Makine Öğrenmesi Algoritmalarını Kullanarak Kredi Kartı Dolandırıcılığının Tespiti

Yazılım Mühendisliği Ana Bilim Dalı

Dönem Projesi

Ali ASLAN

Proje Danışmanı: Prof. Dr. Femin Yalçın Küçükbayrak

Ocak 2024

Makine Öğrenmesi Algoritmalarını Kullanarak Kredi Kartı Dolandırıcılığının Tespiti

Öz

Dünya ekonomisinin hızla büyümesi ve finans sektörünün bu büyüme trendine ayak uydurması, yeni güvenlik sorunlarını beraberinde getirmiştir. Finansal işlem hacminin artmasıyla birlikte dijital işlemlerin yaygınlaşması, dolandırıcılar için çekici bir alan oluşturarak dolandırıcılık faaliyetlerini artırmıştır. Bu bağlamda, kredi kartı dolandırıcılığı dijital finansal dünyanın en önemli güvenlik tehditlerinden biri haline gelmiştir.

Dolandırıcılar, kişisel ve finansal bilgilere ulaşmak adına gelişmiş teknikler kullanarak kredi kartı dolandırıcılığı gibi suçları gerçekleştirmektedirler. Dolandırıcılık faaliyetlerindeki bu gelişmiş yöntemler, klasik güvenlik önlemlerinin yetersiz kaldığı bir ortam yaratmıştır.

Bu durum, klasik yöntemlerin ötesinde gelişmiş tespit stratejilerini zorunlu kılmış ve günümüz teknolojisinin avantajlarından yararlanma ihtiyacını doğurmuştur.

Tüm gelişmeler göz önünde bulundurulduğunda, veri bilimi teknikleri, büyük veri analitiği ve yapay zekâ (AI) gibi alanlar, kredi kartı dolandırıcılığıyla mücadelede önemli bir rol oynamaktadır. Bu teknikler, kredi kartı dolandırıcılığı veri setleri üzerinde anormal işlem desenlerini ve dolandırıcılık göstergelerini belirlemeyi hedeflemektedir.

Bu çalışma, kredi kartı dolandırıcılığını önlemek ve tespit etmek amacıyla kullanılan makine öğrenmesi (ML) tekniklerinin etkinliğini incelemektedir. Kullanılan algoritmalar arasında Destek Vektör Makineleri (SVM), Rastgele Orman (RF), Naive Bayes (NB), Lojistik Regresyon (LR), K-En Yakın Komşu (KNN) ve Karar Ağaçları (DT) bulunmaktadır.

Analiz sonuçlarına göre, Rastgele Orman, K-En Yakın Komşu ve Karar Ağaçları algoritmaları %99 doğruluk oranı ile öne çıkmaktadır. Bu algoritmaları %98 doğruluk oranı ile Destek Vektör Makineleri, %94 doğruluk oranı ile Lojistik Regresyon takip etmiştir. Naive Bayes algoritması en düşük performansı göstermiş ve %91 doğruluk oranın ulaşmıştır.

Modellerin seçimi, özellikle eğitim süresi, karmaşıklık ve anlaşılabilirlik gibi faktörlerin göz önünde bulundurulması ile belirlenmelidir. Her bir algoritmanın avantajları ve dezavantajları vardır; bu nedenle, finans sektörü ve benzeri kurumlar için en uygun modelin seçilmesi, kredi kartı dolandırıcılığıyla mücadelede başarılı sonuçlar elde etmede kritik öneme sahiptir.

Sonuç olarak bu çalışma, kredi kartı dolandırıcılığını tespit etmek ve önlemek için kullanılan makine öğrenmesi (ML) tekniklerinin etkinliğini değerlendirmekte olup, finansal kurumlar ve işletmelerin daha güvenli bir deneyim sunmasını sağlamayı amaçlamaktadır. Elde edilen sonuçlar, gelecekteki araştırmalara ilham kaynağı olabilir ve kredi kartı dolandırıcılığıyla mücadelede daha ileri düzeyde stratejilerin geliştirilmesine katkıda bulunabilir.

Anahtar Sözcükler: Kredi Kartı Dolandırıcılığı, Dolandırıcılık Tespiti, Yapay Zekâ, Makine Öğrenmesi, Sınıflandırma Algoritmaları

Credit Card Fraud Detection Using Machine Learning Algorithms

Abstract

The rapid growth of the world economy and the financial sector keeping pace with this growth trend have brought new security challenges. With the increase in the volume of financial transactions, the widespread use of digital transactions has created an attractive area for fraudsters and increased fraudulent activities. In this context, credit card fraud has become one of the most important security threats in the digital financial world.

Fraudsters commit crimes such as credit card fraud by using advanced techniques to access personal and financial information. These advanced methods in fraudulent activities have created an environment where classical security measures are insufficient.

This situation has necessitated advanced detection strategies beyond classical methods and created the need to utilize the advantages of today's technology.

Considering all developments, areas such as data science techniques, big data analytics and artificial intelligence play an important role in the fight against credit card fraud. These techniques aim to identify anomalous transaction patterns and fraud indicators on credit card fraud datasets.

This study examines the effectiveness of machine learning techniques used to prevent and detect credit card fraud. The algorithms used include Support Vector Machines (SVM), Random Forest (RF), Naive Bayes (NB), Logistic Regression (LR), K-Nearest Neighbor (KNN) and Decision Trees (DT).

According to the analysis results, Random Forest, K-Nearest Neighbor and Decision Trees algorithms stand out with 99% accuracy. These algorithms were followed by Support Vector Machines with 98% accuracy and Logistic Regression with 94% accuracy. Naive Bayes algorithm showed the lowest performance and reached 91% accuracy.

