜZENLENMESİ 36
4.4. EĞİTİM AŞAMASI 40
4.5. TEST AŞAMASI 44
4.6. İYİLEŞTİRME AŞAMASI 47
4.7. MODELİN KULLANILABİLECEĞİ KULLANICI ARAYÜZÜNÜN GELİŞTİRİLMESİ 49

SONUÇ VE ÖNERİLER 54
KAYNAKÇA 56

KISALTMALAR

GİB	Gelir İdaresi Başkanlığı
SGYD	Sistem Geliştirme Yaşam Döngüsü
SQL	Structured Query Language
VUK	Vergi Usul Kanunu
XML	Extensible Markup Language
YBS	Yönetim Bilişim Sistemleri

ŞEKİLLER LİSTESİ

Şekil 1: Yazılım Projelerinde Zamanla Ortaya Çıkan Ufak Ve Büyük Değişiklik İhtiyaçları s. 4
Şekil 2: Mevcut Yazılımın Yaşam Döngüsü s. 5
Şekil 3: Karar Destek Sistemi Katmanları s. 10
Şekil 4: Sınıflandırma Kuralları Örnekleri s. 12
Şekil 5: Makine Öğrenmesi Çalışmalarında İzlenen Adımlar s. 14
Şekil 6: Makine Öğrenmesi Kategorileri s. 15
Şekil 7: Kümelere Ayrılmış Verilerin Koordinat Düzleminde Gösterimi s. 16
Şekil 8: Müşteri Tablet Satın Alır/Almaz Probleminin Karar Ağacı Şeması Örneği s. 19
Şekil 9: Rassal Orman Algoritmasının Sınıflandırma Yapısı Görseli s. 20
Şekil 10: 3 Sınıflı Sentetik Bir 2b Veri Kümelerinde Knn Kümelemesi Örneği s. 22
Şekil 11: Modelin Uygulamaya Dahil Edilebileceği Uygulama Yaşam Döngüsü Görseli s. 24
Şekil 12: Sistem Geliştirme Yaşam Döngüsü s. 25
Şekil 13: Waterfall Modeli s. 28
Şekil 14: YBS nedir? s. 29
Şekil 15: Uygulama Veri Tabanında Kullanılan Yapının Temsili Gösterimi s. 31
Şekil 16: ID verileri alt tire (‘_’) karakteri içermeyen verilerin toplamının ters eğik çizgi (‘\’) kullanılmadan Sql dili ile sorgulanması s. 32
Şekil 17: ID numaraları alt tire (‘_’) karakteri içermeyen verilerin toplamının ters eğik çizgi (‘\’) kullanılarak Sql dili ile sorgulanması s. 32
Şekil 18: Tablo1 Tablosunda ID Numaraları Alt Tire (‘_’) Karakteri İçeren Verilerin Toplamının Sorgulanması s. 33
Şekil 19: Tablo1’de Yer Alan ID Numaraları Alt Tire İçeren Kayıtların Silinmesini Sağlayan Sql Sorgusu s. 33
Şekil 20: MySQL Veri Tabanında Tablo1 Ve Tablo2 Tablolarının Birleştirildiği Sql Sorgusu s. 34
Şekil 21: MySQL Veri Tabanında Tablo3 Tablosunda Yer Alan Farklı Fatura Sınıflarının Ve Adetlerinin Çekildiği Sql Sorgusu. s. 35

Şekil 22: MySQL Veri Tabanından Çekilen Tablo3 Tablosunda Yer Alan Farklı Fatura Sınıflarının, Adetlerinin Ve Oranlarının Excel Uygulamasında Grafik Haline Getirilmiş Hali s. 36
Şekil 23: FaturaProfil Başlığı Altında Yer Alan Farklı Değerlerin MySQL Uygulamasında Sorgulanması s. 37
Şekil 24: FaturaTip Başlığı Altında Yer Alan Farklı Değerlerin Mysql Uygulamasında Sorgulanması s. 38
Şekil 25: ParaBirimi Başlığı Altında Yer Alan Farklı Değerlerin Mysql Uygulamasında Sorgulanması s. 39
Şekil 26: FaturaERPTip başlığı altında yer alan girdilerin kaç farklı değer içerdiğinin MySQL uygulamasında sorgulanması. s. 40
Şekil 27: Kütüphanelerin, Makine Öğrenmesi Algoritmalarının Ve Excel Dosyası Olarak Hazırlanmış Olan Veri Setinin Eğitim Koduna Dahil Edildiği Python Kod Bloğu Görseli s. 41
Şekil 28: Lojistik Regresyon Algoritmasının Başarı Oranı s. 41
Şekil 29: Naive Bayes Algoritmasının Başarı Oranı s. 42
Şekil 30: Karar Ağacı algoritmasının başarı oranı s. 42
Şekil 31: Rassal Orman Algoritmasının Başarı Oranı s. 43
Şekil 32: Destek Vektör Makinesi algoritmasının başarı oranı s. 43
Şekil 33: Destek K-En Yakın Komşu Algoritmasının Başarı Oranı s. 44
Şekil 34: Test işlemi için kullanılan Excel dosyasına modelin sınıf tahminlerinin yazdırılmasını sağlayan Python kod bloğu görseli s. 45
Şekil 35: Modellerin tahmin sonuçlarının yazdırıldığı Excel sütunları görseli s. 46
Şekil 36: Verilerin Modele Etkilerini Grafik Şeklinde Gösterilmesini Sağlayan Python Kod Bloğu Görseli s. 48
Şekil 37: Verilerin, Modelin Eğitimine Etkilerinin Grafiği s. 48
Şekil 38: Flask Frameworkü İle Geliştirilen Uygulamada Kullanılan Python Kod Bloğu Görseli s. 49
Şekil 39: Kullanıcı Giriş Ekranı s. 50
Şekil 40: Giriş Ekranında Hatalı Kullanıcı Adı Ve/Veya Şifre Uyarısı s. 51
Şekil 41: Geliştirilen Uygulamanın Kullanıcı Arayüzü s. 52
Şekil 42: Verileri Girilen Faturanın Sınıfının Ekrana Yazdırılması s. 53

Şekil 43: Rassal Orman Algoritmasının Kullanıldığı Python Kod Bloğu Görseli s. 54

GİRİŞ

Dijitalleşme çağının getirilerini gerek sosyal hayatta gerekse iş dünyasında yoğun bir şekilde yaşadığımız bugünlerde, işletmelerin muhasebe ve finans gibi yoğun ve rutin hesaplama ağırlıklı iş yükü olan departmanlarındaki iş ve işlemler dijital muhasebe uygulamaları, robotik süreç otomasyonları gibi farklı bilgi teknolojilerinden yararlanılarak gerçekleştirilmektedir.

Şirketler çalışma şekillerine, büyüklüklerine ve evrak yoğunluklarına göre çok sayıda muhasebe personeli istihdam edebilmektedirler. Yine çalışma şekillerine bağlı olarak muhasebe departmanlarında iş bölümü ve görev paylaşımı yapılmaktadır.

Bu çalışmada; kurumsal bir şirketin muhasebe departmanında aktif olarak kullanılmakta olan ve 100'ün üzerinde kullanıcısı olan bir muhasebe uygulamasında kullanıcılar tarafından manuel olarak gerçekleştirilmekte olan faturaların sınıflandırılması işlemini otomatikleştirecek bir karar destek sistemi geliştirilmesi amaçlanmıştır.

Mevcut sistemde sınıflandırma işleminden sorumlu 2 muhasebe kullanıcısı bulunmaktadır. Kayıt işlemlerini yapan kullanıcılar da faturaların sınıflarına göre görevlendirilmiştir. Uygulamaya yüklenen faturalar ilk olarak sınıflandırma kullanıcılarının görevlerine düşmektedir. Sınıflandırma kullanıcıları tarafından birer birer açılarak sınıfları manuel olarak belirlenen faturalar daha sonra kayıt işlemlerinin yapılması için belirlenen sınıfın kullanıcılarının görevlerine akmaktadır.

Sınıflandırma işleminin manuel olarak yapılması özellikle iş yoğunluğunun yüksek olduğu dönemlerde ciddi gecikmelere sebebiyet vermektedir. Bu durum da şirket için ayrıca bir maliyet oluşturmaktadır. Bu nedenle sınıflandırma işleminin otomatik hale getirilmesi departmanın verimliliğinin artması için önem arz etmektedir.

Bu çalışma kapsamında sınıflandırma işleminin otomatik hale getirilmesi için hangi yöntemlerin kullanılabileceği konusunda araştırmalar yapılmış, yöntemlerin avantaj ve dezavantajları incelendiğinde makine öğrenmesi yöntemi uygulanmasına karar verilmiştir. Makine öğrenmesi yöntemi ile yapılan sınıflandırma çalışmasında uygulamanın veri tabanında kayıtlı geçmiş yıllara ait yaklaşık 450 bin adet faturanın verileri kullanılarak Python programlama dili ile bir makine öğrenmesi modeli eğitilmiştir. Modelin geliştirilmesi ve eğitimi tamamlandıktan sonra ise Python Flask

framework'ü ile bir web uygulaması ve kullanıcı arayüzü geliştirilmiştir. Geliştirilen arayüz, kullanıcı tarafından veri girişlerinin yapılabileceği ve bir buton ile veri girişi yapılan faturanın modelin tahminine göre hangi sınıfa ait olduğu sonucunun ekrana getirileceği şekilde tasarlanmıştır.

BİRİNCİ BÖLÜM
İŞLETME YÖNETİMİNDE BİLGİ TEKNOLOJİLERİNİN KULLANIMI

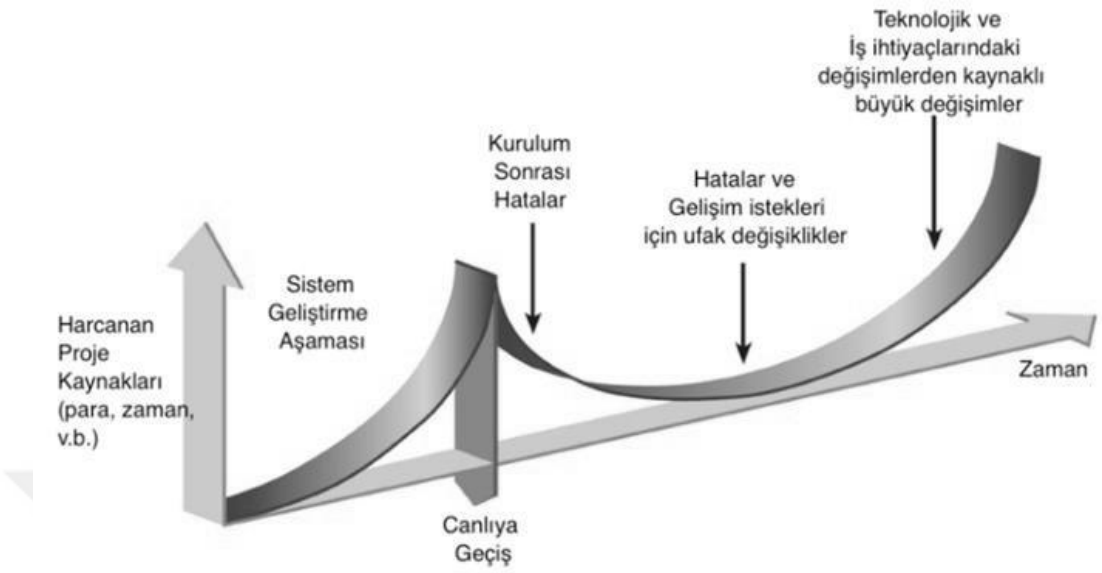
Bilgi teknolojilerinin etkin kullanılması işletmeler için büyük önem taşımaktadır. Belli bir dönemde satın alınan programlar zamanla ihtiyaçlara karşılık verememeye, yetersiz ve yavaş kalmaya başlayabilmekte veya piyasaya yeni sürülen bir ürün mevcutta kullanılana göre işletme açısından çok daha verimli olabilmektedir. İşletmeler yönetimsel veya operasyonel ihtiyaçlarına göre bilgi teknolojilerine yatırım yapmakta ve gerektiği durumlarda da güncellemeler veya yeni satın alımlar yaparak bilgi teknolojilerinden daha yüksek verimle faydalanmayı amaçlamaktadırlar. Bu güncelleme veya satın almaların ne şekilde yapılacağına karar verilmesi de önemli bir iş sürecidir. Karar vericiler bu aşamada seçenekleri araştırıp avantaj ve dezavantajlarını analiz ederek çıkarımlar yapmakta ve bu çıkarımlara göre karar vermektedirler. Bu aşamada da ilk olarak sorunun net bir şekilde tanımlanması gerekmektedir.

1.1. PROBLEMİN TANIMI

Günümüzde iş dünyası sistematik zorluklarla doludur. İş bölümleri kapsamlı ve birbiriyle bağlantılı birçok boyuttan oluşmaktadır. Pek çok araştırma karmaşıklaşan iş modellerinin hızla gelişen ve değişen iş dünyasında verimlilik ve başarının sürdürülmesinin önündeki büyük bir engel olduğunu göstermektedir. İş modelinde verimlilik tekrarlanabilir, sistematik ve düzenli süreçlerle sağlanabilir. Bu doğrultuda iş modelinin değişen şartlara göre güncellenmesi, ihtiyaca göre veya belli periyotlarla etkin bir şekilde denetimi ve iyileştirilmesi işletmeler açısından oldukça önemlidir (Sebetçi ve diğerleri, 2018).

İşletmeler bilgisayar ve internetle birlikte dijital muhasebe uygulamalarına geçmiş ve pek çok alanda olduğu gibi muhasebe departmanlarında da bilgi teknolojilerinin kullanılarak çeşitli yazılım uygulamalarından yararlanılması durumu giderek artmıştır (Tektüfekçi, 2012).

Şekil 1: Yazılım Projelerinde Zamanla Ortaya Çıkan Ufak Ve Büyük Değişiklik İhtiyaçları



Kaynak: Kendall, K. E., Kendall, J. E., Kendall, K. E. ve Kendall, J. A. 2011: 20.

Şekil 1'de yazılım projelerinde ortaya çıkan ufak düzeltme ve köklü değişiklik ihtiyaçlarının zamana bağlı gösterimi yer almaktadır. Grafikte gösterildiği gibi zaman ilerledikçe daha kapsamlı değişikliklere ihtiyaç duyulmaktadır.

Bu çalışmada ele alınan muhasebe programı da belli bir iş modeline göre oluşturulmuştur. Bu iş modelinde programı kullanan çalışanlar arasında iş bölümü ve yetki sınırlamaları yapılmıştır. Sisteme yüklenen faturaların hangi sınıfa ait olduğunun belirlenmesi ve o sınıfın kullanıcılarının görevlerine akması için kural tabanlı kodlama ile belli kurallar tanımlanmıştır. Sistem, yüklenen faturalardan tanımlanan bu kuralların kapsamına girenlerin otomatik olarak sınıflandırılacağı, kapsama girmeyen faturaların da manuel olarak sınıflandırılmak üzere 'sınıflandırma kullanıcısı' olarak belirlenmiş olan kullanıcıların görevlerine düşeceği şekilde tasarlanmıştır. Kayıtlı faturalar incelendiğinde tanımlanan bu kuralların toplam faturaların yaklaşık olarak %7'sini otomatik olarak sınıflandırdığı, bu kuralların kapsamadığı geriye kalan kısmın ise sınıflandırma kullanıcıları tarafından manuel olarak sınıflandırıldığı tespit edilmiştir.

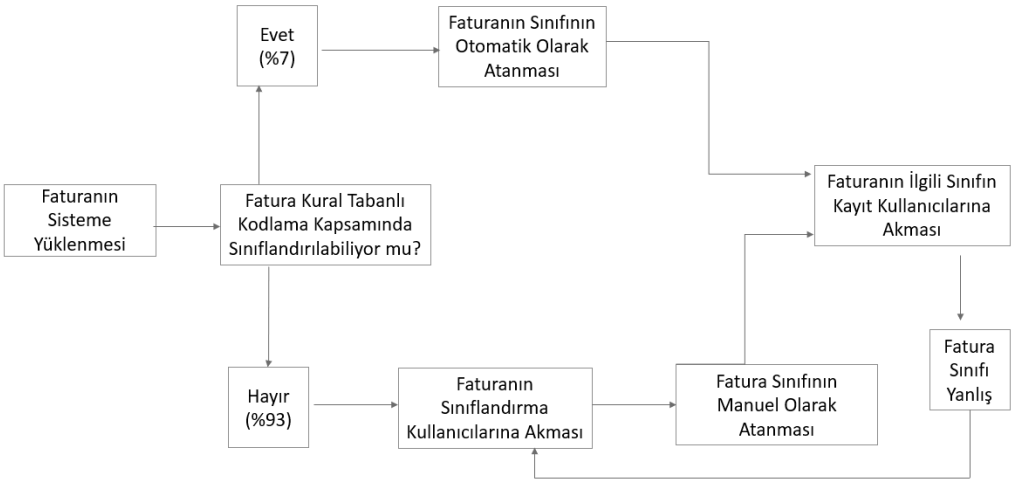
İş sürecinin verimliliği açısından incelendiğinde;

- Faturaların kayıt kullanıcılarına akmadan önce sınıflandırma kullanıcılarının görevlerinde beklemesi,

- Faturaların, sınıflandırma kullanıcıları işlem yapana kadar kayıt aşamasına geçilememesi,
- Fatura kayıtlarının gecikmesinden kaynaklı departmanda iş sürecinin yavaşlaması, bu durumun ödemelerde gecikmeye ve ekstra maliyetlere sebebiyet vermesi,
- Sınıflandırma işlemi için 2 personelin istihdam edilmesi

Maddeleri geliştirmenin çözüm sağlayacağı başlıca sorunlar olarak tespit edilmiştir. Bu maddelere ek olarak, günümüzde 'uzaktan/görev bazlı çalışma, esnek mesai saatleri' gibi uygulamaların yaygınlaştığı göz önünde bulundurulduğunda çalışma ayrıca önem kazanmaktadır. Bu çalışma yöntemlerinde departman kullanıcılarının mesai saatleri de farklılık gösterebilmektedir. Sınıflandırma kullanıcılarının mesai saatleri dışında sisteme yüklenen faturalar da sonraki mesai başlangıcına kadar sınıflandırma aşamasında bekleyerek ekstra gecikmelere sebebiyet verebilmektedir. Bu süre zarfında kayıt kullanıcıları işlem yapamamaktadır. Geliştirmeyle birlikte bu problemin de önüne geçilmesi hedeflenmektedir.

Şekil 2: Mevcut Yazılımın Yaşam Döngüsü



Kaynak: Yazar tarafından derlenmiştir.

Şekil 2'de mevcut sistemde kullanılmakta olan iş akışı şeması yer almaktadır.

1.2. ARAŞTIRMA SORULARI

Mevcut uygulamada sınıflandırma işleminin belli bir oranda (yaklaşık %7) kural tabanlı kodlama ile otomatik yapılabildiği bilinmektedir. Bu bilgi göz önünde bulundurularak belirlenen araştırma soruları aşağıdaki gibidir:

1- Faturaların otomatik sınıflandırılmasında başarı oranını yükseltmek için uygulanabilecek yöntemler nelerdir?

2- Faturaların makine öğrenmesi yöntemi ile sınıflandırılması mümkün müdür?

Birinci araştırma sorusu için mevcut programda aktif bulunan kural tabanlı işleyişe yeni eklemeler yapılarak başarı oranının artırılmasının mümkün olup olmadığı, bu yöntemin hangi şekillerde uygulanabileceği incelenmiştir.

İkinci araştırma sorusu için ise uygulamanın veri tabanında kayıtları bulunan geçmiş yıllara ait sınıflandırılmış fatura verileri incelenerek bir uygulama geliştirilmiş, verilerin toplanması, düzenlenmesi, uygulama haline getirilmesi ve sonuçları çalışmada detaylı olarak paylaşılmıştır.

1.3. LİTERATÜR ÇALIŞMASI

Yangın ve diğerleri (2021) deki çalışmalarında ABD'de 130 farklı hastanenin kayıtlarında yer alan 1999-2008 yılları arasında karşılaşılmış toplam 70 bin sağlık vakıası kayıtlarından elde ettikleri veri setini kullanarak hastaların diyabet rahatsızlıklarını sınıflandırmışlardır. Bu çalışmada sınıflandırma işlemi için veri setini Karar Ağaçları, K-En Yakın Komşu, Lojistik Regresyon, Naive Bayes ve Rassal Orman algoritmaları olmak üzere 5 farklı sınıflandırma algoritması ile eğittiklerinde en iyi ve en doğru sınıflandırma başarısını Rassal Orman algoritması ile elde etmişlerdir. Bağımsız 22 değişkenin yer aldığı veri setinin kullanıldığı çalışmada bireylerin diyabet rahatsızlığı olma olasılıklarını %84,78 doğruluk oranı ile tahmin edilebileceği sonucuna ulaşmışlardır.

Serap KAZAN ve diğerleri (2019) daki çalışmalarında bir e-ticaret sitesinden elde ettikleri ürün bilgileri etiketlenerek oluşturulmuş veri setini eğiterek ürünleri 9 farklı sınıfa ayıracak şekilde tahminleme yapmayı amaçlamışlardır. Çalışmada veri

seti farklı makine öğrenmesi algoritmaları ile eğitilerek algoritmaların ürünlerin kategorilerini doğru tahminleme başarıları karşılaştırılmıştır. Eğitim aşamasında denedikleri makine öğrenmesi algoritmaları Karar Ağacı, Random Forest, Multinominal Naive Bayes (Multinominal NB): Lojistik Regresyon, Yapay Sinir Ağları (YSA) ve Destek Vektör Makineleri (DVM) algoritmalarıdır. Çalışma sonunda ürünler 6 kategoride sınıflandırılmış ve sonuçların başarı oranları birbirleriyle karşılaştırılmıştır. Birbirine yakın özellikleri taşıyan veri setlerinde sınıflandırmadaki başarı oranının diğer veri setlerinde elde edilen başarı oranlarına göre daha düşük çıktığı gözlemlenmiştir. Çalışmada kullanılan sınıflandırma algoritmalarının ortalama başarıları karşılaştırıldığında Destek Vektör Makineleri (DVM) ve Yapay Sinir Ağları (YSA) ((Multi-Layer Perceptron (MLP) (aktivasyon='logistic sigmoid')) algoritmalarının yaklaşık olarak %97 başarı oranları ile diğer yöntemlere göre daha yüksek sonuçlar verdiği görülmüştür.

Filiz ve diğerleri (2017) deki çalışmalarında BIST-50 endeksi üzerinde etkisi bulunan faktörleri baz alarak oluşturdukları veri seti ile makine öğrenmesi yöntemleri ve yapay sinir ağları ile sınıflandırma çalışmaları yapmışlardır. Çalışmada farklı sektörlerden firmaların belli oranlarda etkisiyle oluşan İstanbul Borsası'nda işlem gören en büyük piyasa değerine sahip 50 şirketin göstergesi olarak kabul edilen BIST-50 endeksinin yönünün, artış ve azalışların tahminlenmesi bu sayede yatırımcılara doğru pozisyon alarak kar elde edebilmeleri doğrultusunda fikir verebilecek bir sınıflandırma işlemi yapılması amaçlanmıştır. Çalışma kapsamında veri setine dahil edilen girdilerin sonuca ne düzeyde etki ettikleri de incelenmiş ve girdilerin önem sırası da ayrıca tespit edilmiştir. Sonuç olarak endeksi etkileyen girdilerin değerlerinin bilinmesi durumunda yön tahminini %89-%93 oranları arasında bir başarı ile doğru sınıflandırma sonucu elde etmişlerdir. K-NN, Naive Bayes ve C4.5 algoritmaları ile yapılan bu çalışmada en yüksek başarı oranını veren algoritma %92.71 ile bir karar ağacı algoritması olan C4.5 algoritması olmuştur.

Bardelli ve diğerleri (2020) deki çalışmalarında elektronik faturalarda yer alan bilgileri kullanarak muhasebecilerin işini basitleştirecek akıllı bir sistem geliştirmeyi amaçlamışlardır. Çalışmada, muhasebe işlemleri yapılırken özel kodlar halinde sınıflandırılan faturaların hesap ve vergi kodlarının tahmin edilebilmesini sağlayacak çok sınıflı bir sınıflandırma algoritması önerilmiştir. Farklı sınıflandırma

algoritmalarının başarı oranlarının karşılaştırıldığı çalışmada en yüksek sınıflandırma başarı oranının Rassal Orman algoritması ile elde edildiği gözlemlenmiştir.

Aydın (2018) deki çalışmasında İzmir ilinde yer alan aktif itfaiye istasyonlarını baz alarak yeni istasyonların kurulması gereken bölgelerin tespit edilmesinin amaçlandığı bir sınıflandırma çalışması yapmıştır. Dünya standartlarına göre itfaiye istasyonlarının, istasyona yakın bölgelerde meydana gelen yangınlara 5 dakika içerisinde müdahale edebiliyor olması gerektiği göz önünde bulundurularak gerçekleştirilen çalışma, coğrafi bilgi sistemleri araçları ile elde edilen veri seti kullanılmış ve makine öğrenmesi yöntemlerinden faydalanılmıştır. Veri setinde itfaiye araçlarının belirlenen bölgelere ulaşım süreleri, bölgesel nüfus yoğunlukları, belirlenen bölgelerde itfaiye istasyonunun bulunup bulunmadığı ve bölgelere ulaşması gereken ortalama ana ve yardımcı araç sayısı verilerinin yer aldığı çalışmada sınıflandırma algoritmalarından Random Forest (RF), K-Nearest Neighbourhood (KNN), Naive Bayes (NB), Gradient Boosted Trees (GBT), Decision Tree (J48), Destek Vektör Makineleri (SVM) algoritmaları kullanılmıştır. Çalışmada, kullanılan makine öğrenme algoritmalarının sonuçları karşılaştırılmış ve en yüksek doğrulukla sınıflandırma işleminin %93,84 doğruluk oranıyla Random Forest algoritması ile elde edildiği paylaşılmıştır. Çalışmanın sonuç bölümünde ise itfaiye istasyonlarının hizmet alma durumu yönünden makine öğrenmesi yöntemleri ile sınıflandırılmasının mümkün olduğu kapsama alanı ile %85,43 oranında uyumluluk sağladığı bilgisi paylaşılmış ve gelecekte konuyla ilgili yapılacak çalışmalarda veri setine bölgelerdeki yolların genişlikleri ve bölgelerdeki trafik yoğunlukları gibi verilerin eklenmesi durumunda daha başarılı sonuçlar elde edilebileceği önerisinde bulunulmuştur.

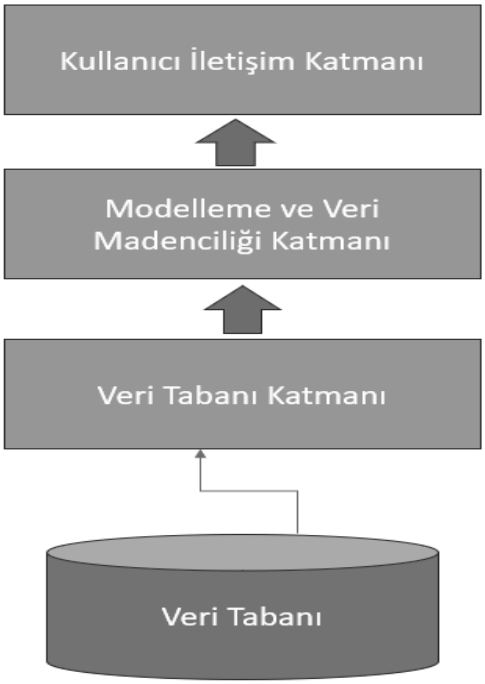
İKİNCİ BÖLÜM
KARAR DESTEK SİSTEMLERİ, ELEKTRONİK FATURA KAVRAMI VE FATURALARIN SINIFLANDIRILMASINDA UYGULANABİLECEK YÖNTEMLER

2.1. KARAR DESTEK SİSTEMLERİ

Karar Destek Sistemleri, farklı veri kaynaklarından topladığı bilgileri çeşitli yöntemlerle düzenleyip kararı modelleyerek ve verileri analiz eden ve elde ettiği sonuçları raporlayarak karar verme aşamasında karar vericiye destek sağlayan bilgisayar tabanlı sistemlerdir (Çelik, 2009). Karar vericinin verdiği kararın doğruluğu, onun tecrübesi, bilgi birikimi ve kişisel yetenekleriyle birlikte elindeki veri kümesinin doğruluğu ve yeterliliği ile de doğrudan ilişkilidir. Bir diğer ifadeyle kararın başarısında, verilerin doğru depolanıp sınıflanması, işlenmesi ve yorumlanması büyük önem taşımaktadır (Yaralıoğlu, 2004).

Karar destek sistemleri temel olarak veri tabanı katmanı, modelleme ve veri madenciliği katmanı ve kullanıcı iletişim katmanı olmak üzere 3 katmandan oluşmaktadırlar.

Şekil 3: Karar Destek Sistemi Katmanları



Kaynak: Yazar tarafından derlenmiştir.

Şekil 3'te karar destek sistemlerini oluşturan katmanlar yer almaktadır. Veri tabanı katmanı, verilerin herhangi bir işlem yapılmamış ham bir şekilde depolandığı katmandır. Modelleme ve veri madenciliği katmanı, veri tabanındaki ham verilerin çeşitli işlemlerden geçirilerek belli amaçlar doğrultusunda fayda elde edilebilecek, anlamlı bir hale dönüştürüldüğü, modellendiği katmandır. Kullanıcıların oluşturulan modelden anlamlı çıkarımlar yapabilmeleri için geliştirilen arayüzler ve hazırlanan raporlar ise kullanıcı iletişim katmanıdır. Karar vericiler, karar verme aşamasında kullanıcı iletişim katmanını kullanarak daha hızlı ve daha sağlıklı kararlar alabilmektedirler.

2.2. ELEKTRONİK FATURA KAVRAMI

Türkiye'de 397 Sıra Numaralı Vergi Usul Kanunu (VUK) Genel Tebliği ile hayata geçirilen ve 5 Mart 2010 tarihi itibariyle aktif olan elektronik fatura (e-Fatura) veri formatı ve standardı Gelir İdaresi Başkanlığı (GİB) tarafından belirlenen, 213 sayılı VUK hükümleri gereği bir faturada bulunması gerekli olan bilgileri kapsayan,

satıcı ve alıcı arasındaki iletiminin merkezi bir platform (GİB) üzerinden gerçekleştirildiği bir elektronik belgedir. E-fatura uygulamasıyla tek format ve standarda göre alıcı ve satıcı arasında maliyet ve zaman tasarrufu sağlayan güvenli bir sistem oluşturulması amaçlanmıştır. Bu doğrultuda yurtdışı e-fatura uygulamaları araştırılmış ve başarılı sistemler dikkate alınarak GİB tarafından açık kaynak bir standart olan Extensible Markup Language (XML) tabanlı ticaret hayatının bütün aşamalarını barındıracak etiketlemeleri içeren uluslararası bir standart olan UBL-TR benimsenmiştir. E-fatura belgesi, yeni bir belge türü olmamakla birlikte kağıt fatura belgesi ile aynı nitelikleri taşımaktadır (GİB, 2023).

E-fatura uygulamasında faturalar elektronik ortamda, XML formatında oluşturulmaktadır. XML, Türkçede genişletilebilir işaretleme dili anlamına gelmektedir. Bu formatta oluşturulan faturalar, insanlar tarafından kolaylıkla okunabilmekle birlikte veriler etiketli bir şekilde yer aldığı için içerdikleri veriler de bilgisayar uygulamaları ile basit bir şekilde çekilebilmekte ve veri tabanlarına kaydedilebilmektedir.

Çalışmada ele alınan fatura yönetimi uygulamasında XML formatında sisteme yüklenen elektronik faturaların verileri otomatik olarak veri tabanına kayıt edilmektedir.

2.3. KURAL TABANLI KODLAMA YÖNTEMİ

Kural tabanlı kodlama ile sınıflandırma; mevcut sistemde de kullanılmakta olan ve çalışmanın önceki aşamalarında değinilen, aktif olarak yaklaşık %7 başarı oranıyla çalışmakta olan yöntemdir. Bu yöntemde kullanıcılar tarafından yazılım ekibiyle paylaşılan kurallar sisteme kod güncellemesi yapılarak entegre edilmiştir. Bu sayede sisteme yüklenen faturalardan bu kuralların kapsamına girenler kuralda belirtilen sınıfa otomatik olarak atanmaktadır. Şekil 4'te yer alan tabloda kullanıcılar tarafından belirtilen kural örnekleri yer almaktadır.

Şekil 4: Sınıflandırma Kuralları Örnekleri

Kullanıcı Tarafından Belirtilen Kural	Yazılımcı Tarafından Kuralın Kod Olarak Sisteme Eklenmesi
'xyz' firmasına ait bütün faturalar 'ithalat' sınıfı faturasıdır	Eğer firma ismi = 'xyz' ise fatura sınıfı ='ithalat' tır.
Açıklama kısmında 'abc' metni yer alanlar 'ihracat' sınıfı faturasıdır	Eğer açıklama 'xyz' içeriyorsa fatura sınıfı ='ihracat' tır.
'xyz' firmasına ait faturalardan açıklama kısmında 'abc' metni yer alanlar 'hizmet' sınıfı faturalarıdır.	Eğer firma ismi = 'xyz' ve açıklama 'abc' içeriyorsa fatura sınıfı ='hizmet' tir.
'xyz' firmasına ait faturalardan açıklama kısmında 'abc' metni yer almayanlar 'mal' sınıfı faturalarıdır.	Eğer firma ismi = 'xyz' ve açıklama 'abc' içermiyorsa fatura sınıfı ='mal' dır.

Kaynak: Yazar tarafından derlenmiştir.

Şekil 4'te mevcut sistemde kullanılmakta olan ve yeni kural olarak eklenebilecek kural tipi örnekleri yer almaktadır. Sınıflandırma işleminin kural tabanlı kodlama yönteminin tercih edilmesi durumunda aşağıdaki maddelere göre hareket edilmelidir;

- Kurallar kesin olmalıdır. Sisteme eklenen kural yanlış sınıflandırma yapmamalıdır.
- İlerleyen süreçte değişiklik göstermeyeceği varsayılan kurallar sisteme eklenmelidir.
- Kullanıcılar, işlem yaparken tespit ettikleri kesin kuralları belli periyotlarla yöneticilerine iletmelidirler.
- Farklı kullanıcılar tarafından belirlenen kurallar birleştirilerek yazılım ekibine iletilmelidir.

Böylece yazılım ekibine iletilen kurallar, yazılım ekibi tarafından güncelleme ve test işlemleri yapıldıktan sonra sisteme dahil edilebilecektir. Bu şekilde yapılan her başarılı güncelleme, eklenen her doğru kural otomatik sınıflandırmadaki başarıya olumlu etki edecektir. Bu yöntemin öngörülen dezavantajları ise aşağıda maddeler halinde sıralanmıştır;

- Yeni kuralların tespiti ve raporlanması kullanıcılar için yeni bir iş yükü oluşturacaktır,
- Kullanıcılar tarafından raporlanan yeni kuralların birleştirilerek belirli periyotlarla yazılım ekibine iletilmesi gerekecektir,
- Yeni kuralların sisteme dahil edilmesi için yazılım ekibi tarafından belirli aralıklarla geliştirme, test ve canlıya alma süreçlerinin tekrarlanması gerekecektir,

- Çok sayıda program kullanıcısının ve yazılım ekibinin iş sürecine dahil olması gerekecektir.

Kural tabanlı kodlama yönteminin uygulanmasında tercih edilebilecek alternatif bir yol da aşağıda ayrıca belirtilmiştir.

Kuralların Veri Tabanı Üzerinden Sisteme Entegre Edilmesi: Bu yöntem veri tabanına kuralların yer alacağı yeni bir tablo eklenerek ve kod tarafında sınıflandırmanın bu tabloya göre yapılacağı şekilde güncelleme yapılarak uygulanabilir. Bu yöntemin tercih edilmesi durumunda yeni kurallar yazılım ekibi tarafından veri tabanına eklenerek sisteme entegre edilebilecektir. Bu sayede yeni kural eklenmesi işlemi için yazılım güncellemesi yapılmasına gerek kalmayacaktır.

Ayrıca veri tabanına yeni kural eklenmesi işlemi için yeni bir arayüz geliştirilerek bu yeni kuralların kullanıcılar tarafından eklenmesi sağlanabilecektir. Bu ekrana erişim yetkisi verilen kullanıcılar tespit edilen yeni kuralları yazılım ekibine gerek kalmadan sisteme entegre edebileceklerdir.

2.4. MAKİNE ÖĞRENMESİ YÖNTEMİ

Makine öğrenmesi uygulamalarının temeli, makinelerin de insanlara benzer bir şekilde verilerden bilgi ve tecrübeye dayalı analizler çıkarabileceği düşüncesine dayanmaktadır (Michalski ve Kodratoff, 1990). Makine öğrenmesi büyük ölçekli verilerden anlamlı analizler yapılabilmesini kolaylaştıran, istatistiksel hesaplamalara dayalı bir yöntemdir. Büyük miktarlardaki verilerin ilk bakışta tespit edilmesi mümkün olmayan ilişkilerinin analiz edilebilmesine olanak sağlayan bu yöntem aynı zamanda yapay zekanın bir alt dalı olarak kabul edilmektedir. Makine öğrenmesi yöntemi verilerdeki gözlemlenmesi güç olan davranış ve kuralları ortaya çıkararak daha hızlı ve kolay bir şekilde varsayım ve çıkarımlarda bulunabilmemizi sağlar (Alpaydin, 2004).

Günümüzde makine öğrenmesinden tıp, finans, online alışveriş, borsa, bankacılık, sosyal medya platformları, reklamcılık gibi verinin olduğu pek çok farklı alanda faydalanılmaktadır. Makine öğrenmesi çalışmasının uygulanmasında izlenen adımlar ise şekil 5'te yer alan görseldeki gibidir.

Şekil 5: Makine Öğrenmesi Çalışmalarında İzlenen Adımlar



Kaynak: Yazar tarafından derlenmiştir.

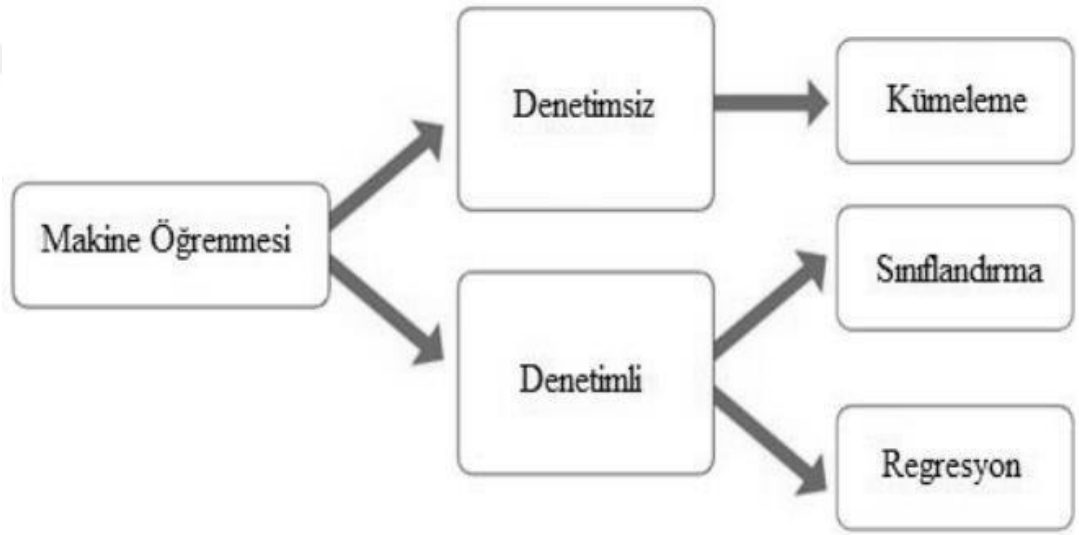
Şekil 5'te makine öğrenmesi çalışmalarının aşamaları yer almaktadır. Bu aşamalar çalışmanın bu bölümünde açıklanmış ve proje aşamasında da bu adımlar izlenmiştir.