The choice of models should be determined by considering factors such as training time, complexity and comprehensibility. Each algorithm has its advantages and disadvantages; therefore, choosing the most appropriate model for the financial sector and similar organizations is critical in achieving successful results in the fight against credit card fraud.

In conclusion, this study evaluates the effectiveness of machine learning techniques for detecting and preventing credit card fraud and aims to provide a safer experience for financial institutions and businesses. The results may inspire future research and contribute to the development of further strategies to combat credit card fraud.

Keywords: Credit Card Fraud, Fraud Detection, Artificial Intelligence, Machine Learning, Classification Algorithms

İçindekiler

Şekiller Listesi

Tablolar Listesi

Kısaltmalar Listesi

AI	Artificial Intelligence
ANN	Artificial Neural Networks
ATM	Automated Teller Machine
CNN	Convolutional Neural Networks
CVC	Card Verification Code
DT	Decision Tree
FN	False Negative
FP	False Positive
K-NN	K-Nearest Neighbors
LR	Logistic Regression
MFA	Multi-factor authentication
ML	Machine Learning
NB	Naive Bayes
OTP	One Time Password
RF	Random Forest
SMOTE	Synthetic Minority Over-Sampling Technique
SVM	Support Vector Machines
TP	True Positive
TN	True Negative

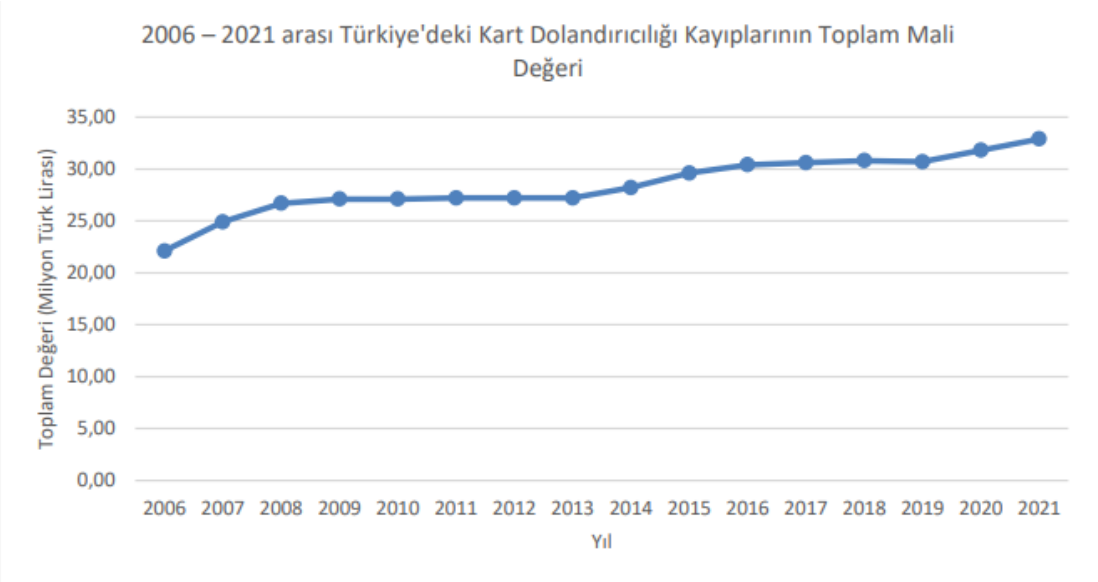
Bölüm 1

Giriş

Bugünün modern dünyasında, teknolojik ilerlemeler ve dijital dönüşüm, finansal işlemleri kolaylaştırırken aynı zamanda çeşitli güvenlik riskleri de beraberinde getirmektedir. Özellikle kredi kartları, online alışveriş, dijital ödemeler ve e-ticaret gibi alanlarda yaygın olarak kullanılan bir ödeme yöntemi haline gelmiştir. Ancak, bu yaygın kullanım, kredi kartı dolandırıcılığının artmasına neden olmuştur (Awoyemi, Adetunmbi., Oluwadare, 2017).

Kredi kartı dolandırıcılığı, bireylerin veya kurumların kredi kartı bilgilerini izinsiz olarak ele geçirerek, haksız kazanç elde etme amacıyla kullanması sürecidir. Bu süreç sahte alışverişler yapmak, çevrimiçi hizmetlere üye olmak veya başkalarının hesaplarına para transfer etmek gibi işlemleri içermektedir. Bu tür dolandırıcılık vakaları, finansal kayıplara ve mağduriyetlere neden olabilmektedir. Aynı zamanda, bireylerin ve işletmelerin itibarını zedelerken, güvenlik açıklarının da göstergesi olarak kabul edilmektedir.

Kredi kartı dolandırıcılığı, geleneksel suç yöntemlerinin yanı sıra, teknolojinin ilerlemesiyle birlikte yeni ve sofistike taktikler geliştirmiştir. Dolandırıcılar, kimlik avı, kart kopyalama ve manyetik şerit bilgilerinin kopyalanması gibi yöntemlerle kişisel ve finansal bilgilere ulaşmaya çalışmaktadır (Yero, 2018). Bu durum, bireylerin ve işletmelerin güvenliğini tehlikeye atarak, ekonomik kayıplara ve zaman kaybına neden olmaktadır. Kredi kartı dolandırıcılığının ekonomik zararına örnek olması açısından Şekil 1.1'de 2006 – 2021 yılları arasında Türkiye'deki kart dolandırıcılığı kayıplarının toplam mali değeri belirtilmiştir.



Şekil 1.1: 2006 – 2021 arası Türkiye'deki Kart Dolandırıcılığı Kayıplarının Toplam Mali Değeri

Bu alandaki gelişmeleri anlamak ve etkili çözümler üretmek, finansal kurumlar ve toplumlar için hayati bir önem taşımaktadır. Teknolojinin nimetlerinden faydalanarak, finansal dolandırıcılığı önleme ve tespit etme noktasında yapay zekâ (AI) ve makine öğrenmesi (ML) gibi yenilikçi teknolojilere başvurmak, bu mücadelede kritik bir rol oynamaktadır.

Yapay zekâ (AI) ve makine öğrenimi (ML), günümüzde büyük veri analizi alanında önemli bir role sahiptir. Bu teknolojiler, finansal işlemleri anlık olarak izleyip dolandırıcılık faaliyetlerini tespit edebilme yeteneği ile öne çıkar. Her geçen gün gelişen dolandırıcılık yöntemlerine karşı esneklik sağlayan yapay zekâ (AI) ve makine öğrenimi (ML), otomatik aksiyonlar alabilme kabiliyetiyle de operasyonel verimliliği artırır (Phua, Lee, Smith, & Gayler, 2010).

Bu çalışmada, yapay zekâ (AI) ve makine öğrenimi (ML) ile dolandırıcılık tespiti konusu detaylı bir şekilde incelenecektir. Kredi kartı dolandırıcılığıyla birlikte dolandırıcıların bu alanda kullandığı yöntemlerden bahsedilecektir. Ayrıca makine öğrenmesi (ML) üzerinde durularak temel prensipleri açıklanacaktır. Makine öğrenmesi (ML) ile kredi kartı dolandırıcılığının tespiti arasındaki ilişki incelenecek ve bu alandaki yöntemler, modeller ve uygulamalar anlatılacaktır.

Bölüm 2

Genel Bilgiler

2.1 Kredi Kartı Dolandırıcılığı Türleri

Kredi kartı dolandırıcılığı, kart kullanım oranın artması ve teknolojinin gelişmesi nedeniyle çok çeşitli yollar ile gerçekleştirebilir hale gelmiştir. Bu tür dolandırıcılık vakaları çeşitli yöntemlerle gerçekleştirilir ve hem bireyleri hem de işletmeleri etkileyebilir. Dolandırıcılık türleri arasında sahte kartlar, boş plastik kartlar, çalıntı kartlar, kart kopyalama, manyetik şeridi değiştirme, ATM dolandırıcılığı, kimlik avı yöntemi, işyeri iş birliği ve veri ihlali gibi farklı yöntemler bulunmaktadır.

2.1.1 Sahte Kart

Sahte kart kullanımı, dolandırıcıların kredi veya banka kartlarını sahte olarak üretip kullanarak hileli işlemler gerçekleştirdiği bir dolandırıcılık yöntemidir. Bu tür dolandırıcılık genellikle kart kopyalama gibi yöntemlerle elde edilen bilgilerle sahte kartlar üretilerek yapılır.

Dolandırıcılar, sahte kartları kullanarak alışveriş yapabilir, para çekebilir veya diğer finansal işlemleri gerçekleştirebilirler. Sahte kartlar, genellikle orijinal kartların bilgilerinin kopyalanması veya üçüncü taraflardan elde edilen bilgilerle üretilir. Bunlar, manyetik şeritler deki bilgilerin kopyalanması, çalınan kart bilgilerinin basılması veya çalınan kart bilgilerinin online alışverişlerde kullanılması gibi yöntemlerle oluşturulabilir.

2.1.2 Boş Plastik Kart

Boş plastik kart dolandırıcılığı, dolandırıcıların kredi veya banka kartı gibi finansal kartların üzerine sahte bilgiler yazarak hileli işlemler gerçekleştirdiği bir dolandırıcılık türüdür. Bu tür dolandırıcılık genellikle kart kopyalama veya çalıntı kart bilgileriyle ilişkilendirilebilir.

Dolandırıcılar, boş plastik kartları genellikle kredi veya banka kartı gibi gerçek kartlara benzer şekilde üretirler. Ancak, bu kartların içinde geçerli bir banka hesabı veya finansal bilgi bulunmaz. Dolandırıcılar, bu boş kartların üzerine sahte kart numaraları, son kullanma tarihleri ve güvenlik kodları gibi bilgileri yazarak, gerçek gibi görünmelerini sağlarlar.

Boş plastik kart dolandırıcılığı, genellikle ATM'lerde veya mağazalarda alışveriş yaparken kullanılır. Dolandırıcılar, sahte kartları bu işlemler de kullanarak mal veya hizmet alırlar veya nakit para çekerler. Bu tür dolandırıcılık, kart sahiplerinin banka hesaplarını ve finansal güvenliğini tehlikeye atabilir ve ciddi maddi zararlara neden olabilir.

2.1.3 Çalıntı Kart

Çalıntı kart dolandırıcılığı, dolandırıcıların başkalarının kredi veya banka kartlarını çalarak hileli işlemler gerçekleştirdiği bir dolandırıcılık yöntemidir. Bu tür dolandırıcılık, kart sahibinin farkına varmadan gerçekleşir ve genellikle kart sahibinin maddi zarara uğramasına neden olur.

Dolandırıcılar, kartları çalarak veya ele geçirerek elde ettikleri kart bilgilerini kullanarak alışveriş yapabilir, para çekebilir veya diğer finansal işlemleri gerçekleştirebilirler. Bu tür dolandırıcılıkta, kart sahibi genellikle kısa sürede farkına varmaz ve dolandırıcılar bu süre zarfında büyük miktarda para çekebilir veya harcayabilirler.