- Veri Toplama: Bu aşama, çalışmada kullanılacak olan verilerin elde edilmesi aşamasıdır. Makine öğrenmesinde veri miktarı ve çeşitliliği arttıkça modelden daha sağlıklı sonuçlar elde edilebilmektedir. Bu nedenle çalışmayla alakalı ulaşılabilen bütün verilere ulaşılmalı ve bu veriler uygun bir şekilde bilgisayar ortamına taşınmalıdır.
- Veri Ön İşleme: Veri kalitesinin artırılması ve analiz sonuçlarındaki başarı oranının yükseltilmesi amacıyla veri üzerinde veri entegrasyonu, veri temizleme, veri dönüştürme ve veri indirgeme işlemlerinin yapıldığı aşamadır. Bu aşamada farklı kaynaklardan toplanan veriler bütünleştirilerek tek bir kaynakta birleştirilir, verideki parazitler temizlenir, eksik veriler tamamlanır ve tutarsızlıklar ortadan kaldırılır. Verideki sıra dışı değerler tespit ve analiz edilir. Daha sonra normalleştirme, bölümlere ayırma/birleştirme gibi teknikler uygulanarak veri, modelleme aşaması için hazırlanır. Son olarak veri hacmini küçültmek amacıyla örnekleme, faktör analizi, boyut indirgeme gibi teknikler uygulanarak bu aşama tamamlanır (Akpınar, 2014).
- Eğitim: Hazırlanmış olan verideki korelasyonların algoritmalar yardımıyla ortaya çıkarıldığı aşamadır. Bu aşamada bir model oluşturularak veri ön işleme aşamaları tamamlanmış olan verinin bir kısmı bu modelde eğitim verisi olarak

kullanılır. Model, verideki pozitif ve negatif ilişkileri öğrenip anlamlı çıkarım ve tahminlemeler yapması için bu eğitim verisi ile eğitilir.

- Test: Modelin başarısının ölçüldüğü aşamadır. Verinin eğitim verisi olarak kullanılmayan kısmı bu aşamada test verisi olarak kullanılır. Eğitim verisi ile eğitilmiş olan modele daha önce karşılaşmamış olduğu test verileri verilerek modelin bu veriler üzerine tahminleme yapması beklenir. Bu sayede modelin tahminlemelerindeki başarı oranı tespit edilebilmektedir.

Şekil 6: Makine Öğrenmesi Kategorileri



Kaynak: Sakal, 2021.

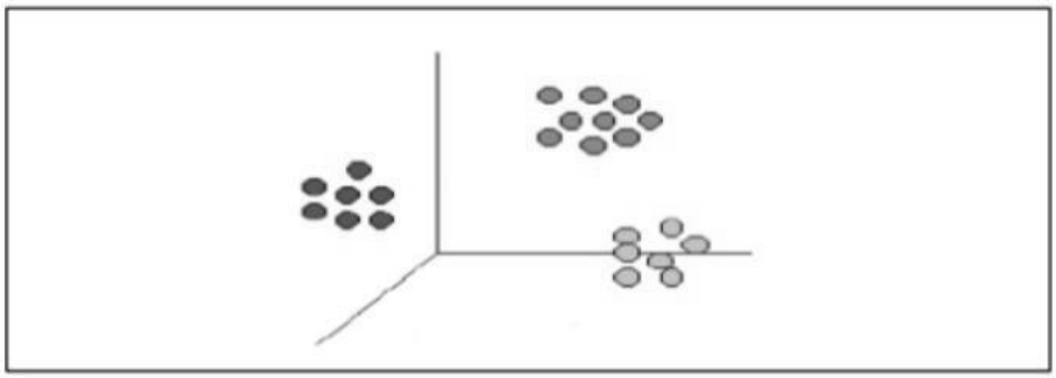
Şekil 6'da makine öğrenmesinin kategorilere ayrılmış gösterimi yer almaktadır. Bu çalışmada gerçekleştirilen sınıflandırma projesi de denetimli makine öğrenmesi kategorisi kapsamında yer almaktadır.

- Denetimli Öğrenme (Supervised Learning): Modelin, daha önceden etiketlenmiş veriler üzerinden eğitildiği makine öğrenmesi türüdür. Eğitim aşamasında makineye girdiler ve bu girdilerden hangi sonucu elde etmesi gerektiği verilir. Daha sonra makineye test verileri girdileri verilerek sonuç tahmini yapması istenir. Makinenin tahminlerindeki doğruluk oranına göre başarısı tespit edilir.

Denetimli öğrenme algoritmalarına örnek olarak doğrusal ve lojistik regresyon modelleri, karar ağaçları, destek vektör makineleri ve yapay sinir ağları verilebilir (Niculescu-Mizil ve diğerleri, 2005).

- Yarı Denetimli Öğrenme: Yarı denetimli öğrenme, modelin etiketli verilerle birlikte etiketsiz verilerin de yer aldığı bir eğitim verisiyle eğitildiği makine öğrenmesi türüdür.
- Denetimsiz Öğrenme (Unsupervised Learning): Denetimsiz öğrenmede modelin eğitilmesi işlemi tamamen etiketlenmemiş verilerden oluşan eğitim verisiyle gerçekleştirilir. Bu yöntem aynı zamanda yoğunluk tahmini olarak da bilinmekte olup, benzer özelliklere sahip verilerin kümelenmesi amacıyla kullanılmaktadır. Modelin başarısı, kümelenen verilerin aynı sınıftaki verilere ne ölçüde benzediklerinin ve diğer kümelerdeki verilerden ne ölçüde farklı olduklarına göre tespit edilmektedir (Berry ve Linoff, 2004).

Şekil 7: Kümelere Ayrılmış Verilerin Koordinat Düzleminde Gösterimi



Kaynak: Sariman, 2011

Şekil 7'de kümeleme işlemi yapılmış verilerin 3 eksenli olarak görselleştirilmesi örneği yer almaktadır.

- Pekiştirmeli Öğrenme: Pekiştirmeli öğrenmede model deneme/yanılma yöntemiyle eğitilmektedir. Model yanlış tahmin yaptığında tahmininin yanlış olduğu modele bildirilir fakat doğrusunun ne olması gerektiği bildirilmez. Model doğru tahminde bulunana kadar farklı ihtimalleri araştırmak ve denemek zorundadır (Fatima ve Pasha, 2017). Bu şekilde doğru sonuca nasıl ulaşacağını model kendisi öğrenmektedir.

2.4.1. Makine Öğrenmesi Algoritmaları

Makine öğrenmesi çalışmalarında kullanılan çok sayıda farklı algoritma bulunmaktadır. Çalışmanın bu aşamasında makine öğrenmesi projelerinde yaygın olarak kullanılmakta olan bazı sınıflandırma ve regresyon algoritmaları hakkında bilgiler verilmiştir.

- Lojistik Regresyon: Lojistik regresyon, ikili veya kategorik sonuçları tahmin etmek için kullanılan istatistiksel bir makine öğrenmesi algoritmasıdır. Bağımlı ve bağımsız değişkenler arasındaki ilişkiyi açıklamak için kullanılmaktadır. Lojistik regresyonda amaç, bağımsız değişkenlerin değerlerine dayalı olarak ortaya çıkan sonucun olasılığını tahmin etmektir.

Lojistik regresyonun sıralı lojistik regresyon, multinominal lojistik regresyon ve ikili lojistik regresyon olmak üzere üç farklı türü vardır. Bunlardan sıralı lojistik regresyon kategoriye göre sınıflandırmada belli periyotlarla sıralı tahminler veren çalışmalarda kullanılmaktayken multinominal lojistik regresyon belli bir sırası olmayan en az üç veya daha fazla kategorik tahminin olduğu durumlarda kullanılmaktadır. Lojistik regresyonun bir diğer türü olan ikili lojistik regresyon ise kategorik olarak sadece iki farklı sonucun olduğu durumlarda kullanılmaktadır (Aydın,2020).

- Naive Bayes: Sınıflandırma işlemini, sınıflarda yer alan değişkenlerin farklı özelliklere sahip olduğu varsayımından hareketle yapan ve verideki farklı sınıfların yapılarını öğrenip yeni bir veri eklendiğinde bu yeni verinin hangi sınıfa ait olduğunu öğrenmiş olduğu sınıf yapılarına göre tespit eden basit ve etkili bir yöntemdir (Silahtaroğlu,2013). Bayes teoremi, hesaplamaların koşullu olasılıklar ve bağımlı değişkenler üzerinden yapıldığı bir teoremdir. Kavram olarak koşullu olasılık, yeni oluşacak bir sonucun gerçekleşmiş olaylardaki sonuçlar üzerinden tahminlenmeye çalışılması yöntemine dayanmaktadır. Örnek olarak bir maçın sonucunun tahminlenmesinde farklı etkenler de göz önünde bulundurularak takımların eski maçlarının sonuçlarına bakılması verilebilir (Kavuncu, 2018).

Verilerin eğitilmesi işleminin hızlı gerçekleşmesi ve tahminlemedeki başarı oranının yüksek olması Naive Bayes algoritmalarının avantajlı tarafları olarak kabul edilmekle birlikte, dezavantaj olarak ise sınıflandırma problemlerindeki karmaşıklığın

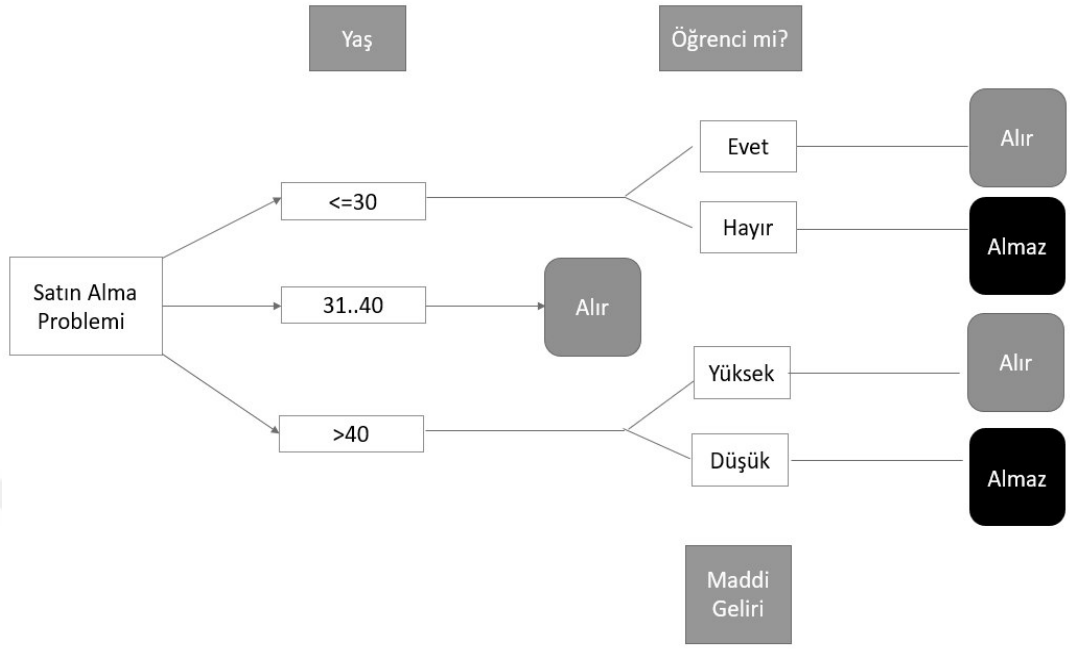
yüksek olduğu durumlarda bu algoritmanın yetersiz kaldığı karşımıza çıkmaktadır (Uzun, 2007).

- Karar Ağaçları: Karar ağaçları, karar verme alanında en sık kullanılan klasikleşmiş yöntemlerden birisidir. Verilen kararlar önceden belirlenen olasılıklara göre şekillenmektedir ve her bir karar farklı karar noktalarını etkilemektedir. Karar verme sürecinin görselleştirilmesi ve hesaplanması karar ağaçları yöntemi ile basit bir şekilde yapılabilmektedir (Akpınar, 2014).

Karar ağaçları da tıpkı doğadaki ağaçlar gibi kök, dal ve yapraklardan oluşmaktadır. Kök kısmında heterojen bir şekilde bir arada bulunan girdilerin dal ve yapraklara doğru gidildikçe daha homojen sınıflara ayrılması amaçlanmaktadır. Karar ağaçları, farklı veri kümelerini, belli bir hedef değişkeni doğrultusunda, karar verme kuralları yardımıyla birbirinin benzeri olan yani daha homojen alt gruplara ayırır (Linoff ve Berry, 2004).

Karar verme sürecinde temel adımlar birbirine çok benzerdir. İlk aşamada kök düğüm oluşturulur. Daha sonra bütün veriler belirlenen karar verme noktalarından geçerek sınıflara ayrılır. Daha sonra aynı sınıfa düşen verilerin birbirleriyle benzerlikleri kontrol edilerek o sınıfın daha fazla bölünüp bölünmemesine karar verilir. Hem bölünme aşamasında hem de bölünme sonrası homojenlik kontrolü aşamasında faydalanılan çok sayıda farklı yöntem bulunmaktadır (Murphy,2012).

Şekil 8: Müşteri Tablet Satın Alır/Almaz Probleminin Karar Ağacı Şeması Örneği



Kaynak: Yazar tarafından derlenmiştir.

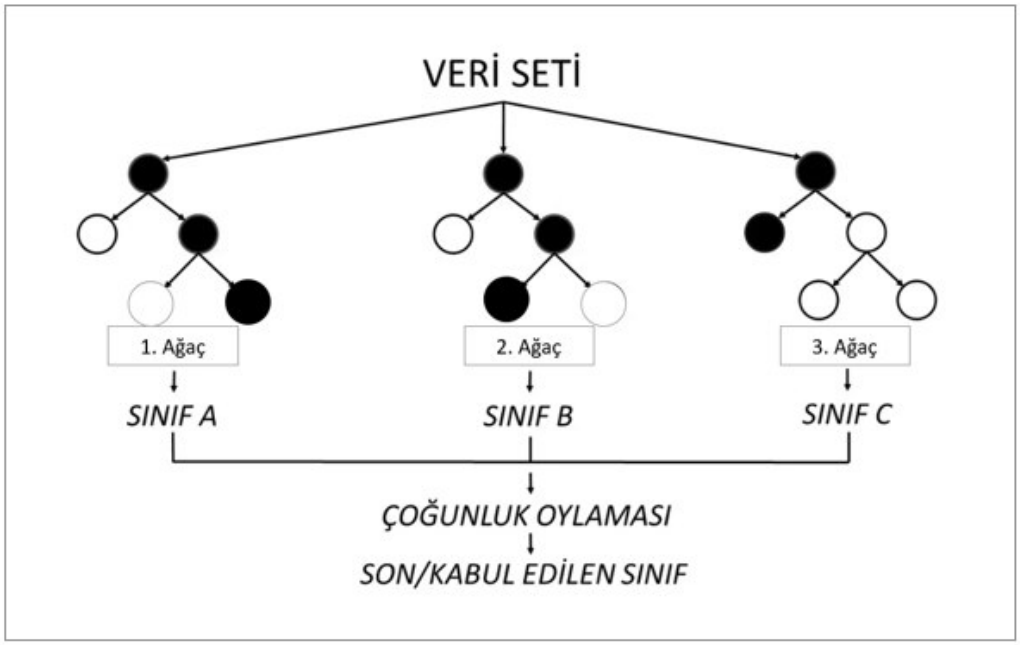
Şekil 8'de bir mağazaya giren müşterilerin yaş, öğrencilik durumu ve maddi geliri bilgilerine göre tablet satın alıp almayacaklarının tahminlenmesi amacıyla oluşturulan karar ağacı şeması örneği yer almaktadır.

- Rassal Orman: Denetimli makine öğrenmesi yöntemlerinden birisidir. Boyut sayısının çok yüksek olduğu verilerin eğitiminde kullanılır. Karar ağaçlarında sık rastlanan problemlerden birisi olan aşırı öğrenme probleminin aşılmasında çözüm olarak bu yöntem kullanılabilmektedir. Birbirinden bağımsız karar ağaçlarının eğitim ve test verilerindeki girdileri örneklemeye dayanmaktadır. Sınıflandırma çalışmalarında kullanıldığında bağımsız karar ağaçlarından birer tane sınıf oyu alarak oy çokluğuna göre sınıflandırma işlemini yapmaktadır. Regresyon problemlerinde ise tahmin ortalamalarına göre işlem yapmaktadır (Dangeti, 2017).

Rassal Orman yönteminde veri kümesi de değişken kümesi de rastgele kullanılarak ağaçlar oluşturulur. Rastgele veri ve değişkenlerde oluşan bu rastgele ağaçlar bir araya geldiğinde rastgele bir ormanı tanımlar. Bu yöntemde sonuç iki farklı inanca dayanır. Bunlardan birincisi her bir ağacın verinin bir bölümü için doğru tahminde bulunmasıdır. İkincisi ise değişik yerlerde hatalarla karşılaşıldığıdır. Bu

şekilde karar, karar ağaçları arasındaki oylama sonucuna göre verilmektedir (Gollapudi ve Laxmikanth 2016).

Şekil 9: Rassal Orman Algoritmasının Sınıflandırma Yapısı Görseli



Kaynak: Yazar tarafından derlenmiştir.

Şekil 9'da rassal orman algoritması kullanılarak yapılan sınıflandırma çalışmasında oluşturulan karar ağaçlarının yapısının temsili gösterimi yer almaktadır.

- Destek Vektör Makineleri: Yaygın olarak sınıflandırma problemlerinde kullanılır. Gözetimli öğrenme yöntemlerinden birisidir. Bir düzlem üzerindeki farklı sınıflara ait olan noktaları, bu noktalara maksimum uzaklıktaki bir çizgi ile bölmeyi amaçlar. Karmaşık ama veri miktarının fazla büyük olmadığı çalışmalar için uygun yöntemlerden birisidir.

Genellikle makine öğrenimi yöntemi olarak adlandırılan karar destek sistemleri, belirli durumlarda açıkça tanımlanamayan işlemler için, verilen girdilere karşılık gelen çıktının, mevcut bilgi ve deneyimlerin yardımıyla tahmin edildiği bir işlemdir (Alpaydın, 2004).