2.1.4 Kart Kopyalama

Kart kopyalama, dolandırıcıların hileli bir şekilde insanların kredi veya banka kartı bilgilerini çalmak için kullandığı bir yöntemdir. Bu dolandırıcılık türünde, dolandırıcılar ATM'lerde veya diğer kart okuyucu cihazlarda sahte cihazlar kullanarak kart bilgilerini kopyalarlar. Bu sahte cihazlar, kartın manyetik şeridinden bilgileri alarak dolandırıcıların eline geçirir.

Genellikle, kart kopyalama cihazları kart okuyuculara takılır ve bu cihazlar kartın manyetik şeridinden bilgileri kopyalar. Dolandırıcılar daha sonra bu kopyaladıkları bilgileri sahte kartlar oluşturmak veya online alışverişlerde kullanmak için kullanabilirler. Ayrıca, dolandırıcılar kredi kartı numarası, son kullanma tarihi ve güvenlik kodu gibi bilgileri de ele geçirmeye çalışabilirler.

2.1.5 Manyetik Şeridi Değiştirme

Dolandırıcılar, kredi veya banka kartlarının manyetik şeridini değiştirme yöntemini kullanarak hileli işlemler gerçekleştirmeye çalışırlar. Bu dolandırıcılık yönteminde, dolandırıcılar sahte kart oluşturmak için encoder adı verilen bir cihaz kullanarak kendi bilgilerini kredi kartının manyetik şeridine yüklerler. Bu işlem sonucunda, kartın üzerindeki orijinal bilgiler silinir ve kart sahibinin gerçek bilgileri yerine dolandırıcının bilgileri bulunur.

Manyetik şeridi değiştirme yöntemi, diğer kredi kartı dolandırıcılık yöntemlerine göre daha zor tespit edilebilir. Çünkü alışveriş sırasında, mağaza görevlisi veya POS cihazı, kartın manyetik şeridindeki bilgilerin gerçek mi yoksa sahte mi olduğunu kontrol etmekten ziyade sadece kartı okur ve işlemi gerçekleştirir. Dolandırıcılar, sahte kartları kullanarak alışveriş yapabilir, para çekebilir veya diğer finansal işlemleri gerçekleştirebilirler.

2.1.6 ATM Dolandırıcılığı

ATM dolandırıcılığı, insanların banka hesaplarından para çalmak amacıyla ATM'leri kullanarak gerçekleştirilen hileli veya yasa dışı faaliyetlerdir. Dolandırıcılar, genellikle ATM'lere sahte cihazlar yerleştirerek veya kullanıcıların kart bilgilerini ve

şifrelerini çalarak bu dolandırıcılığı gerçekleştirirler (Adeel & Hussain, 2017). Örneğin, kart kopyalama cihazları kullanılarak, ATM'lerin kart okuyucularına sahte cihazlar monte edilir ve böylece kart bilgileri kopyalanır.

Ayrıca, klavye yakalama yöntemiyle ATM'lerin tuş takımlarına sahte klavye kapakları veya kamera sistemleri yerleştirilerek, kullanıcıların PIN numaraları ele geçirilebilir. Dolandırıcılar, bu yolla elde ettikleri bilgileri kullanarak hesaplardan para çekerler veya başka yasa dışı işlemler gerçekleştirirler.

2.1.7 Kimlik Avı Yöntemi

Kimlik avı, dolandırıcıların sahte web siteleri, e-postalar veya mesajlar aracılığıyla insanların kişisel bilgilerini (kullanıcı adları, şifreler, kredi kartı bilgileri vb.) ele geçirmeye çalıştığı bir dolandırıcılık yöntemidir. Kimlik avı genellikle kurumsal kuruluşların veya finansal kurumların adını kullanarak yapılan sahte iletişimlerle gerçekleştirilir.

Bu dolandırıcılık yönteminde, dolandırıcılar, masum insanları gerçek bir kurum veya kuruluş gibi davranan sahte web sitelerine veya e-postalara yönlendirirler. Bu sahte iletişimler genellikle meşru bir şekilde görünür ve insanların güvenini kazanmak için kurum logosu, resmi renkler ve dilbilgisi hataları gibi unsurlar kullanılır. İnsanlar, bu sahte iletişimler aracılığıyla kişisel bilgilerini girmeye veya paylaşmaya teşvik edilirler.

2.1.8 İşyeri İş birliği

Bu yöntemde, dolandırıcılar işyerlerinde çalışanlarla veya işletme sahipleriyle iş birliği yaparlar ve bu kişilerin yardımıyla kredi kartı bilgilerini çalarlar veya sahte işlemler gerçekleştirirler. İş birliği yapan kişiler, ödül veya para karşılığında dolandırıcılara müşteri bilgilerini veya kart bilgilerini sağlayabilirler.

Örneğin, bir restoran veya mağaza çalışanı, müşterilerin kredi kartı bilgilerini çalmak veya kart bilgilerini kopyalamak için özel bir cihaz kullanabilir. Bu bilgiler daha sonra dolandırıcılar tarafından kötüye kullanılabilir. Aynı şekilde, işletme sahipleri veya yöneticiler de işyerinde gerçekleştirilen sahte işlemlere göz yumarak dolandırıcılığa ortak olabilirler.

2.1.9 Veri İhlali

Veri ihlali, bir kuruluşun veya bireyin kontrolü altındaki hassas bilgilerin izinsiz olarak erişilmesi, çalınması, ifşa edilmesi veya kaybolması durumudur. Bu tür bilgiler genellikle müşteri bilgileri, kredi kartı numaraları, sosyal güvenlik numaraları veya tıbbi kayıtlar gibi kişisel veya hassas bilgileri içerir.

Veri ihlalleri, genellikle siber saldırılar, kötü amaçlı yazılımlar, sistem zafiyetleri veya çalışan hataları gibi faktörler nedeniyle meydana gelir. Bir kuruluşun veri tabanı veya ağ sistemlerine yetkisiz erişim sağlanması sonucunda, kredi kartı numaraları gibi hassas bilgilerin çalınması veya ifşa edilmesi mümkün hale gelir. Bu ihlaller hem kuruluşun hem de bilgilerin sahiplerinin güvenliğini tehlikeye atar ve ciddi maddi ve itibari zararlara neden olabilir.

2.2 Kredi Kartı Dolandırıcılığının Tespiti Kavramı

Kredi kartı dolandırıcılığı vakaları hem bireyler hem de işletmeler için ciddi maddi kayıplara ve itibar zedelenmesine neden olmaktadır. Kredi kartı dolandırıcılığının tespiti, işletmelerin ve finansal kurumların güvenlik önlemlerini güçlendirmesi ve dolandırıcılık riskini azaltması için kritik bir öneme sahiptir.

Dolandırıcılığın tespiti, kredi kartı işlemlerindeki şüpheli veya anormal aktiviteleri belirlemeye yönelik karmaşık süreçler ve tekniklerin uygulanmasıdır. Bu süreçler, teknolojik çözümleri ve insan faktörünü içeren stratejileri kapsar. Bu teknikler, bankalar, ödeme işlemcileri ve perakende işletmeleri tarafından kullanılır.

2.2.1 Kredi Kartı İşlem Analizi

Kredi Kartı İşlem Analizi, finansal kuruluşlar ve işletmeler tarafından kullanılan önemli bir dolandırıcılık tespit yöntemidir. Bu yöntem, kredi kartı işlemlerinin detaylı bir şekilde incelenmesini ve analiz edilmesini içerir.

İşlem analizi, bir kartın kullanım geçmişini, işlem miktarlarını, işlem tarihlerini, işlem yerlerini ve diğer ilgili verileri inceleyerek gerçekleştirilir. Bu analiz, alışılmadık veya şüpheli işlemlerin belirlenmesi için önemli bir araçtır. Örneğin, bir kartın normal

alışveriş alışkanlıklarından farklı bir işlem yapması veya bir kartın sıra dışı bir şekilde büyük miktarda para çekmesi gibi durumlar, işlem analizi tarafından potansiyel dolandırıcılık işaretleri olarak tanımlanabilir.

Kredi Kartı İşlem Analizi, genellikle otomatik olarak gerçekleştirilen bir süreçtir ve büyük miktarda işlem verisini analiz etmek için karmaşık veri analizi ve algoritmalar kullanır. Bu analiz, işletmelerin ve finansal kuruluşların dolandırıcılık vakalarını tespit etmelerine yardımcı olur ve müşterilerin hesaplarını korumak için önemli bir rol oynar.

2.2.2 Şüpheli Davranış Analizi

Şüpheli Davranış Analizi, finansal işlemlerde belirli davranış kalıplarının incelenerek potansiyel dolandırıcılık işaretlerinin tespit edilmesi için kullanılan bir yöntemdir. Dolandırıcılar genellikle belirli davranış kalıpları sergilerler ve bu kalıplar, şüpheli işlemleri belirlemek için kullanılabilir. Bu analiz, bir kartın normal alışveriş alışkanlıklarından farklı bir işlem yapması veya bir kartın sıra dışı bir şekilde büyük miktarda para çekmesi gibi durumları inceler. Örneğin, aynı kişinin farklı hesaplardan aynı anda sıra dışı işlemler yapması veya normal harcama alışkanlıklarında ani değişikliklerin olması gibi durumlar, şüpheli davranışlar olarak değerlendirilir.

Teknolojik gelişmelerle birlikte, şüpheli davranış analizi daha da geliştirilmiştir. Yapay zekâ (AI) ve makine öğrenimi (ML) gibi teknolojilerin kullanımı, analizin daha etkili ve hassas hale gelmesine olanak tanır. Bu sayede, işletmeler dolandırıcılık vakalarını daha hızlı bir şekilde tespit edebilir ve müşterilerin hesaplarını daha etkili bir şekilde koruyabilir.

2.2.3 İzleme ve Uyarı Sistemleri

İzleme ve Uyarı Sistemleri, kredi kartı işlemlerini sürekli olarak izler ve anormal aktiviteleri tespit etmek için tasarlanmıştır. İzleme ve uyarı sistemleri, şüpheli işlemleri tanımlamak ve ilgili taraflara hızlı bir şekilde uyarılar göndermek için kullanılır. Bu sistemler genellikle otomatik olarak çalışır ve büyük miktarda işlem verisini analiz etmek için karmaşık algoritmalar ve veri analizi teknikleri kullanır (Shen & Hsiao, 2021).

İzleme ve uyarı sistemleri, işletmelerin dolandırıcılık vakalarını hızlı bir şekilde tespit etmelerine yardımcı olur ve müşterilerin hesaplarını korumak için önemli bir rol oynar. İzleme ve uyarı sistemleri, çeşitli parametreler ve önceden tanımlanmış kural setleri kullanarak şüpheli işlemleri tanımlar.