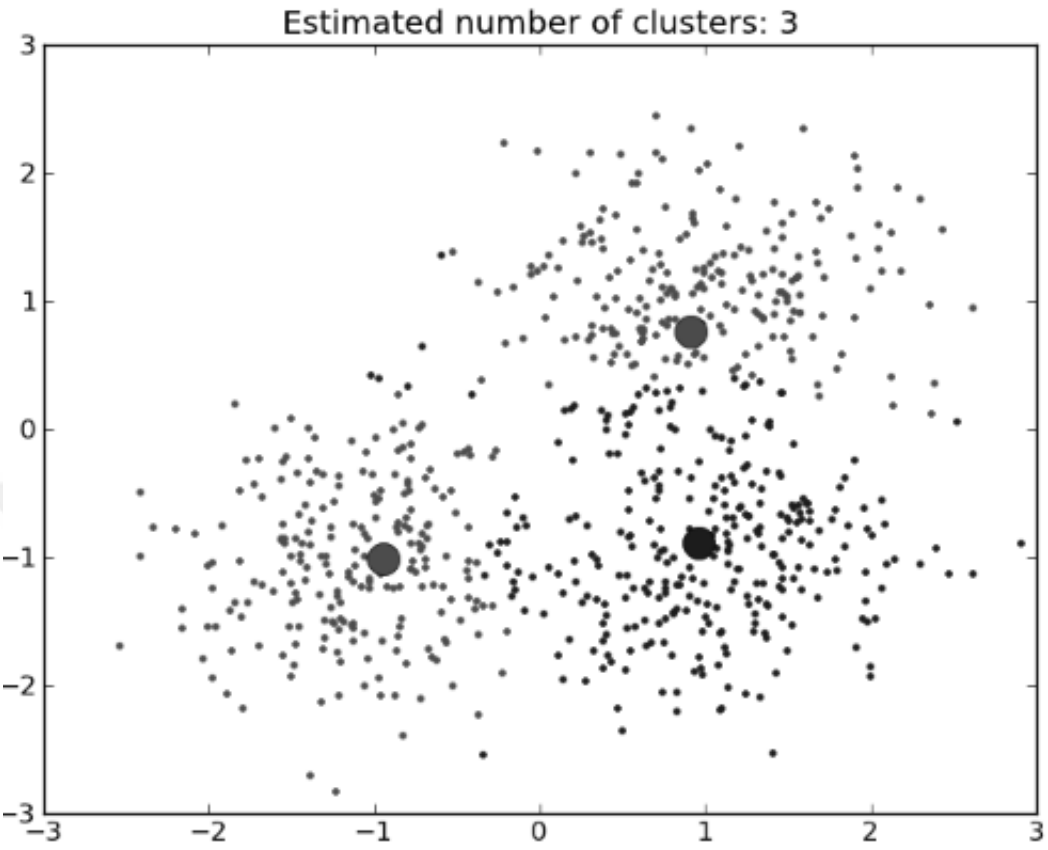
Makine öğrenmesi çalışmalarında çoğu zaman büyük eğitim verisiyle çalışma isteği, yakınsama oranının düşük olması, aşırı öğrenme/eksik öğrenme gibi sorunlarla

karşılaşılabilmektedir (Lu ve diğerleri, 2002). Yapısal riskleri minimize etme yöntemini baz alarak çalışan Destek Vektör Makineleri bu sorunları aşabilmektedir. Sınıflandırma çalışmalarında uygulanan Destek Vektör Sınıflandırma ve regresyon çalışmalarında uygulanan Destek Vektör Regresyon olmak üzere iki çeşidi bulunmaktadır. Bu yöntem veri sayısı az olmakla birlikte boyutu yüksek olan verilerle yapılan çalışmalarda da başarılı sonuçlar verebilmektedir (Shen ve diğerleri, 2004).

- K- En Yakın Komşu Algoritması (K-NN): Verilerin sınıfını tespit etmek amacıyla regresyon ve sınıflandırma çalışmalarında kullanılan denetimli öğrenme yöntemlerden birisidir. Sınıflandırma işlemini, bir düzlem üzerindeki noktaların birbirine olan mesafelerini baz alarak yapmaktadır (Özkan, 2008).

K-NN, bir öğrenme aşaması bulunmayan, tembel (lazy) bir öğrenme yöntemidir. Eğitim verilerini öğrenme yöntemiyle değil veri setini ezberleme yöntemiyle çalışır. Bir verinin sınıfı bu yöntemle tahmin edilmeye çalışıldığında, sınıfı tahmin edilmeye çalışılan verinin veri setinin tamamındaki komşularına bakarak tahmin sonucu vermektedir (Karadağ ve diğerleri, 2020).

Şekil 10: 3 Sınıflı Sentetik Bir 2b Veri Kümelerinde Knn Kümelemesi Örneği



Kaynak: https://scikitlearn.sourceforge.net/0.6/modules/clustering.html, 12.06.2022.

Şekil 10'da K-NN algoritması kullanılarak gerçekleştirilen bir kümeleme çalışmasının 2 boyutlu olarak gösterimi yer almaktadır.

K-NN algoritmasıyla çalışılırken ilk olarak tahmin yaparken kaç tane veriye bakılacağı yani algoritmanın isminde de bulunan 'k' değeri belirlenir. Daha sonra tahmin için veri girildiğinde veriye en yakın 'k' adet veri belirlenerek girilen veri ile aralarındaki mesafeler hesaplanır. Mesafelerin hesaplanmasında yaygın kullanılan fonksiyon "Öklid" fonksiyonu olmakla birlikte "Manhattan", "Minkowski" ve "Hamming" gibi alternatif fonksiyonlar da kullanılabilmektedir. Mesafelerin hesaplanmasının sonrasında ise hesaplanan değerler sıralanarak girilen veri kendisine hangi sınıf daha yakınsa o sınıfa atanır (Karadağ ve diğerleri, 2020).

ÜÇÜNCÜ BÖLÜM
ARAŞTIRMA YÖNTEMİ VE SİSTEM GELİŞTİRME YAŞAM DÖNGÜSÜ

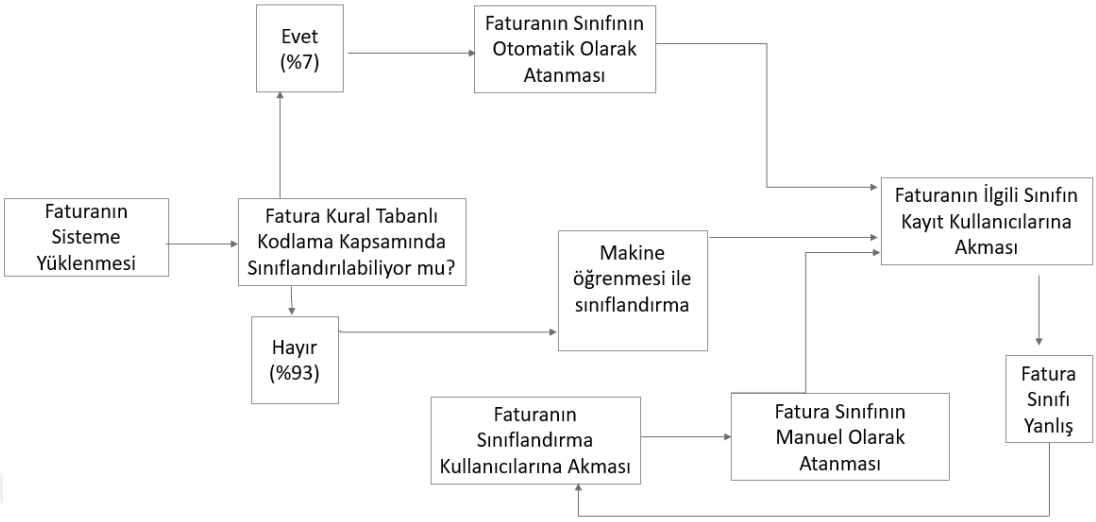
3.1. ARAŞTIRMA YÖNTEMİ

Çalışmada kapsamında mevcut uygulamanın çalışma yöntemi, işlem adımları, veri tabanı yapısı, veri tabanı kayıtları ve uygulamaya teknik destek sağlayan yazılım destek personelleri tarafından hazırlanmış olan, uygulamada karşılaşılan hataların ve çözümlerinin yer aldığı dokümanlar incelenmiştir. Bu incelemelerde uygulamada gerçekleştirilen fatura yönetimi işlemlerinin ilk aşamasının, uygulamaya yüklenen faturaların sınıflandırılması işlemi olduğu görülmüştür.

Uygulamanın kullanıldığı muhasebe departmanının yöneticileri ve uygulama kullanıcıları ile toplantılar düzenlenerek görüş, talep ve önerilerine başvurulmuştur. Departman yöneticileri tarafından fatura kayıt işlemlerinin daha hızlı yapılmasının departman verimliliği açısından oldukça önemli olduğu ve sınıflandırma aşamasının hızlandırılmasının iş süreçlerinin daha hızlı tamamlanması açısından önem arz ettiği yönünde görüşler alınmıştır. Makine öğrenmesi yönteminin uygulanabilirliğinin araştırılmasına karar verildikten sonra fatura verilerinin yer aldığı kayıtlar detaylı olarak incelenmiş ve sınıflandırma kullanıcıları ile toplantılar düzenlenmeye devam edilmiş, hangi verilerin veri setine dahil edilebileceği yönünde uzman görüşlerine sık sık başvurulmuştur. Veri setine dahil edilecek veriler belirlenirken, faturaların sisteme yüklendikleri ilk aşamadaki, hiçbir kullanıcı işlemi yapılmamış verilerin seçilmesi hususuna özellikle dikkat edilmiştir.

Makine öğrenmesi ile sınıflandırma çalışması yapılırken şekil 5'te yer alan makine öğrenmesi çalışması aşamaları izlenmiş, oluşturulan veri seti farklı makine öğrenme algoritmaları ile eğitilerek test edilmiş ve test sonuçları karşılaştırılmıştır. Karşılaşılan problemler araştırılmış ve iyileştirme çalışmaları yapılarak veri seti tekrar eğitilmiştir. Daha sonra eğitilen modelin kullanılabileceği bir kullanıcı arayüzü tasarlanmıştır. Projenin tamamında sistem geliştirme yaşam döngüsü adımları takip edilmiştir.

Şekil 11: Modelin Uygulamaya Dahil Edilebileceği Uygulama Yaşam Döngüsü Görseli



Kaynak: Yazar tarafından derlenmiştir.

Şekil 11'de, makine öğrenmesi modelinin, mevcut uygulama yaşam döngüsünün hangi aşamasında uygulamaya dahil edilebileceğinin gösterimi yer almaktadır. Mevcut sistemde kural tabanlı kodlama ile sınıflandırılamayan faturalar direkt olarak sınıflandırma kullanıcılarına iletilirken, şekil 11'de yer alan yeni yaşam döngüsünde bu aşamaya makine öğrenmesi modelinin dahil edilmesi önerilmektedir. Modelin hatalı sınıflandırdığı faturalar ise mevcut sistemde olduğu gibi; kayıt kullanıcıları tarafından, yeniden sınıflandırılmak üzere sınıflandırma kullanıcılarına iletilmeye devam edilecektir.

3.2. SİSTEM GELİŞTİRME YAŞAM DÖNGÜSÜ

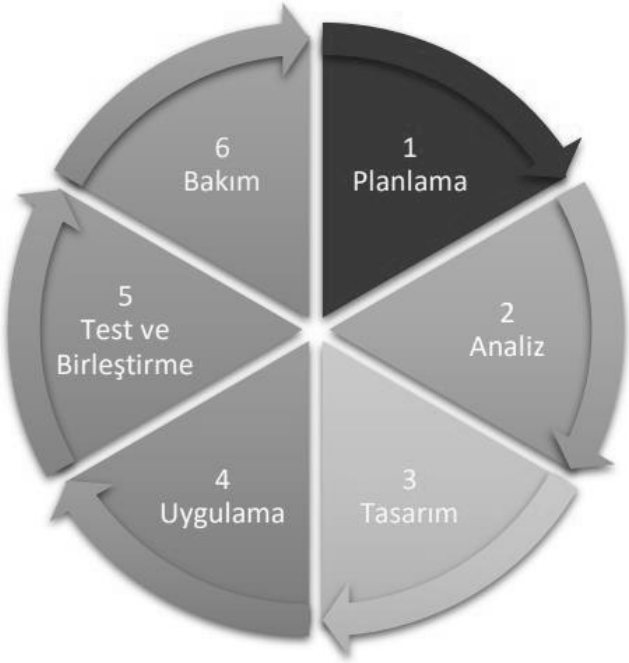
İş ve proje geliştirme süreçleri gerek yazılım projelerinde gerekse yazılımsal olmayan projelerde belli bir plan dahilinde gerçekleştirilmektedir. Yeni teknolojiler beraberinde yeni sektörlerin, yeni sektörler de yeni iş modellerinin ortaya çıkmasına neden olabilmektedir. Sektöre ve işin kapsamına bağlı olarak değişiklik göstermekle birlikte bu iş süreçleri zaman içerisinde bazı iş dallarında bir kurallar silsilesi haline dönüşebilmektedir.

Yazılım sektöründeki gelişmeler ve sektörün yaygınlaşması da beraberinde bazı iş geliştirme modellerini getirmiştir. Bu modeller belli noktalarda farklılık

gösterseler de temelde sistem geliştirme yaşam döngüsü (SGYD) adımlarına göre hareket etmektedirler. SGYD, bir sistem geliştirme çalışmasının bütün adımlarının detaylı bir şekilde planlanması, uygulanması ve kontrolünün sağlanması için bir yol haritası sağlar. Bu döngü, geliştirme sürecindeki olası hataları en aza indirgemek ve projelerden daha başarılı sonuçlar elde edebilmek amacıyla tasarlanmıştır (Tecim, 12.06.2023a).

SGYD genel yapısı itibariyle, planlama aşaması, analiz aşaması, tasarım aşaması, uygulama aşaması, test aşaması ve bakım ve destek aşaması olmak üzere 6 adımdan oluşmaktadır. Bu aşamalar farklı kaynaklarda farklı isimlerle yer alabilmektedir ve aşama sayısı da değişiklik gösterebilmektedir.

Şekil 12: Sistem Geliştirme Yaşam Döngüsü



Kaynak: Hussung, 2016:1.

Şekil 12'de SGYD adımlarının bir döngü şeklinde gösterimi yer almaktadır.

Planlama aşaması, proje gereksinimlerinin ve önceliklerinin incelendiği, sistemin geliştirilme amacı ve geliştirmenin hangi yöntemlerle gerçekleştirilebileceği

konularının belirlendiği aşamadır (Dennis vd., 2009). Bu aşamada problem tanımlanmış, departman yöneticileri ve uygulama kullanıcıları ile toplantılar düzenlenerek geliştirmenin amacı ve geliştirmenin hangi yöntemlerle gerçekleştirilebileceği, yöntemlerin avantaj ve dezavantajlarının neler olabileceği konuları hakkında fizibilite çalışmaları yapılmıştır. Bu aşamayla ilgili detaylı açıklamalar birinci ve ikinci bölümde yer almaktadır.

Analiz aşamasında, mevcuttaki sistemler ve iş süreçleri incelenir. Yeni sistemde hangi verilere ihtiyaç olacağı belirlenir. Yeni sistemin işlevselliği belirlenip tasarımı gerçekleştirilir. Sistemin teknik ihtiyaçları ve kullanıcı gereksinimleri de bu aşamada belirlenir (Tecim, 12.06.2023a).

Bu doğrultuda mevcut sistemin işleyişi, sınıflandırma öncesi ve sınıflandırma sonrası süreçler, veri tabanı kayıtları, otomatik olarak ve manuel olarak (kullanıcı tarafından) sisteme kaydolan verilerin neler olduğu, hangi verilerin hangi aşamada sisteme dahil olduğu detaylı olarak incelenmiştir. Çalışmanın temel amacı sınıflandırma aşamasının otomatik hale getirilmesi olduğu için uygulama aşamasında kullanılacak olan verilerin veri tabanına kaydı otomatik olarak gerçekleşen verilerden oluşması önem arz etmektedir. Bu aşamada da sınıflandırma kullanıcıları ile görüşmeler gerçekleştirilmiş ve uzman görüşünden faydalanılmıştır.

Tasarım aşaması: “Bu aşamada, analiz aşamasında belirlenen gereksinimlere dayanarak sistem tasarımı gerçekleştirilir. Bu aşama, kullanıcı arayüzü tasarımını, veri tabanı tasarımını ve sistem mimarisini içerir.” (Tecim, 12.06.2023a).

Bu aşamada uygulama aşamasının adımları belirlenmiş ve eğitilecek olan modelin kullanılabileceği arayüz uygulaması için giriş ve uygulama arayüzleri tasarlanmıştır.

Uygulama aşaması, sistem geliştirme yaşam döngüsünün önceki aşamalarında belirlenen ve tasarlanan sistemin gerçekleştirildiği aşamadır. Kod yazımı, uygulamanın geliştirilmesi, test işlemleri, kodların kontrol ve yönetim işlemleri bu aşamada yapılır (Şeker, 2014).

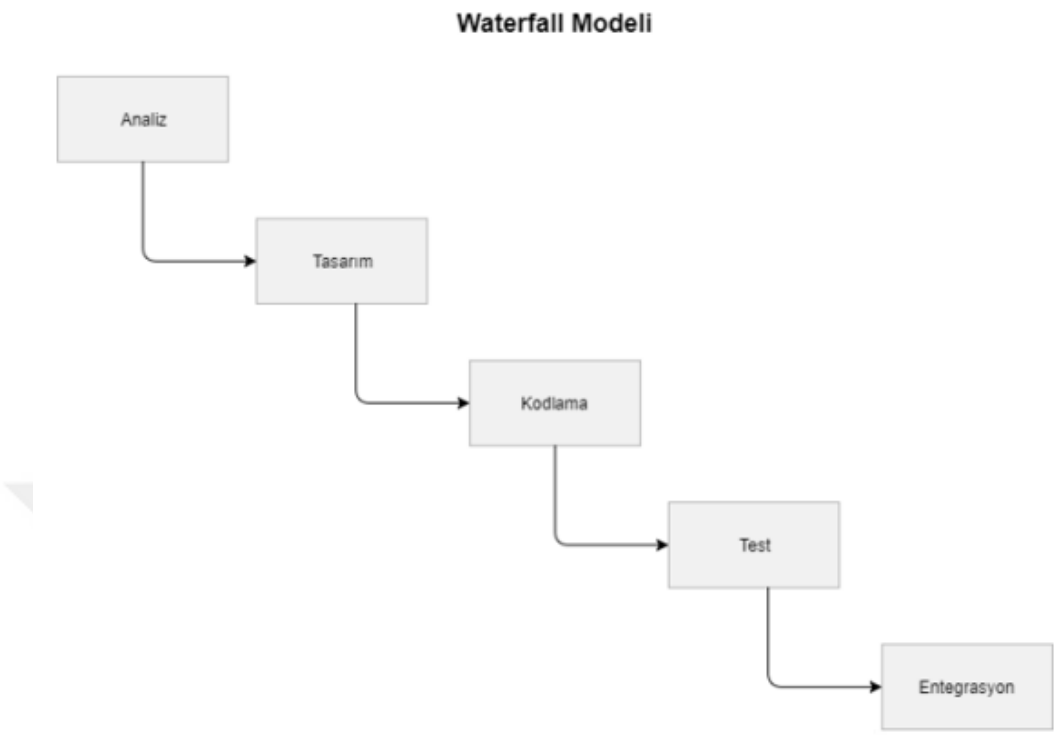
Uygulama aşamasında ilk olarak makine öğrenmesi ile sınıflandırma çalışması yapılmıştır. Bu çalışma yapılırken çalışmanın ikinci bölümünde yer alan makine öğrenmesi çalışmalarında izlenen adımlara göre hareket edilmiştir. Bu doğrultuda ilk olarak analiz aşamasında belirlenmiş olan veriler uygulama veri tabanından indirilmiş,

farklı tablolarda yer alan veriler veri bütünlüğü korunarak birleştirilmiştir. Sonraki adım olan veri ön işleme aşamasında sayısal olmayan veriler numerik değerlere dönüştürülerek eğitim aşaması için hazırlanmıştır. Daha sonra sırasıyla eğitim, test ve iyileştirme çalışmaları gerçekleştirilerek makine öğrenmesi modeli elde edilmiştir. Son olarak modelin kullanılabileceği bir kullanıcı arayüzü uygulaması geliştirilmiştir. Uygulama aşamasının bütün adımları dördüncü bölümde detaylı olarak anlatılmıştır.

Bakım ve destek aşaması, uygulama aşaması tamamlandıktan sonra başlayan aşamadır. Uygulama kullanılırken karşılaşılan hataların düzeltilmesi, sistemdeki yeni ihtiyaçların karşılanması, sistem bakım ve güncellemeleri ve kullanıcı desteği bu aşamanın parçalarıdır (Tecim, 12.06.2023a).

Bakım ve destek aşaması, uygulama aşamasında test işlemleri gerçekleştirilirken tespit edilemeyen, uygulama kullanılmaya başlandıktan sonra geliştiriciler veya kullanıcılar tarafından tespit edilen hataların düzeltilmesi, zaman içerisinde ortaya çıkan yeni ihtiyaçlara göre iyileştirme çalışmalarının yapılması, uygulama kullanıcılarına teknik destek sağlanması gibi işlemleri içermektedir. Şekil 1'de sistemin geliştirilme aşaması, canlıya geçişi (kullanıma sunulması), kurulum sonrası hataların düzeltilmesi, kullanım aşamasında tespit edilen ihtiyaçlar için ufak düzeltmelerin yapılması, zamanla teknolojide ve iş süreçlerinde meydana gelen yeni ihtiyaçlardan dolayı büyük değişikliklerin yapılması aşamaları ve aşamaların maliyet açısından karşılaştırılmaları detaylı olarak yer almaktadır. Şekil 13'te yer alan görselde ise yaygın kullanılan SGYD modellerinden olan Waterfall modeli aşamaları yer almaktadır.