Örneğin, bir kartın alışveriş alışkanlıklarında ani ve belirgin bir değişiklik olması veya bir kartın normalden farklı bir coğrafi bölgeden işlem yapması gibi durumlar, uyarı sistemleri tarafından tespit edilir ve ilgili taraflara bildirilir.

2.2.4 Risk Değerlendirme ve İşlem İzleme

Risk Değerlendirme ve İşlem İzleme, potansiyel dolandırıcılık risklerini belirlemek ve işlem güvenliğini sağlamak için tasarlanmıştır. Risk değerlendirmesi, işletmelerin işlem yaparken karşılaştığı riskleri belirlemek ve bunları değerlendirmek için kullanılır. Bu süreçte, işletmelerin karşılaştığı potansiyel riskler analiz edilir ve işlem güvenliğini etkileyebilecek faktörler değerlendirilir. Örneğin, bir kartın alışveriş alışkanlıklarında ani bir değişiklik olması veya bir kartın sıra dışı bir şekilde büyük miktarda para çekmesi gibi durumlar, risk değerlendirmesi sürecinde dikkate alınabilir.

İşlem izleme, işletmelerin gerçekleşen işlemleri sürekli olarak izlemesi ve anormal aktiviteleri tespit etmesi için tasarlanmıştır. Bu süreçte, işletmelerin gerçekleşen işlemleri yakından takip etmesi ve potansiyel dolandırıcılık işaretlerini belirlemesi önemlidir. İşlem izleme, otomatik olarak çalışabilen sistemler aracılığıyla gerçekleştirilir ve şüpheli işlemlerin tespit edilmesine yardımcı olur.

2.2.5 Yasal Düzenlemeler ve Uyumluluk

Kredi kartı dolandırıcılığı ile mücadelede yasal düzenlemeler ve uyumluluk standartları, işletmelerin dolandırıcılıkla mücadelede etkili bir şekilde hareket etmelerini sağlar. Bu süreç, işletmelerin kredi kartı işlemleri için geçerli yasal düzenlemelere uyum sağlamalarını ve uygun güvenlik protokolleri ve önlemlerini uygulamalarını gerektirir. Yasal düzenlemeler genellikle müşteri bilgilerinin korunması, dolandırıcılık önleme önlemleri ve müşteri hakları gibi konuları kapsar (Santos & Maynard, 2018).

Uyumluluk süreci, işletmelerin kredi kartı işlemleri için geçerli yasal düzenlemelere uyum sağladığını ve gereken güvenlik önlemlerini aldığını doğrulamak için tasarlanmıştır. Bu süreçte, işletmeler genellikle dış denetimlerden geçer ve uygunluklarını kanıtlamak için belgelendirme süreçlerine tabi tutulurlar.

Yasal düzenlemeler ve uyumluluk standartları, işletmelerin dolandırıcılıkla mücadelede etkili bir şekilde hareket etmelerini sağlar ve müşterilerin güvenliğini korumak için önemlidir. Bu süreçlerin etkin bir şekilde uygulanması, işletmelerin dolandırıcılık risklerini minimize etmelerine ve müşteri bilgilerinin güvenliğini sağlamalarına olanak tanır.

2.2.6 Kart Sahibi ve Kimlik Doğrulama

Kart sahibi ve kimlik doğrulama sürecinde, çeşitli yöntemler kullanılır. Bunlar arasında, kart doğrulama kodu (CVC), çok faktörlü kimlik doğrulama (MFA), tek kullanımlık şifreler (OTP) ve kart sahibinin adıyla eşleşen bir güvenlik kodu girmesi gibi ek doğrulama adımları bulunur. Bu tür yöntemler, kart sahibinin kimliğini doğrulamak için etkili bir yol sağlar.

Çok faktörlü kimlik doğrulama (MFA), kart sahibinin kimliğini doğrulamak için kullanılan etkili bir yöntemdir. Bu yöntemde, kart sahibinin kullanıcı adı ve şifresiyle birlikte bir SMS veya e-posta yoluyla gönderilen bir doğrulama kodunu girmesi istenebilir. Bu, kart sahibinin gerçek bir kişi olduğunu ve doğru bilgilere sahip olduğunu doğrular.

Güvenlik protokolleri, kart sahibi ve kimlik doğrulama sürecinde önemli bir rol oynar. Özellikle online alışveriş işlemlerinde kullanılan 3D Secure gibi protokoller, kart sahibinin işlemi gerçekleştirdiğini doğrulamak için ek bir güvenlik katmanı sağlar.

Tek kullanımlık şifreler (OTP), kart sahibinin kimliğini doğrulamak için yaygın olarak kullanılan bir başka yöntemdir. OTP'ler genellikle SMS veya uygulama aracılığıyla kullanıcıya gönderilen bir şifredir. Bu şifre, kullanıcının bir işlemi onaylamak veya kimliğini doğrulamak için kullanılır (Jin, Gao, & Liu, 2021).

Son olarak, kart doğrulama kodu (CVC) gibi özel güvenlik kodları da kart sahibinin kimliğini doğrulamak için kullanılır. Bu kodlar, kart sahibinin kartın fiziksel kopyasına erişimi olup olmadığını belirlemeye yardımcı olur.

2.2.7 Olay Yönetimi ve Müdahale

Olay yönetimi ve müdahale, kredi kartı dolandırıcılığıyla mücadelede kritik bir rol oynar. Bu süreç, şüpheli işlemlerin tespit edilmesi, incelenmesi ve gerektiğinde müdahale edilmesini içerir. Öncelikle, sisteme giren herhangi bir anormal işlem veya aktivite tespit edilir. Ardından, bu işlemler detaylı bir şekilde incelenir ve doğrulanır. İncelenen işlemler hakkında ilgili taraflara uyarılar ve bildirimler gönderilir. Risk değerlendirmesi yapılır ve uygun kararlar alınır. Şüpheli işlemler doğrulandıktan ve risk değerlendirmesi yapıldıktan sonra, gerekirse müdahale edilir ve işlem engellenir. Bu süreç hem işletmenin hem de müşterilerin güvenliğini sağlamak için önemlidir (Shen & Hsiao, 2021).

2.2.8 Teknolojik İlerleme

Teknolojinin sürekli olarak gelişmesiyle birlikte, finansal alanda dolandırıcılıkla mücadele stratejileri de sürekli olarak uyum sağlamaktadır. Yeni dolandırıcılık yöntemlerinin ortaya çıkması, finansal kurumları ve ödeme sistemlerini bu yöntemlere karşı önlem almaya ve tespit mekanizmalarını güçlendirmeye zorlamaktadır. Bu sebeple, teknolojik gelişmelerin yakından takip edilerek finansal sistemlere entegre edilmesi, dolandırıcılık tespiti ve önleme stratejilerinin daha etkili hale gelmesini sağlar.

Yapay zekâ (AI), makine öğrenimi (ML), büyük veri analitiği ve blok zinciri gibi yeni teknolojiler, finansal kurumlar ve ödeme sistemlerinin dolandırıcılık tespiti ve önleme yöntemlerine entegre edilerek, dolandırıcılığın tespit edilmesini ve engellenmesini hızlandırır. Bu teknolojiler ayrıca, müşteri verilerinin korunmasını sağlar ve finansal hizmetlerin güvenliğini ve bütünlüğünü destekler.

2.2.9 Yapay Zekâ ve Makine Öğrenmesi

Yapay Zekâ ve Makine Öğrenmesi, büyük veri setlerini analiz ederek dolandırıcılık desenlerini tanımlamak ve tespit etmek için kullanılır. Makine öğrenmesi ve derin öğrenme teknikleri, yapay zekânın dolandırıcılık tespitinde etkinliğini artırır. Yapay zekâ algoritmaları, sürekli olarak yeni dolandırıcılık yöntemlerini öğrenerek kendini günceller ve daha akıllı hale gelir. Bu sayede, finansal kurumlar ve ödeme sistemleri, yapay zekâ ve makine öğrenmesi kullanarak kredi kartı dolandırıcılığını daha etkili bir şekilde tespit edebilir ve önleyebilir (Ahmad & Saini, 2019).

Yapay Zekâ ve Makine Öğrenmesi, sahtekarlık işaretlerini belirleme ve gerçek zamanlı olarak müdahale etme yeteneğiyle finansal güvenliği artırır. Örneğin, bu teknolojiler, alışılmadık veya anormal işlem desenlerini tanımlayabilir ve şüpheli işlemleri otomatik olarak belirleyerek hızlı bir şekilde müdahale edebilir. Ayrıca, yapay zekâ ve makine öğrenmesi, sahtekarlıkla mücadelede insan hatalarını azaltabilir ve manuel işlemleri otomatikleştirerek zaman ve maliyet tasarrufu sağlayabilir. Bu nedenle, finansal kurumlar ve ödeme sistemleri, yapay zekâ ve makine öğrenmesini dolandırıcılık tespiti ve önleme stratejilerinin merkezine yerleştirerek daha güvenli bir ortam oluşturabilirler.

Bölüm 3

Kredi Kartı Dolandırıcılığı Tespitinde Yapay Zekâ: Kullanılan Yöntemler ve Literatür Taraması

Kredi kartı dolandırıcılığının tespitinde yapay zekâ, son yıllarda önemli bir araç haline gelmiştir. Yapay zekâ teknikleri, büyük veri analitiği, makine öğrenmesi ve derin öğrenme gibi yöntemlerle dolandırıcılık desenlerini tanımlama, şüpheli işlemleri belirleme ve dolandırıcılığı önleme amacıyla kullanılmaktadır.

Literatürde yapılan araştırmalar, yapay zekâ yöntemlerinin kredi kartı dolandırıcılığının tespitinde başarılı olduğunu göstermektedir. Makine öğrenmesi algoritmaları, büyük veri setlerini analiz ederek sahtekarlık desenlerini tanımlama ve tespit etme yeteneğine sahiptir. Derin öğrenme teknikleri, karmaşık sahtekarlık desenlerini belirleme ve gerçek zamanlı olarak müdahale etme imkânı sağlar. Bu yöntemler, kredi kartı dolandırıcılığına karşı daha hızlı ve etkili bir koruma sağlayabilir.

Literatür taramaları yapay zekâ tekniklerinin kredi kartı dolandırıcılığı tespitinde kullanımının arttığını ve yeni yöntemlerin sürekli olarak geliştirildiğini göstermektedir. Özellikle, derin öğrenme ve yapay sinir ağları gibi gelişmiş tekniklerin kullanımı ile, sahtekarlık desenlerinin daha hassas bir şekilde tanımlanması ve önlenmesi amaçlanmaktadır.