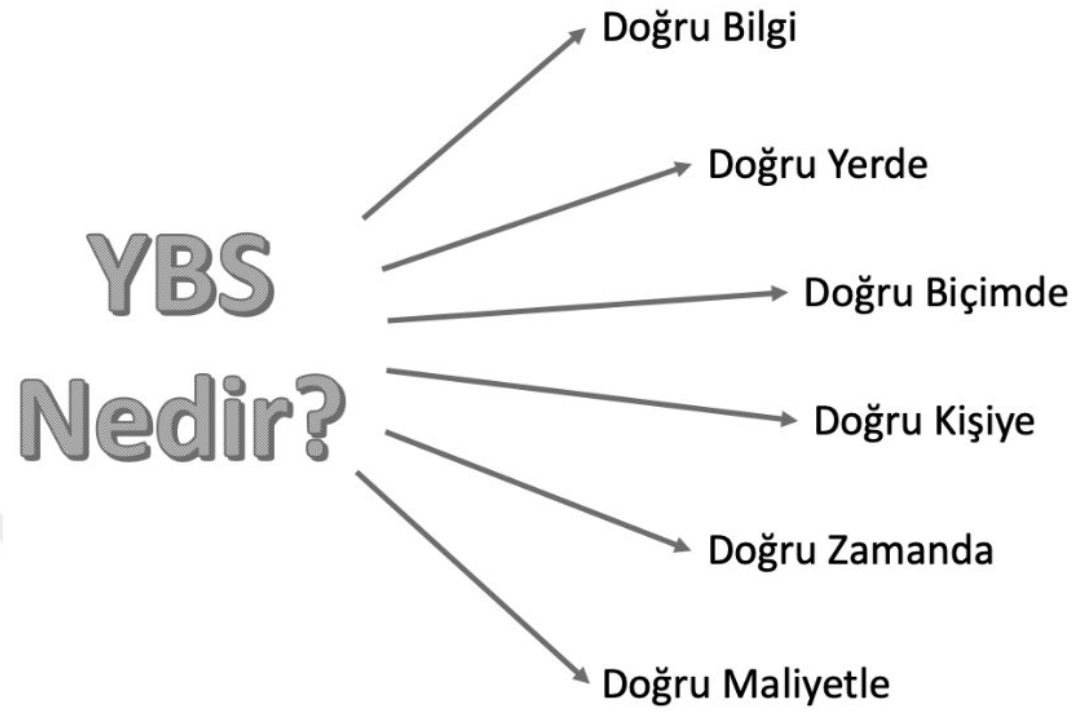
Şekil 13: Waterfall Modeli



Kaynak: Kılınç, 2018: 13.

Çalışma, dördüncü bölümde detaylı olarak açıklanan uygulama kısmı ve eğitilen modelin kullanılabileceği arayüz uygulaması itibariyle bütün SGYD aşamalarını içermektedir. Fatura yönetimi uygulamasına entegre edilmesi durumunda bu çalışma, fatura yönetimi uygulaması için gerçekleştirilmiş bir bakım ve destek çalışmasıdır diyebiliriz.

Şekil 14: YBS nedir?



Kaynak: Tecim, 12.06.2023

YBS, "Grafikte görüldüğü üzere, doğru bilgiyi, doğru yerde, doğru biçimde, doğru kişiye, doğru zamanda ve uygun maliyetle ulaşmasını sağlayan teknoloji tabanlı bilgi sistemidir." (Tecim, 12.06.2023)

DÖRDÜNCÜ BÖLÜM
UYGULAMA PROJE YÖNETİMİ VE GELİŞTİRİLEN ARAYÜZÜN BÖLÜMLERİ

4.1. VERİLERİN TOPLANMASI VE ANLAŞILMASI AŞAMASI

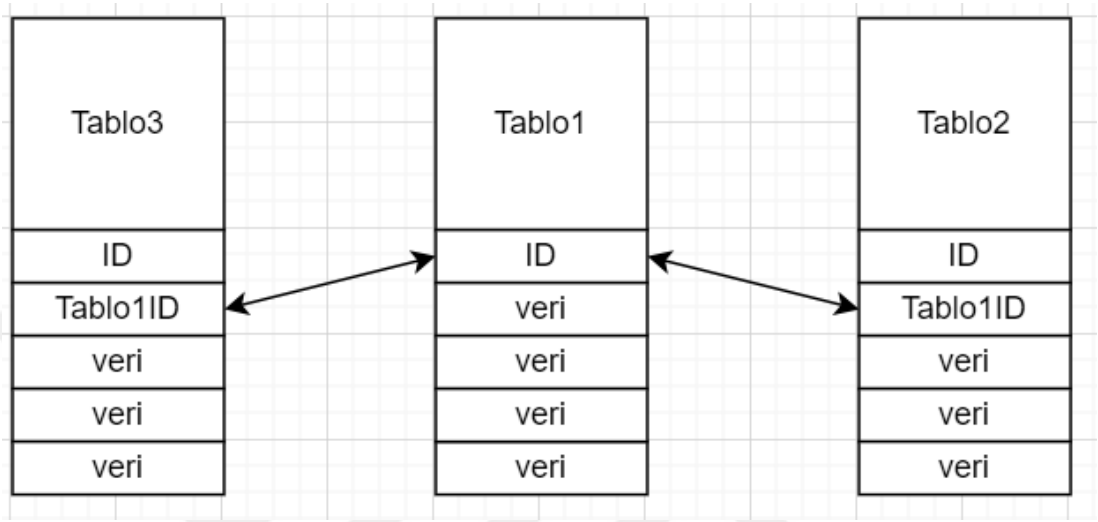
Verilerin toplanması aşamasında geçmiş faturaların uygulamanın veri tabanında kayıtlı olan işlem yapılmamış ham haldeki verileri araştırılmıştır. Bu aşamada ilk olarak veri tabanında yer alan tablolar incelenmiş, fatura verilerinin bulunduğu tablolar tespit edilmiştir. Amacımız faturaların sınıflandırılması işleminin otomatik hale getirilmesi olduğu için uygulamada kullanılacak verilerin geçmiş faturaların otomatik olarak veri tabanına kaydedilen verilerinden ve işlemleri tamamlanmış olan geçmiş faturaların sınıf bilgilerinden oluşması gerekmektedir. Bu veriler tespit edildikten sonra sınıflandırma kullanıcıları ile toplantı düzenlenerek faturaların sınıflarının belirlenmesinde etkili olabilecek verilerin hangi veriler olabileceği sorulmuş ve kullanıcıların belirttiği veriler de göz önünde bulundurularak veri seti oluşturulmuştur. Bu doğrultuda veri tabanı incelendiğinde veri setine dahil edilecek verilerin 3 farklı tabloda yer aldığı anlaşılmıştır.

Uygulamada ilişkisel veri tabanı olarak MySQL veri tabanı kullanılmaktadır ve faturaların verileri de bu veri tabanında tutulmaktadır. İlişkisel veri tabanları verilerin birden fazla tabloda tutulmasına olanak sağlayan ve benzersiz kimlik numaraları (ID) ile tablolar arası bağlantıların ve veri bütünlüğünün korunduğu veri tabanlarıdır.

Uygulamanın veri tabanı incelendiğinde veri setine dahil edilecek verilerin 3 farklı tabloda yer aldığı anlaşılmıştır. Çalışmanın devamında bu tablolardan TABLO1, TABLO2 ve TABLO3 şeklinde bahsedilmiştir. Bu tablolardan birisinde bir fatura için birden fazla kayıt tutulabildiği ve yapılan her yeni işlemde ID numarasına alt tire (‘_’) karakteri ile yeni numaralar eklenerek yeni kayıt eklendiği tespit edilmiştir. Bu tablodaki verilerin son hali ise ID numarası alt tire (‘_’) karakterini içermeyen kayıtlardan oluşmaktadır. İnceleme işlemlerinden sonra indirme işlemi esnasında yeni fatura verilerinin veri tabanına akmaya devam etmesinden kaynaklı sorun

yaşanmaması için uygulama durdurulmuş ve veri tabanı bütün olarak lokal bilgisayara indirilmiştir.

Şekil 15: Uygulama Veri Tabanında Kullanılan Yapının Temsili Gösterimi

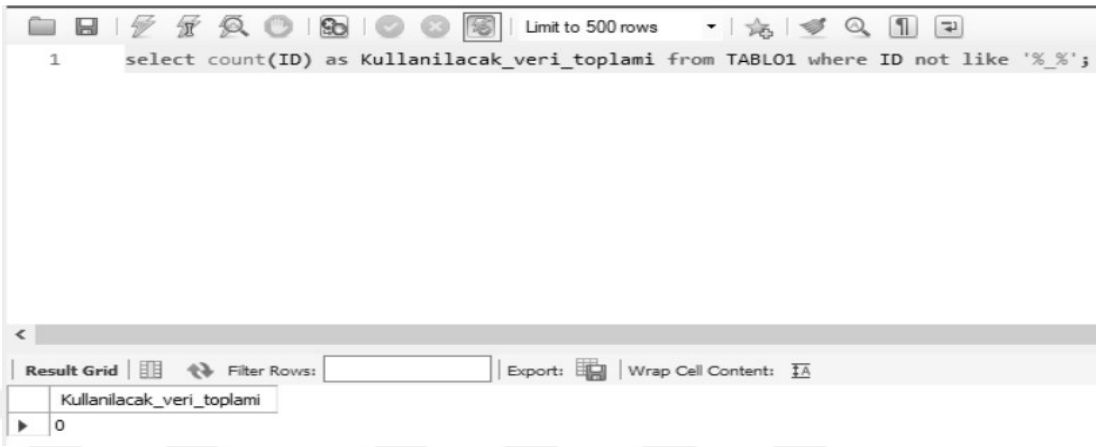


Kaynak: Yazar tarafından derlenmiştir.

4.2. VERİ ÖN İŞLEME AŞAMASI

Bu aşamada ilk olarak lokal bilgisayara indirilen veriler bu bilgisayardaki MySQL uygulamasına yüklenmiştir. Daha sonra veri tabanı üzerinde veri temizleme çalışmasına başlanmıştır. Veri temizleme aşamasında ilk olarak bir öndeki aşamada değinilmiş olan ID verilerinden alt tire (‘_’) karakteri içeren fazla kayıtlar temizlenmiştir. MySQL veri tabanında veri çekme işlemleri Sql sorgulama dili ile gerçekleştirilmektedir. Sql dili ile sorgu yazılırken alt tire (‘_’) karakterinin direkt olarak yazılması durumunda sorgu sonucunun hatalı döndüğü anlaşılmıştır. Araştırma yapıldığında doğru sonuçlar elde edilebilmesi için bu karakterin önüne ters eğik çizgi (‘\’) karakteri eklenerek ‘_’ şeklinde yazılması gerektiği tespit edilmiştir. Sql dilinde kullanılan ‘like’ komutu ise veri çekilirken içerisinde belli karakterler bulunduran verilerin çekilmesine olanak sağlamaktadır. Bu komut kullanılarak çekilen alt tire (‘_’) içeren veriler Sql dilinde silme işlemi yapılmasını sağlayan ‘delete’ komutu ile temizlenmiştir.

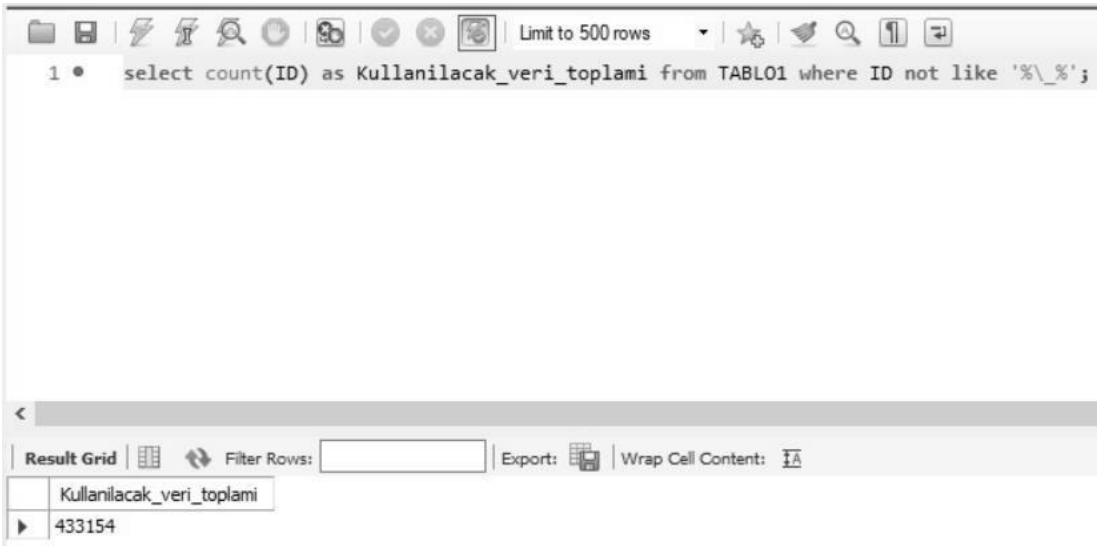
Şekil 16: ID verileri alt tire (‘_’) karakteri içermeyen verilerin toplamının ters eğik çizgi (‘\’) kullanılmadan Sql dili ile sorgulanması



Kaynak: Yazar tarafından derlenmiştir.

Şekil 16’da görüldüğü üzere alt tire karakteri, ters eğik çizgi karakteri eklenmeden kullanıldığında 0 sonucu alınmaktadır.

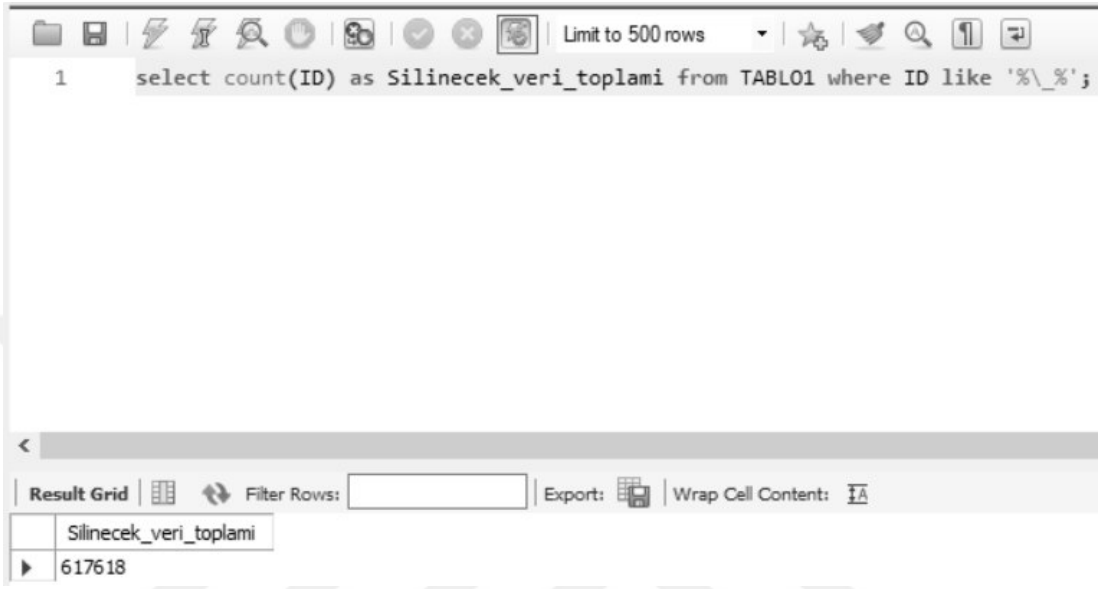
Şekil 17: ID numaraları alt tire (‘_’) karakteri içermeyen verilerin toplamının ters eğik çizgi (‘\’) kullanılarak Sql dili ile sorgulanması



Kaynak: Yazar tarafından derlenmiştir.

Tablo1 tablosundan veri setine dahil edilebilecek fatura sayısının doğru sonucu Sql sorgusuna ters eğik çizgi eklenerek sorgu çekildiğinde elde edilebilmiştir.

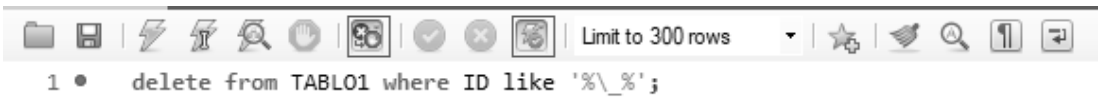
Şekil 18: Tablo1 Tablosunda ID Numaraları Alt Tire (‘_’) Karakteri İçeren Verilerin Toplamının Sorgulanması



Kaynak: Yazar tarafından derlenmiştir.

Şekil 18’de yer alan sorgu ile Tablo1’de yer alan ve ID numaraları alt tire (‘_’) karakteri içeren yani Tablo1 den silinmesi gereken toplam kayıt sayısı elde edilmiştir.

Şekil 19: Tablo1’de Yer Alan ID Numaraları Alt Tire İçeren Kayıtların Silinmesini Sağlayan Sql Sorgusu

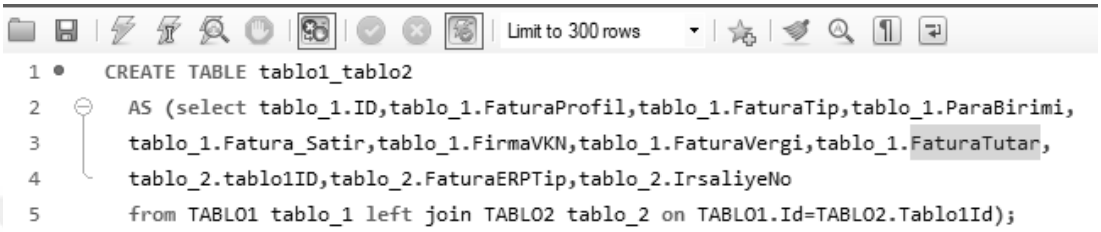


Kaynak: Yazar tarafından derlenmiştir.

Şekil 19’da yer alan sorgu çalıştırılarak Tablo1’de yer alan ve ID numaraları alt tire (‘_’) karakteri içeren kayıtlar Tablo1’ den silinmiştir. Bu aşamadaki silme işlemi yapılırken ters eğik çizgi karakteri eklenmeden sadece alt tire karakteri ile Sql sorgusunun çalıştırılması durumunda tablodaki verilerin tamamı silinecektir. Bu nedenle Sql sorgusu yazılırken alt tire karakterine ters eğik çizgi karakterinin eklenmesi özellikle bu aşamada yüksek önem taşımaktadır.

Bu işlemden sonra Tablo1 ve Tablo2 tablolarında bulunan fatura verileri MySQL veri tabanında oluşturulan yeni bir tabloda birleştirilerek tek tablo haline getirilmiştir. Birleştirme işleminin yapıldığı Sql sorgusunda ID numaraları kullanılarak veri bütünlüklerinin korunması sağlanmıştır.

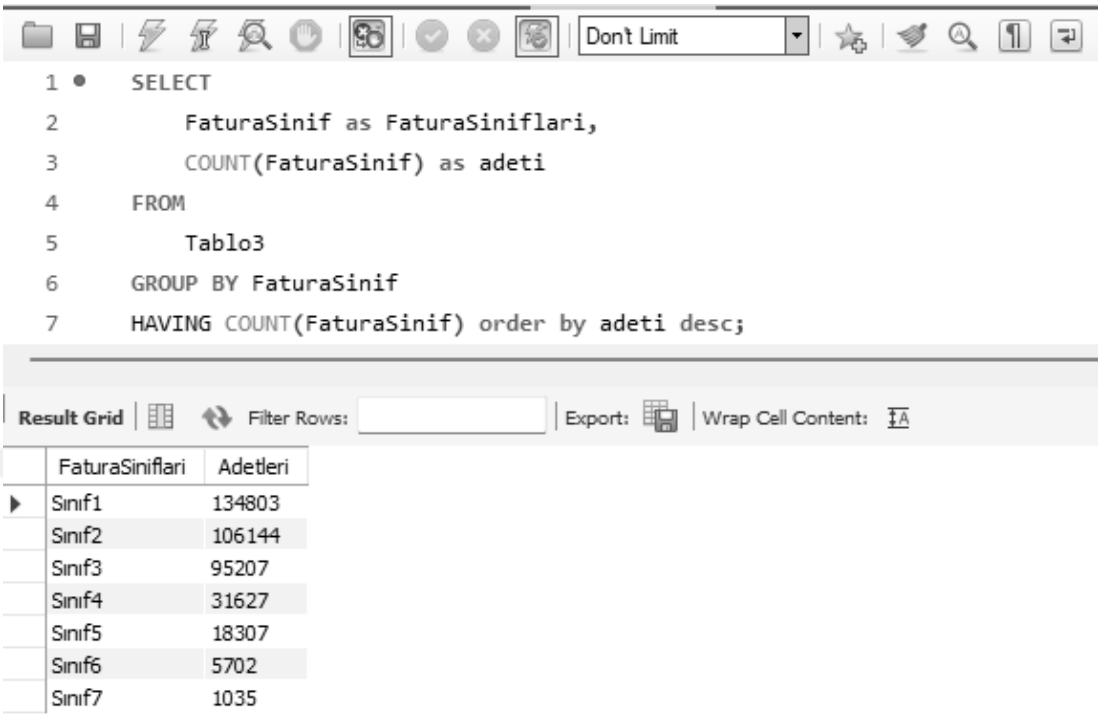
Şekil 20: MySQL Veri Tabanında Tablo1 Ve Tablo2 Tablolarının Birleştirildiği Sql Sorgusu



Kaynak: Yazar tarafından derlenmiştir.

Şekil 20'de yer alan sorgu ile Tablo1 ve Tablo2 tabloları tablo1_tablo2 isimli yeni bir tabloda birleştirildikten sonra tablo1_tablo2 tablosunda yer alan veriler Excel dosyası olarak veri tabanından dışarı aktarılmıştır. Daha sonra Tablo3 tablosunda yer alan fatura sınıfı bilgisi ve tablo1ID bilgileri de Excel dosyası olarak dışarı aktarılmış ve dışarı aktarılan Excel dosyaları Excel'in 'düşeyara' fonksiyonu kullanılarak birleştirilmiş ve tek tablo haline getirilmiştir.

Şekil 21: MySQL Veri Tabanında Tablo3 Tablosunda Yer Alan Farklı Fatura Sınıflarının Ve Adetlerinin Çekildiği Sql Sorgusu.



```
SELECT
    FaturaSinif as FaturaSiniflari,
    COUNT(FaturaSinif) as adeti
FROM
    Tablo3
GROUP BY FaturaSinif
HAVING COUNT(FaturaSinif) order by adeti desc;
```

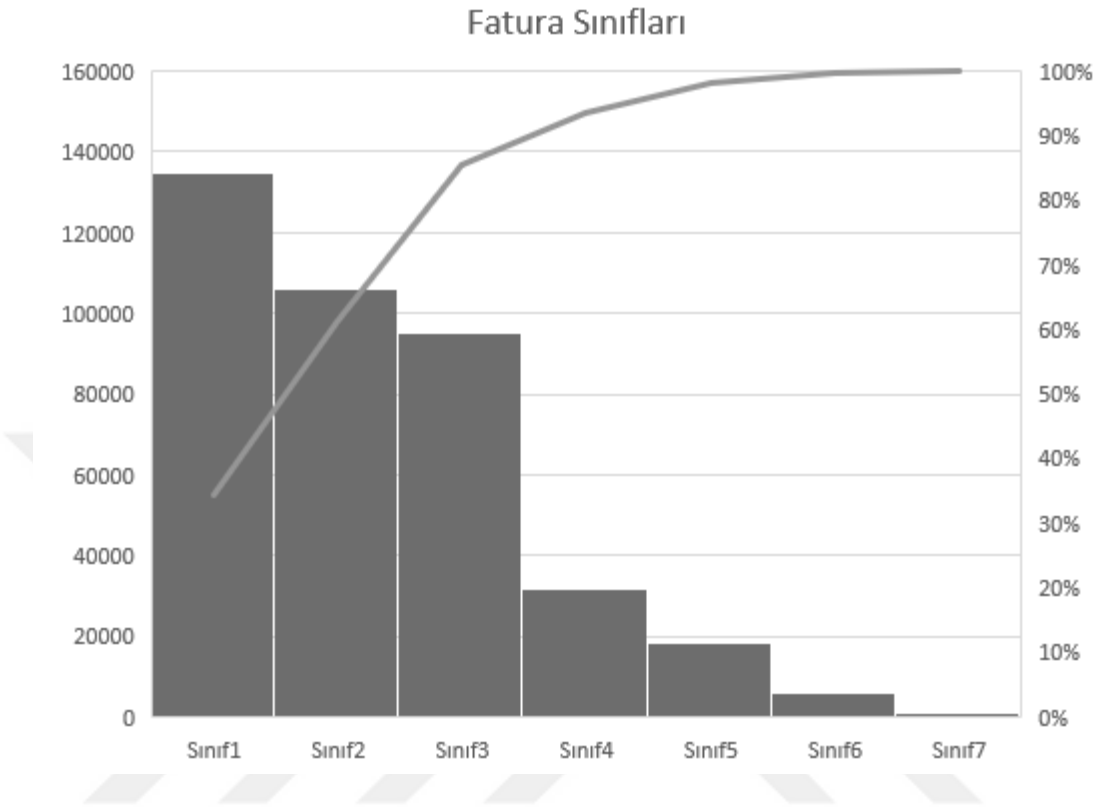
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

FaturaSiniflari	Adetleri
Sınıf1	134803
Sınıf2	106144
Sınıf3	95207
Sınıf4	31627
Sınıf5	18307
Sınıf6	5702
Sınıf7	1035

Kaynak: Yazar tarafından derlenmiştir.

Şekil 21'de yer alan sorgu ile sistemde faturaların 7 farklı sınıfa ayrıldığı ve bu sınıflarda yer alan faturaların hangi sınıftan kaç adet yer aldığı bilgisi elde edilmiştir. Bu sınıflar sırasıyla: Sınıf1: ihracat, Sınıf2: ithalat, Sınıf3: ürün, Sınıf4: hizmet, Sınıf5: konaklama, Sınıf6: ulaşım ve Sınıf7: reklam sınıflarıdır. Çalışmanın devamında bu fatura sınıflarından şekil 21'de FaturaSiniflari başlığı altında yer alan isimlerle bahsedilmiştir.

Şekil 22: MySQL Veri Tabanından Çekilen Tablo3 Tablosunda Yer Alan Farklı Fatura Sınıflarının, Adetlerinin Ve Oranlarının Excel Uygulamasında Grafik Haline Getirilmiş Hali



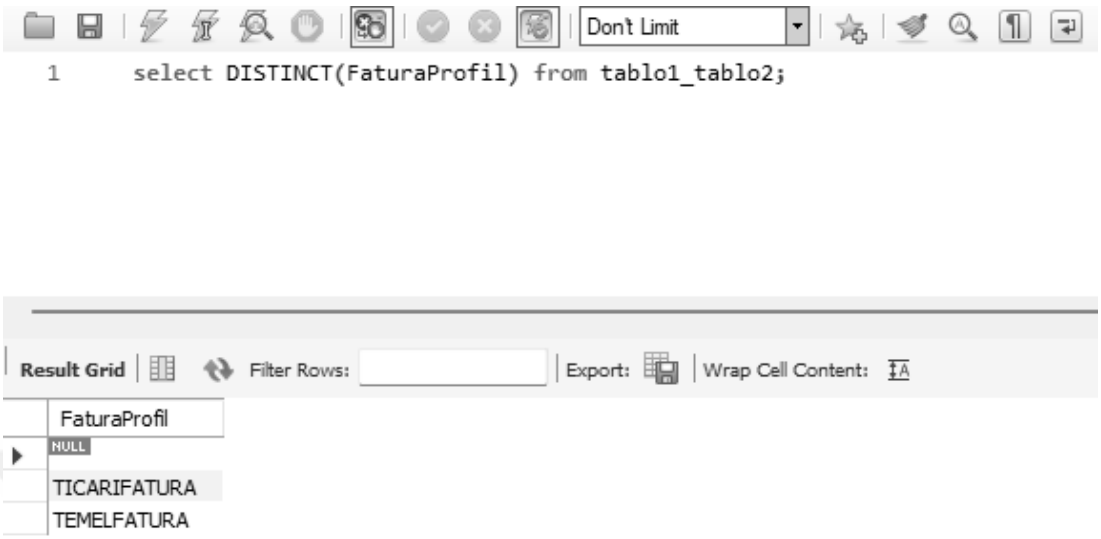
Kaynak: Yazar tarafından derlenmiştir.

Sınıf7 sınıfına dahil olan faturaların miktarı hem sayı olarak hem de oran olarak oldukça düşük olduğu için bu sınıfa ait veriler eğitim kümesinden çıkarılmıştır.

4.3. SAYISAL OLMAYAN VERİLERİN NUMARALANDIRILMASI VE VERİLERİN DÜZENLENMESİ

Verilerin makine öğrenmesi ile eğitilebilmesi için sayısal olmayan verilerin numerik hale getirilmesi gerekmektedir. Bu nedenle fatura verilerinin kayıtlarda kaç farklı tipte yer aldığı ve bu veri tiplerinin neler olduğu MySQL uygulamasında oluşturulan sorgularla incelenmiş ve Excel uygulaması üzerinde düzenleme işlemlerine devam edilmiştir.

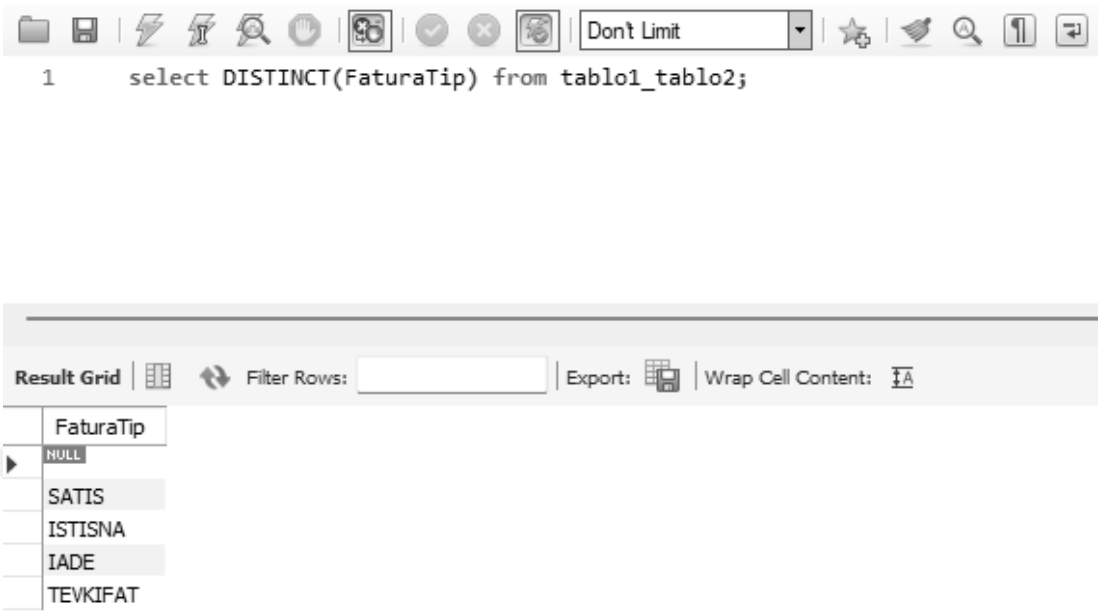
Şekil 23: FaturaProfil Başlığı Altında Yer Alan Farklı Değerlerin MySQL Uygulamasında Sorgulanması



Kaynak: Yazar tarafından derlenmiştir.

Şekil 23'te yer alan sorgu çalıştırıldığında FaturaProfil başlığı altında yer alan verilerin bir kısmının 'TEMELFATURA', bir kısmının 'TICARIFATURA' şeklinde kayıtlı olduğu geriye kalan kayıtlarda ise FaturaProfil bilgisinin yer almadığı, bu başlık altında 'TEMELFATURA' ve 'TICARIFATURA' olmak üzere 2 farklı verinin olduğu ve geriye kalan faturalarda bu alanın boş olduğu tespit edilmiştir. Excel tablosunda düşeyara fonksiyonu kullanılarak FaturaProfil başlığı altında 'TEMELFATURA' olarak yer alan değerler 1 sayısı ile, 'TICARIFATURA' şeklinde kayıtlı olan değerler 2 sayısı ile değiştirilmiştir. Boş olan satırlar ise 0 sayısı ile numaralandırılmıştır.

Şekil 24: FaturaTip Başlığı Altında Yer Alan Farklı Değerlerin Mysql Uygulamasında Sorgulanması



Kaynak: Yazar tarafından derlenmiştir.

Şekil 24'te bulunan sorgu çalıştırıldığında FaturaTip başlığı altında yer alan verilerin 'SATIS', 'ISTISNA', 'IADE' ve 'TEVKIFAT' şeklinde 4 farklı değerden oluştuğu ve bu değerlerin dışında boş sütunların yer aldığı tespit edilmiştir. Excel tablosunda düşeyara fonksiyonu kullanılarak FaturaTip başlığı altında 'IADE' olarak yer alan değerler 1 sayısı ile, 'ISTISNA' şeklinde kayıtlı olan değerler 2 sayısı ile, 'SATIS' olarak yer alan değerler 3 sayısı ile ve 'TEVKIFAT' olarak yer alan değerler 1 sayısı ile değiştirilmiştir. Boş olan satırlar ise 0 sayısı ile numaralandırılmıştır.

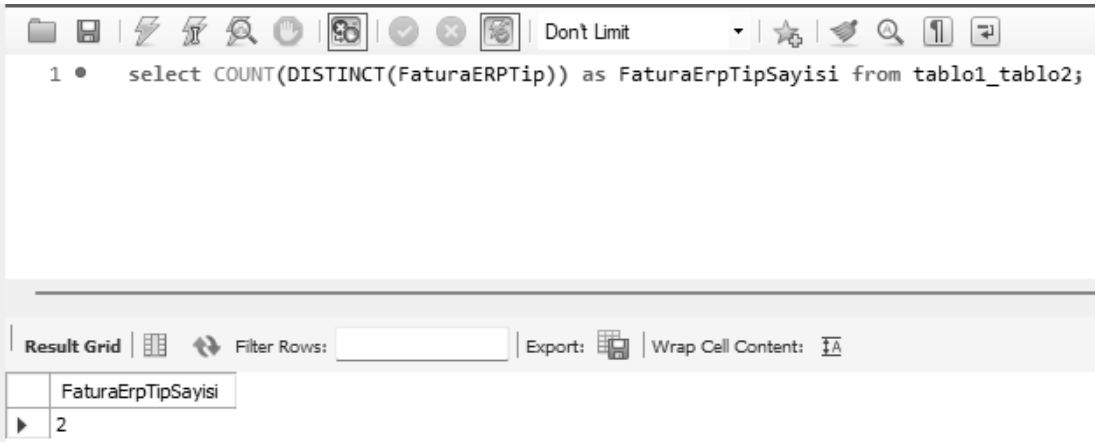
Şekil 25: ParaBirimi Başlığı Altında Yer Alan Farklı Değerlerin Mysql Uygulamasında Sorgulanması



Kaynak: Yazar tarafından derlenmiştir.

Şekil 25'te yer alan sorgu çalıştırıldığında ParaBirimi başlığı altında yer alan verilerin 'TRL', 'TRY', 'YTL', 'USD', 'EUR' ve 'GBP' şeklinde 6 farklı değerden oluştuğu tespit edilmiştir. İnceleme yapıldığında 'TRL', 'TRY' ve 'YTL' verilerinin Türk lirasını ifade ettikleri tespit edilmiştir. Excel tablosunda düşeyara fonksiyonu kullanılarak ParaBirimi başlığı altında 'TRL', 'TRY' ve 'YTL' olarak yer alan değerler 0 sayısı ile, 'EUR' şeklinde kayıtlı olan değerler 1 sayısı ile, 'USD' olarak yer alan değerler 2 sayısı ile ve 'GBP' olarak yer alan değerler 3 sayısı ile değiştirilmiştir.

Şekil 26: FaturaERPTip başlığı altında yer alan girdilerin kaç farklı değer içerdiğinin MySQL uygulamasında sorgulanması.



Kaynak: Yazar tarafından derlenmiştir.

Şekil 26'daki sorgu çalıştırıldığında FaturaERPTip başlığı altında yer alan verilerin 2 farklı değerden oluştuğu tespit edilmiştir. Excel tablosunda düşeyara fonksiyonu kullanılarak FaturaERPTip başlığı altındaki değerler 0 sayısı ve 1 sayısı ile değiştirilmiştir.

Bu aşamada son olarak Excel tablosunda yer alan ve sayısal olmayan değerler içeren IrsaliyeNo başlığı altındaki veriler incelenmiştir. İrsaliye numarası her faturada yer almayan ve tekrarlanmayan bir veri tipi olduğu için bu başlık altındaki sütunlardan boş olanlar 0 ile, İrsaliye numarası bulunanlar 1 sayısı ile değiştirilerek veri seti kaydedilmiş ve eğitim aşamasına geçilmiştir.

4.4. EĞİTİM AŞAMASI

Verilerin eğitilmesi işlemi Python programlama dili ile gerçekleştirilmiştir. Bu işlemde Python'da yer alan numpy, pandas ve sklearn kütüphanelerinden faydalanılmış ve görselleştirme ve grafik oluşturma işlemleri için matplotlib kütüphanesi kullanılmıştır.

Eğitim işlemleri makine öğrenmesi algoritmalarından Lojistik Regresyon, Naive Bayes, Karar Ağaçları, Rassal Orman, Destek Vektör Makineleri ve K- En Yakın Komşu Algoritması (K-NN) algoritmaları ile ayrı ayrı gerçekleştirilmiş ve başarı sonuçları kontrol edilerek karşılaştırılmıştır. Çalışmada veri seti %90 eğitim

kümesi ve %10 test kümesi olacak şekilde belirlenmiş, verilerin %90'lık kısmı ile eğitim çalışmaları gerçekleştirilirken eğitim kümesinde kullanılmayan %10'luk kısmı ile de oluşturulan modellerin başarılarının test edilmesi amaçlanmıştır.

Şekil 27: Kütüphanelerin, Makine Öğrenmesi Algoritmalarının Ve Excel Dosyası Olarak Hazırlanmış Olan Veri Setinin Eğitim Koduna Dahil Edildiği Python Kod Bloğu Görseli

```
from matplotlib import pyplot as plt
import numpy as np
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split


df = pd.read_excel("FaturaVerileri.xlsx")
```

Kaynak: Yazar tarafından derlenmiştir.

Şekil 28: Lojistik Regresyon Algoritmasının Başarı Oranı

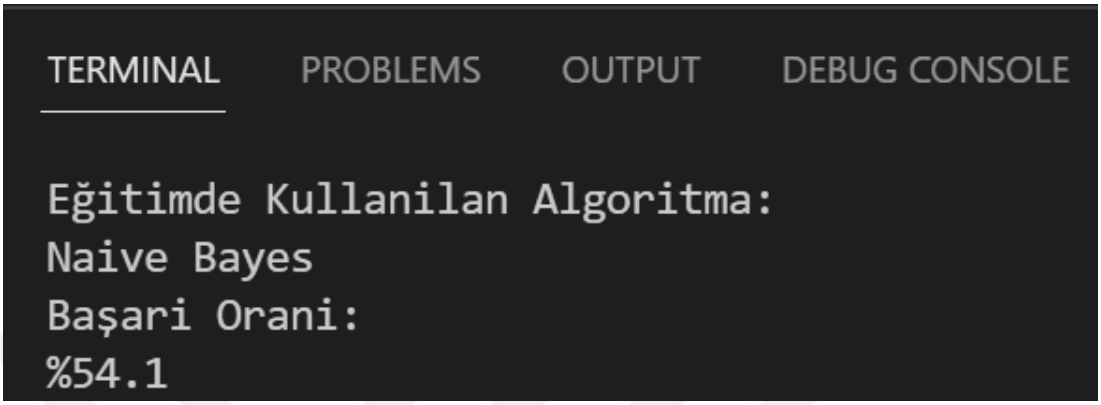
```
TERMINAL   PROBLEMS   OUTPUT   DEBUG CONSOLE

Eğitimde Kullanilan Algoritma:
Lojistik Regresyon
Başari Orani:
%72
```

Kaynak: Yazar tarafından derlenmiştir.

Eğitimde Lojistik Regresyon algoritması kullanıldığında test verilerinin %72 oranında doğru sınıflandırıldığı görülmüştür.

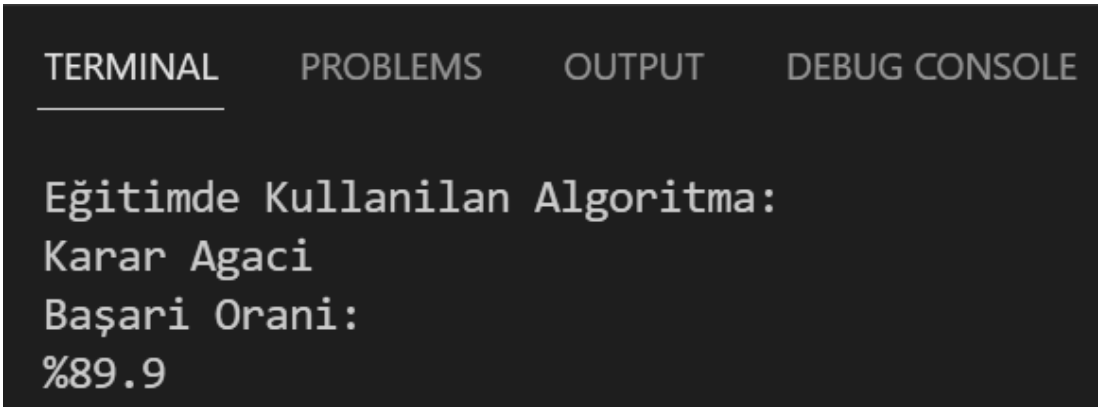
Şekil 29: Naive Bayes Algoritmasının Başarı Oranı



Kaynak: Yazar tarafından derlenmiştir.

Eğitimde Naive Bayes algoritması kullanıldığında test verilerinin %54.1 oranında doğru sınıflandırıldığı görülmüştür.

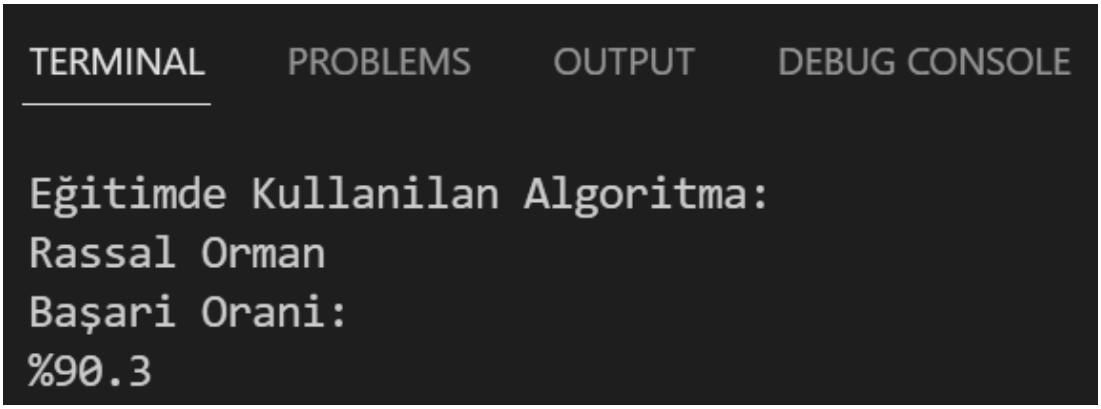
Şekil 30: Karar Ağacı algoritmasının başarı oranı



Kaynak: Yazar tarafından derlenmiştir.

Eğitimde Karar Ağacı algoritması kullanıldığında test verilerinin %89.9 oranında doğru sınıflandırıldığı görülmüştür.

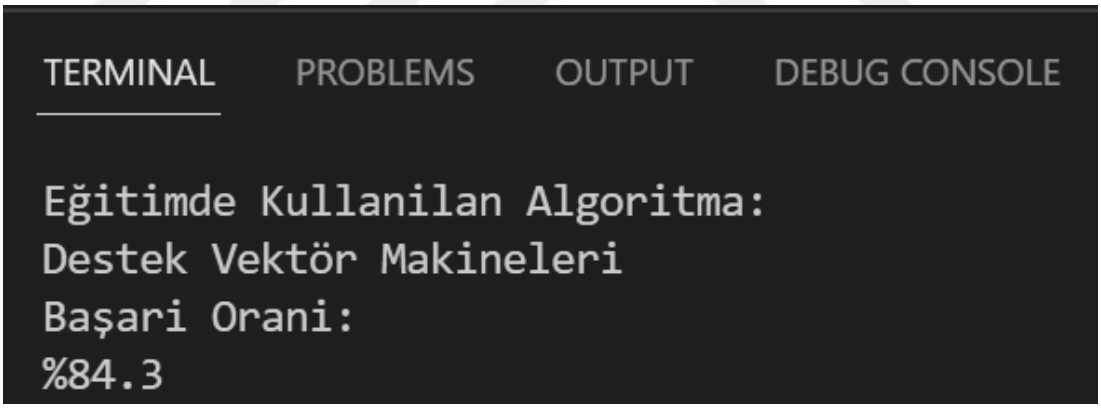
Şekil 31: Rassal Orman Algoritmasının Başarı Oranı



Kaynak: Yazar tarafından derlenmiştir.

Eğitimde Rassal Orman algoritması kullanıldığında test verilerinin %90.3 oranında doğru sınıflandırıldığı görülmüştür.

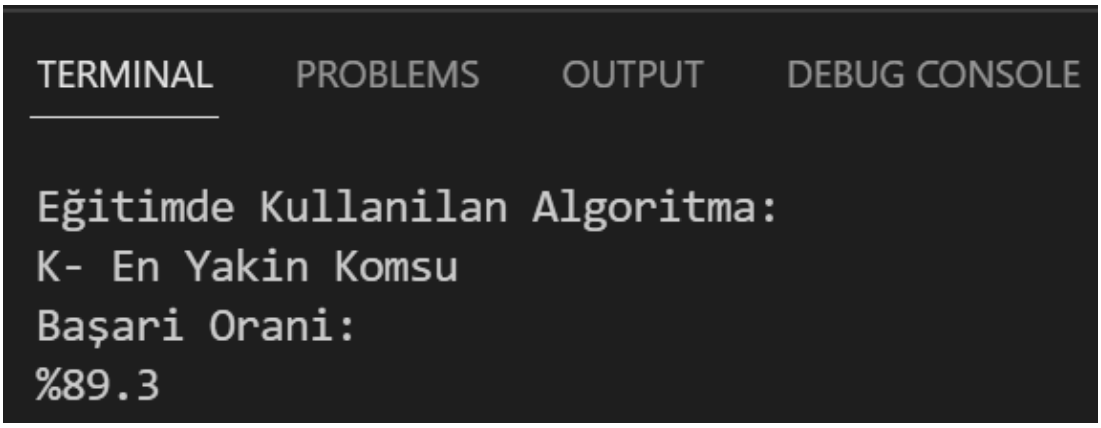
Şekil 32: Destek Vektör Makinesi algoritmasının başarı oranı



Kaynak: Yazar tarafından derlenmiştir.

Eğitimde Destek Vektör Makinesi algoritması kullanıldığında test verilerinin %84.3 oranında doğru sınıflandırıldığı görülmüştür.

Şekil 33: Destek K-En Yakın Komşu Algoritmasının Başarı Oranı



Kaynak: Yazar tarafından derlenmiştir.

Eğitimde K-En Yakın Komşu algoritması kullanıldığında test verilerinin %89.3 oranında doğru sınıflandırıldığı görülmüştür.

4.5. TEST AŞAMASI

Test aşamasında, eğitim aşamasında oluşturulmuş olan modeller, farklı sınıflara ait veriler kullanılarak test edilmiştir. Eğitim aşamasında model, verilerin %10'unu test verisi olarak kullanacak şekilde oluşturulmuş ve test verilerini hangi oranda doğru sınıflandırdığının sonuçları uygulama konsoluna yazdırılmıştı. Bu aşamada test yapılan test işlemi için yeni bir Excel dosyası hazırlanmış, bu Excel dosyasına modelde kullanılan veri setinden farklı sınıflara ait 100 adet fatura verisi aktarılmış ve 'Fatura_Sinif' sütunu boş bırakılmıştır. Oluşturulan modellere eklenen Python kodları ile modelin bu verilere ait sınıf tahminlerini Excel dosyasında boş bırakılmış olan, 'Fatura_Sinif' başlığı altında yer alan hücrelere yazdırılması sağlanmıştır.

Şekil 34: Test işlemi için kullanılan Excel dosyasına modelin sınıf tahminlerinin yazdırılmasını sağlayan Python kod bloğu görseli

```
df_test = pd.read_excel('test_verileri.xlsx')


tahminler = model.predict(df_test.iloc[:, :-1])

df_test['Fatura_Sinif'] = tahminler

print(df_test)
df_test.to_excel('tahmin_sonuclari.xlsx', index=False)
```

Kaynak: Yazar tarafından derlenmiştir.

Oluşturulan modeller üzerinde test işlemi yapılarak tahmin sonuçlarının yazdırıldığı Excel dosyası kontrol edildiğinde modellerin yoğun olarak Sınıf1 ve Sınıf2 tahmin sonucu verdiği ve nadiren de Sınıf3 tahmininde bulunduğu gözlemlenmiş, diğer sınıflara ait tahmin sonucu elde edilemediği tespit edilmiştir. Araştırma yapıldığında bu durumun ‘Aşırı Öğrenme (Overfitting)’ sorunundan kaynaklandığı anlaşılmıştır.

Şekil 35: Modellerin tahmin sonuçlarının yazdırıldığı Excel sütunları görseli

Fatura_Sinif	Doğru Sınıf	Tahmin Sonucu
Sınıf1	Sınıf1	DOĞRU
Sınıf1	Sınıf3	YANLIŞ
Sınıf1	Sınıf4	YANLIŞ
Sınıf1	Sınıf5	YANLIŞ
Sınıf1	Sınıf1	DOĞRU
Sınıf2	Sınıf2	DOĞRU
Sınıf1	Sınıf3	YANLIŞ
Sınıf1	Sınıf3	YANLIŞ
Sınıf1	Sınıf4	YANLIŞ
Sınıf3	Sınıf3	DOĞRU
Sınıf1	Sınıf1	DOĞRU
Sınıf1	Sınıf5	YANLIŞ
Sınıf1	Sınıf5	YANLIŞ
Sınıf1	Sınıf1	DOĞRU
Sınıf1	Sınıf1	DOĞRU
Sınıf1	Sınıf3	YANLIŞ
Sınıf1	Sınıf4	YANLIŞ
Sınıf1	Sınıf5	YANLIŞ
Sınıf1	Sınıf1	DOĞRU
Sınıf2	Sınıf2	DOĞRU
Sınıf2	Sınıf6	YANLIŞ
Sınıf1	Sınıf4	YANLIŞ
Sınıf1	Sınıf6	YANLIŞ
Sınıf1	Sınıf3	YANLIŞ
Sınıf1	Sınıf2	YANLIŞ
Sınıf1	Sınıf1	DOĞRU

Kaynak: Yazar tarafından derlenmiştir.

Aşırı Öğrenme (Overfitting): modelin eğitim kümesinde kullanılan veri kümesini ezberlemesi ve yeni veriler verilerek tahmin sonucu vermesi istenildiğinde modelin doğru sonuçlar verememesi sorunudur.

Aşırı öğrenme sorunu, öz nitelik sayısını azaltmak, daha fazla veri eklemek ve verilerde düzenlemeler yapmak gibi farklı yöntemler uygulanarak aşılabilmektedir.

4.6. İYİLEŞTİRME AŞAMASI

Bu aşamada ilk olarak kodda parametre düzenlemesi işlemi denenmiştir. Makine öğrenmesi algoritmalarında yer alan parametrelerde değişiklikler yapılarak aşırı öğrenme sorununun giderilip giderilmediği incelendiğinde sorunun devam ettiği tespit edilmiştir. Daha sonra veri seti tekrar incelenmiş ve veri setinde düzenleme yapılmasına karar verilmiştir.

İlk olarak en az sayıda verisi bulunan sınıf olan Sınıf6 sınıfına ait veriler veri setinden kaldırılmıştır. Bu işlemden sonra sorunun yine devam ettiği tespit edilmiştir.

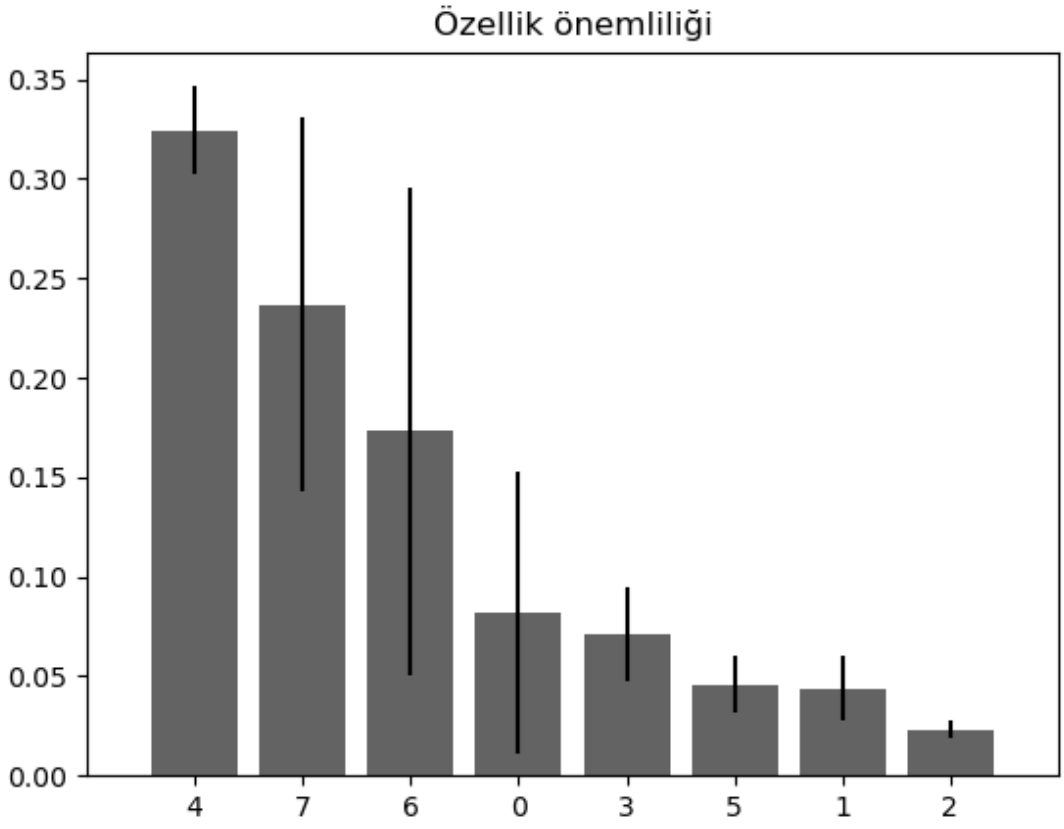
Veri seti tekrar incelendiğinde veri setinde yer alan ‘Fatura Toplam Tutarı’ ve ‘Fatura Toplam Vergi Tutarı’ verilerinin kullanılmasının, farklı yıllara ait faturalardan oluşan bir veri seti için, enflasyon faktörü göz önünde bulundurulduğunda, anlamlı ve sağlıklı olmadığı tespit edilmiştir. Enflasyon, fiyatlar genel düzeyindeki artış anlamına gelmektedir ve bu durum, aynı firmadan farklı yıllarda alınan ve bire bir aynı olan ürün veya hizmetlerin arasında fiyat farklarının olabildiği ve yüksek enflasyonun olduğu dönemlerde bu fiyat farklarının da giderek arttığı anlamına gelmektedir. Özetle içerikleri bire bir aynı olan iki faturanın arasında zaman farkı arttıkça, fiyat ve vergi tutarı da artmaktadır. Bu nedenle veri setinde yer alan ‘Fatura Toplam Tutarı’ ve ‘Fatura Toplam Vergi Tutarı’ verileri veri setinden çıkarılmıştır. Bununla birlikte fiyatlarda değişiklik olsa da vergi oranlarının daha uzun süre sabit kalacağı göz önünde bulundurularak veri setine vergi oranlarının dahil edilmesine karar verilmiştir. Vergi oranları Excel uygulamasında hesaplanmış ve %17’den daha düşük olan veriler ‘0’, %17-%22 arasında olan veriler ‘1’ ve %22’nin üzerinde olan veriler ‘2’ sayısı ile değiştirilerek veri setine dahil edilmiştir. Elde edilen yeni veri seti ile model tekrar eğitilerek, verilerin eğitim sonuçlarına ne oranda etki ettikleri modele eklenen Python kod bloğu ile görselleştirilmiştir.

Şekil 36: Verilerin Modele Etkilerini Grafik Şeklinde Gösterilmesini Sağlayan Python Kod Bloğu Görseli

```
# Özellik önemliliği grafiği çizdirme
plt.figure()
plt.title("Özellik önemliliği")
plt.bar(range(X_train.shape[1]), importances[indices], yerr=std[indices], align="center")
plt.xticks(range(X_train.shape[1]), indices)
plt.xlim([-1, X_train.shape[1]])
plt.show()
```

Kaynak: Yazar tarafından derlenmiştir.

Şekil 37: Verilerin, Modelin Eğitimine Etkilerinin Grafiği



Kaynak: Yazar tarafından derlenmiştir.

Şekil 37'de eğitim setindeki verilerden hangi sütundaki verinin ne ölçüde etki ettiği yer almaktadır. Eğitime en düşük katkıyı yapan verinin 2 numara ile gösterilen veri olduğu görülmektedir. Veri seti incelendiğinde bu verinin 'Para Birimi'

bilgilerinin yer aldığı sütun olduğu anlaşılmıştır. Para birimi sütunu veri setinden çıkartılarak model tekrar eğitilmiştir. Bu işlemlerden sonra aşırı öğrenme sorununun aşıldığı tespit edilmiştir.

4.7. MODELİN KULLANILABİLECEĞİ KULLANICI ARAYÜZÜNÜN GELİŞTİRİLMESİ

Bu aşamada, hazırlanmış olan modelden faydalanılarak sınıf tahmin sonucu elde edilebilecek basit bir kullanıcı ekranı tasarlanması amaçlanmıştır. Python Flask frameworkü ile geliştirilen uygulama ekranında, modelin eğitiminde kullanılan verilerin girişlerinin yapılabileceği alanlar ve tahmin sonucunun ekrana yazdırılması için bir buton oluşturulmuştur. Eğitilmiş olan model Python'da yer alan pickle fonksiyonu kullanılarak pkl formatında kaydedilmiş ve uygulama, arayüzde veriler girilip butona tıklandığında pkl formatında kaydedilmiş olan modelden faydalanılarak sınıf tahmininin ekranda gösterileceği şekilde tasarlanmıştır.

Şekil 38: Flask Frameworkü İle Geliştirilen Uygulamada Kullanılan Python Kod Bloğu Görseli

```
import numpy as np
from flask import Flask, request, jsonify, render_template
import pickle

# Create flask app
flask_app = Flask(__name__)
model = pickle.load(open("model.pkl", "rb"))
@flask_app.route("/")
def Home():
    return render_template("index.html")

@flask_app.route("/predict", methods = ["POST"])
def predict():
    float_features = [float(x) for x in request.form.values()]
    features = [np.array(float_features)]
    prediction = model.predict(features)
```

Kaynak: Yazar tarafından derlenmiştir.

Şekil 39: Kullanıcı Giriş Ekranı



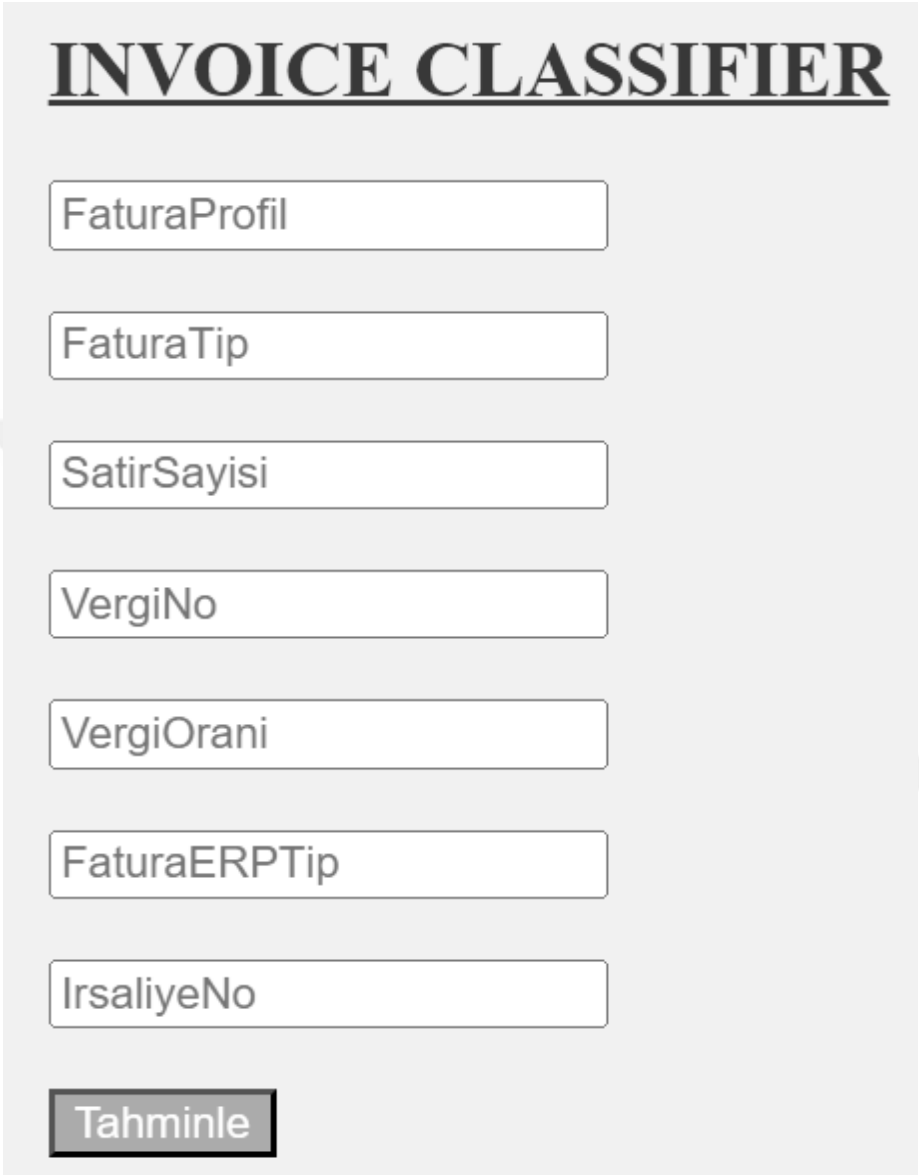
Kaynak: Yazar tarafından derlenmiştir.

Şekil 40: Giriş Ekranında Hatalı Kullanıcı Adı Ve/Veya Şifre Uyarısı



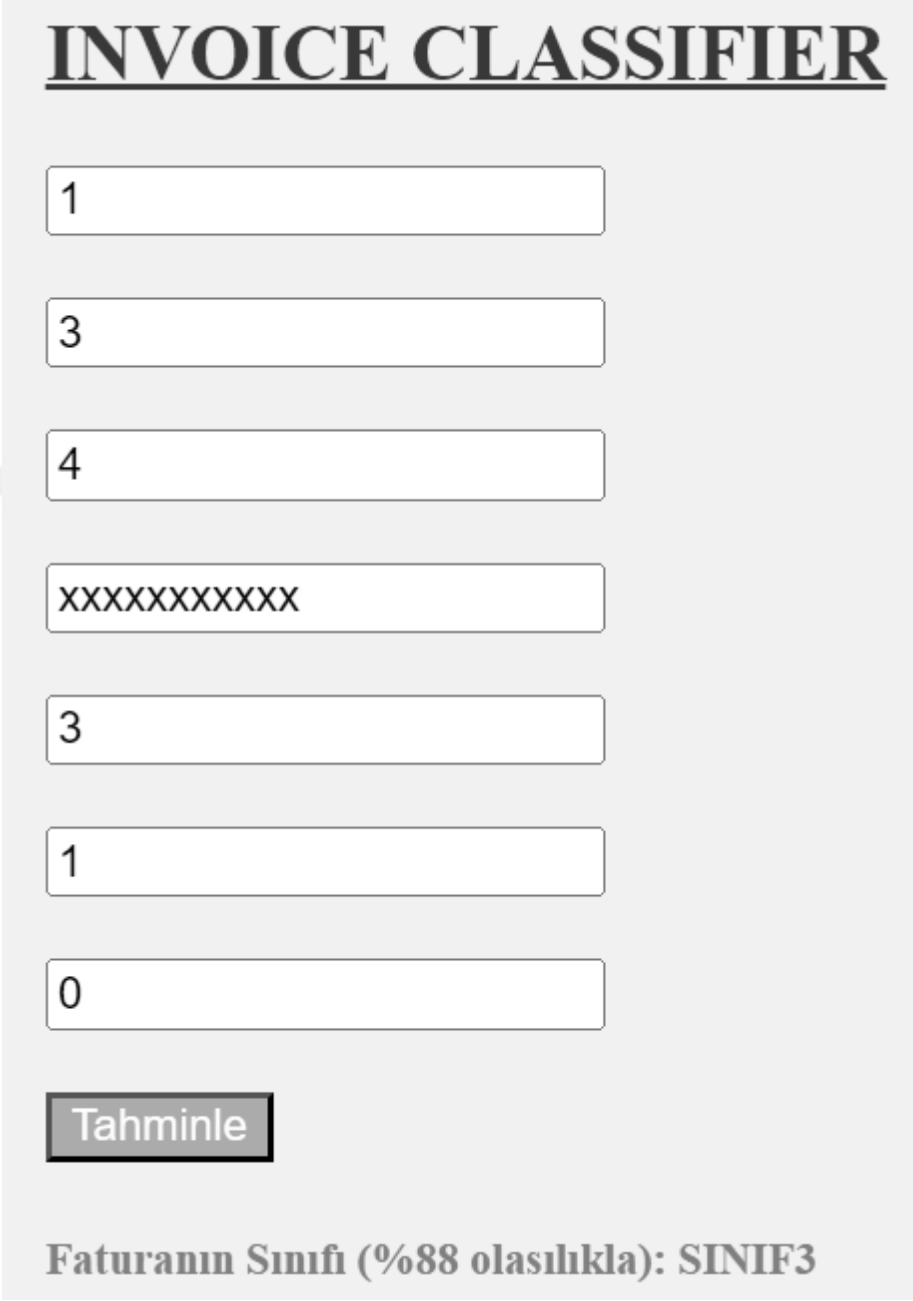
Kaynak: Yazar tarafından derlenmiştir.

Şekil 41: Geliştirilen Uygulamanın Kullanıcı Arayüzü



Kaynak: Yazar tarafından derlenmiştir.

Şekil 42: Verileri Girilen Faturanın Sınıfının Ekrana Yazdırılması



Kaynak: Yazar tarafından derlenmiştir.

SONUÇ VE ÖNERİLER

Bu çalışmada, çalışmanın birinci bölümünde 1. araştırma sorusu olarak yer alan faturaların otomatik sınıflandırılma oranının artırılması için hangi yöntemlerin uygulanabileceği konusunda bilgi verilmiş ve 2. araştırma sorusu olarak yer alan faturaların makine öğrenmesi yöntemi ile sınıflandırılmasının mümkün olup olmadığı sorusuna cevap aranmıştır. Çalışma kapsamında uygulama veri tabanında kayıtlı olan fatura verileri kullanılarak bir makine öğrenmesi çalışması gerçekleştirilmiştir. Makine öğrenmesi çalışmalarında izlenen adımlar sırasıyla verilerin toplanması, veri ön işleme, eğitim, test ve iyileştirme adımları şeklinde bu çalışma kapsamında uygulanmış ve bu adımlar çalışmada detaylı bir şekilde açıklanmıştır. Test aşamasında karşılaşılan aşırı öğrenme sorunu ve bu sorunun aşılabilmesi için uygulanabilecek adımlar hakkında bilgi verilmiş ve bu işlemlerden bazıları iyileştirme aşamasında uygulanmış ve çalışmada açıklanmıştır.

Makine öğrenmesi algoritmaları ile veri seti eğitildiğinde en yüksek başarı oranı %88 ile Karar Ağacı ve Rassal Orman algoritmaları ile elde edilmiştir. Bu algoritmalar yeni veriler girilerek test edildiğinde Karar Ağacı algoritmasında aşırı öğrenme sorununun devam ettiği gözlemlenmiş, Rassal Orman algoritmasında ise bu sorunun aşıldığı tespit edilmiştir. Bu sonuç faturaların makine öğrenmesi yöntemi ile Sınıf6 ve Sınıf7 sınıflarına ait olanlar dışındaki faturaları %88 oranında doğru sınıflandırılabileceğini göstermiştir. Eğitim verisinde dahil edilmeyen faturaların sayısı toplam fatura sayısının yaklaşık olarak %2'sini oluşturmaktadır. Eğitim verisine Sınıf6 ve Sınıf7 sınıflarına ait faturaların verilerinin dahil edilmediği dolayısıyla bu sınıflara ait faturaların model tarafından doğru sınıflandırılamayacakları göz önünde bulundurulduğunda bütün fatura sınıfları için modelin kullanılması durumunda yaklaşık olarak %2 düzeyinde bir kayıpla %86 seviyesinde bir başarı oranının elde edilebileceği anlaşılmaktadır.

Şekil 43: Rassal Orman Algoritmasının Kullanıldığı Python Kod Bloğu Görseli

```
rfc = RandomForestClassifier(n_estimators=10, criterion='entropy')
```

Kaynak: Yazar tarafından derlenmiştir.

Geliştirilen modelin kullanılarak sınıflandırma işleminin uygulamada otomatik hale getirilmesi için modelin uygulamaya entegre edilmesi gerekmektedir. Bu işlem şekil 2'de gösterilen uygulama yaşam döngüsü göz önünde bulundurularak yapılmalıdır.

Veri setinin, sisteme yeni yüklenen faturaların verileri de kullanılarak genişletilebilmesi için ise uygulama veri tabanına yeni bir tablo eklenmelidir. Bu tabloya, yüklenen faturaların verileri modelde kullanılan karşılık değerleri ile güncellenerek bu tabloya kaydedilmelidir. Tabloda eğitim veri setindeki gibi fatura sınıfı bilgisi de yer almalıdır fakat bu alan fatura işlemleri tamamlandıktan sonra tabloya dahil edilecek şekilde tasarlanmalıdır. Bu sayede ilk etapta hatalı sınıflandırılan faturaların sınıf bilgilerinin bu tabloya yanlış şekilde kaydedilmesinin önüne geçilecektir.

Python Flask ile geliştirilen arayüz uygulaması ise mevcut haliyle kullanıcılar tarafından manuel olarak faydalanılabilecek, kullanıcıların faturaların neye göre sınıflandırıldığı hakkında bilgi sahibi olmasına gerek kalmadan direkt olarak veri girişlerini yapıp modelin tahmin sonucunu elde edebilecekleri bir karar destek uygulaması niteliği taşımaktadır.

Çalışma, Yönetim Bilişim Sistemleri (YBS) açısından değerlendirildiğinde, işletmenin muhasebe departmanında karşılaşılan sorunun çözümü için uygulanabilecek çözüm yollarının araştırılması, bu çözüm yollarının avantaj ve dezavantajlarının karşılaştırılarak iş sürecinin iyileştirilmesi için güncel teknoloji ile etkin bir çözüm yolu sunulması dolayısıyla YBS çalışması özelliklerini taşımaktadır.

KAYNAKÇA

Akpınar, H. (2014). *Data: Veri Madenciliği Veri Analizi*. Papatya Yayıncılık Eğitim.

Alpaydın, E. (2004). Introduction To Machine Learning. United States Of America: MIT Press.

Aydın, C. (2018). Makine Öğrenmesi Algoritmaları Kullanılarak İtfaiye İstasyonu İhtiyacının Sınıflandırılması. *Avrupa Bilim Ve Teknoloji Dergisi*, (14) : 169-175.

Aydın, K.E. (2020). Web İçerik Sınıflandırması İçin Makine Öğrenmesi (Yayınlanmamış Yüksek Lisans Tezi). İstanbul Teknik Üniversitesi, İstanbul.

Bardelli, C., Rondinelli, A., Vecchio, R. ve Figini, S. (2020). Automatic Electronic İnvoice Classification Using Machine Learning Models. *Machine Learning And Knowledge Extraction*, *2*(4), 617-629.

Başer, B. Ö., Yangın, M. ve Sarıdaş, E. S. (2021). Makine Öğrenmesi Teknikleriyle Diyabet Hastalığının Sınıflandırılması. *Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, *25*(1), 112-120.

Berry, M. J. ve Linoff, G. S. (2004). *Data Mining Techniques: For Marketing, Sales, And Customer Relationship Management*. John Wiley & Sons.

Berry, M. J. ve Linoff, G. S. (2004). *Data Mining Techniques: For Marketing, Sales, And Customer Relationship Management*. John Wiley & Sons.

Çelik, M. (2009). Veri Madenciliğinde Kullanılan Sınıflandırma Yöntemleri Ve Bir Uygulama. *Yayınlanmamış Yüksek Lisans Tezi*.

Dangeti, P. (2017). *Statistics For Machine Learning*. Packt Publishing Ltd.

Dennis, A., Wixom, B. H., Roth, R. M. (2014). System Analysis And Design (5.Baskı: USA: John Wiley & Sons, Inc.

Fatima, M. Ve Pasha, M. (2017). Survey Of Machine Learning Algorithms For Disease Diagnostic. Journal Of Intelligent Learning Systems And Applications, 9(01: 1.

Filiz, E., Karaboğa, H. A. ve Akoğul, S. (2017). Bıst-50 Endeksi Değişim Değerlerinin Siniflandirilmasinda Makine Öğrenmesi Yöntemleri Ve Yapay Sinir Ağlari Kullanimi. *Çukurova Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, *26*(1), 231-241.

Fu, Y., Shen, R. ve Lu, H. (2004). Watermarking Scheme Based On Support Vector Machine For Colour İmages. *Electronics Letters*, *40*(16), 986-987.

GİB. (2023). *e-Fatura Hakkında*. https://ebelge.gib.gov.tr/efaturahakkinda.html

Gollapudi, S. Ve Laxmikanth, V. (2016). Practical Machine Learning. Birmingham, UK: Packt Publishing.

Hussung, T. (2016). What Is The Software Development Life Cycle? Https://Online.Husson.Edu/Software-Development-Cycle/, 17.06.2023

Karadağ, B., Bölükbaş, O. ve Ünal, M. A. (2020). Makine Öğrenmesi İle Bireysel Müşteriler İçin Finansman Ürün Önerilmesi. *Academic Perspective Procedia*, *3*(1), 438-444.

Kavuncu, S.K. (2018). Makine Öğrenmesi Ve Derin Öğrenme: Nesne Tanıma Uygulaması (Yayınlanmamış Yüksek Lisans Tezi). Kırıkkale Üniversitesi, Kırıkkale.

Kazan, S. ve Karakoca, H. (2019). Makine Öğrenmesi İle Ürün Kategorisi Sınıflandırma. *Sakarya University Journal Of Computer And Information Sciences*, *2*(1), 18-27.

Kendall, K. E., Kendall, J. E., Kendall, K. E. ve Kendall, J. A. (2011). *Systems analysis and design* (Vol. 8, p. 20). Upper Saddle River, NJ: Pearson Prentice Hall.

Kılınç, M. (2018). Dinamik İşletme Verilerine Göre Kısa Ve Orta Vadeli Planların Yapılmasını Sağlayan Web Tabanlı Bir Uygulama Geliştirilmesi. (Yayınlanmamış Yüksek Lisans Tezi). İzmir: Dokuz Eylül Üniversitesi.

Lu, W., Wang, W., Leung, A. Y., Lo, S. M., Yuen, R. K., Xu, Z. ve Fan, H. (2002, May). Air Pollutant Parameter Forecasting Using Support Vector Machines. In *Proceedings Of The 2002 International Joint Conference On Neural Networks. IJCNN'02 (Cat. No. 02CH37290)* (Vol. 1, Pp. 630-635). IEEE.

Michalski, R. S. ve Kodratoff, Y. (1990). Research in machine learning: Recent progress, classification of methods, and future directions. *Machine learning*, 3-30.

Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. MIT Press.

Niculescu-Mizil, A., Caruana, R. (2005). Predicting Good Probabilities With Supervised Learning. Proceedings Of The 22nd International Conference On Machine Learning (Ss. 625-632). 07–11 August 2005, Bonn.

Özkan, Y. (2008). Veri Madenciliği Yöntemler (3.Baskı). İstanbul: Papatya Yayıncılık.

Sakal, M. (2021). Makine Öğrenmesi Algoritmaları Kısa Açıklamaları. http://muratsakal.com/?p=230 murat sakal, 12.06.2023

Sariman, G. (2011). Veri madenciliğinde kümeleme teknikleri üzerine bir çalışma: k-means ve k-medoids kümeleme algoritmalarının karşılaştırılması. *Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, *15*(3), 192-202.

Sebetci, Ö., Günay, M. B. ve Sebetci, E. (2018). İş Süreç Yönetimi (Bpm) Ve İş Akış Yönetimi (Wfm) Kavramlarına Yaklaşım. *Ajıt-E: Academic Journal Of Information Technology*, *9*(33), 115-126.

Seker, S. E. (2014). Yazılım Geliştirme Hayat Döngüsü (Software Development Life Cycle). *Ybs Ansiklobedi, 1 (2)*.

Silahtaroğlu, G. (2013). Kavram Ve Algoritmalarıyla Veri Madenciliği Kavram Ve Algoritmaları (2.Baskı). İstanbul: Papatya Yayıncılık.

Tecim, V. (2023). *Yönetim Bilişim Sistemleri (YBS)*. https://vahaptecim.com.tr/yonetimbilisimsistemleri/#:~:text=YBS%2C%20bir%20kurulu%C5%9Fun%20operasyonlar%C4%B1n%C4%B1n%20belkemi%C4%9Fi,kurumun%20fonksiyonlar%C4%B1n%C4%B1%20sa%C4%9Fl%C4%B1kl%C4%B1%20y%C3%BCr%C3%BCtebilmesi%20m%C3%BCmk%C3%BCnd%C3%BCr., 12.06.2023.

Tecim, V. (2023a). *Sistem Geliştirme Yaşam Döngüsü.* https://vahaptecim.com.tr/sistem-gelistirme-yasam-dongusu/, 12.06.2023.

Tektüfekçi, F. (2012). Bilgi Teknolojilerinin Muhasebe Uygulamalarına Entegrasyonu Ve Bütünleşik Sistemlerle Olan Etkileşim. *Organizasyon ve Yönetim Bilimleri Dergisi*, *4*(2), 51-59.

Uzun, E. (2007). İnternet Tabanlı Bilgi Erişimi Destekli Bir Otomatik Öğrenme Sistemi (Yayınlanmamış Doktora Tezi). Trakya Üniversitesi, Edirne.

Yaralıoğlu, K. (2004). *Uygulamada Karar Destek Yöntemleri*. İzmir: İlkem Ofset.