3.1 Yapay Zekâ ile Kredi Kartı Dolandırıcılığı Tespiti İçin Kullanılan Yöntemler

Yapay zekâ, kredi kartı dolandırıcılığı gibi finansal suçları analiz etmek, alışılmadık desenleri tanımlamak ve anormal işlemleri saptamak için kullanılan önemli bir araçtır. Bu alandaki çalışmalar, karmaşık algoritmalar ve teknikler kullanarak dolandırıcılığı etkili bir şekilde tespit etmek ve önlemek için yapay zekanın potansiyelini araştırmaktadır. Bu bölümde, yapay zekâ ile kredi kartı dolandırıcılığı tespitinde kullanılan yöntemler ve bu teknolojilerin finansal güvenlik alanındaki rolü üzerine daha detaylı bir değerlendirme sunulmaktadır.

3.1.1 Makine öğrenmesi algoritmaları

Makine öğrenmesi, bilgisayar sistemlerinin verilerden öğrenme yeteneğine sahip olmasını sağlayan bir alanıdır. Bu alanda, algoritmalar veri analizi yapar, desenler bulur ve gelecekteki olayları tahmin edebilir. Makine öğrenmesi, kredi kartı dolandırıcılığını tespit etmek için etkili bir araç haline gelmiştir. Bu algoritmalar genellikle büyük miktarda veriyi işler ve karmaşık ilişkileri ortaya çıkarır (Ahmad & Saini, 2019).

Makine öğrenmesi algoritmaları, çeşitli kategorilere ayrılabilir ve farklı türde veri analizi problemleri için kullanılabilir. Örneğin, sınıflandırma algoritmaları, veriyi belirli kategorilere ayırırken kullanılırken, kümeleme algoritmaları benzer özelliklere sahip veri noktalarını gruplandırır. Ayrıca, regresyon algoritmaları, veri arasındaki ilişkileri modellemek için kullanılır. Bu alanda kullanılan bazı algoritmalar aşağıda yer almaktadır:

- Destek Vektör Makineleri (SVM)
- Rastgele Orman (RF)
- Naive Bayes (NB)
- Lojistik Regresyon (LR)
- K-En Yakın Komşu (KNN)
- Karar Ağaçları (DT)
- Yapay Sinir Ağları (ANN)

3.1.2 Derin öğrenme algoritmaları

Derin öğrenme algoritmaları, karmaşık yapıdaki verileri analiz etmek ve özelliklerini çıkarmak için çok katmanlı yapay sinir ağlarını kullanır. Bu algoritmalar genellikle büyük veri setlerinde yüksek performans sergiler ve kredi kartı dolandırıcılığını tespit etmek gibi karmaşık görevlerde etkili olmaktadır.

Derin öğrenme algoritmaları, birçok farklı mimariye sahip olabilir. Bu algoritmalar, girdi verilerindeki karmaşıklığı tanımlamak ve önemli özellikleri vurgulamak için kullanılır. Bu alanda kullanılan bazı algoritmalar aşağıda yer almaktadır:

- Evrişimli Sinir Ağları
- Uzun-Kısa Süreli Bellekli Ağlar
- Derin Otomatik Kodlayıcılar
- Derin Yoğun Ağlar
- Çok Katmanlı Algılayıcılar

3.1.3 Kümeleme ve anomali tespiti

Kümeleme ve anomali tespiti, kredi kartı dolandırıcılığını tespit etmek için önemli bir yöntemdir. Kümeleme algoritmaları, benzer özelliklere sahip işlemleri gruplandırarak dolandırıcılık desenlerini tespit etmeye yardımcı olmaktadır. Anomali tespiti ise normal davranıştan sapmaları belirleyerek şüpheli işlemleri tanımlamayı amaçlar. Bu doğrultuda kullanılan bazı algoritmalar aşağıda gösterilmiştir:

- Hiyerarşik Kümeleme
- Ortalama Kaydırma Algoritması
- Gauss Karışım Modelleri
- K-Means Kümeleme Yöntemi
- İzolasyon Ormanı
- Yerel Aykırı Değer Faktörü

3.1.4 Graf tabanlı ve ağ analizi yöntemleri

Graf tabanlı ve ağ analizi yöntemleri, karmaşık ağ yapıları arasındaki ilişkileri ve desenleri ortaya çıkarmak için kullanılan bir dizi analitik tekniktir.

Bu yöntemler genellikle dolandırıcılık tespiti için kullanılan verilerin yapısal özelliklerini keşfetmek ve anlamak amacıyla kullanılır. Bu teknikler, dolandırıcılık şemalarını ve suç örgütlerini tespit etmek, dolandırıcılık aktivitelerini izlemek ve etkili bir şekilde önlemek için değerli bir araçtır. Bu doğrultuda kullanılan bazı algoritmalar aşağıda gösterilmiştir:

- Sosyal Ağ Analizi
- Hesap Ağı Analizi
- Ağ Yapısı Analizi
- Kümeleme Analizi
- Ağ Akış Analizi

3.1.5 Zaman serisi analizi ve tahmine dayalı yöntemler

Zaman serisi analizi ve tahmine dayalı yöntemler, geçmiş zamanlarda gözlemlenen verilerin düzenli aralıklarla kaydedildiği zaman serilerini inceleyerek gelecekteki değerleri tahmin etmek için kullanılır. Bu yöntemler, zaman içindeki trendleri, mevsimsellikleri ve dönemsel desenleri belirlemeye yardımcı olur. Örneğin, finansal piyasalardaki hisse senedi fiyatlarının gelecekteki performansını tahmin etmek için zaman serisi analizi ve tahmine dayalı yöntemler kullanılabilir. Bu teknikler genellikle regresyon analizi ve hareketli ortalama yöntemleri gibi istatistiksel ve matematiksel teknikleri içerir. Bunlar, geçmiş veri trendlerini ve desenlerini modellemek ve gelecekteki değerleri tahmin etmek için kullanılır. Bu alanda kullanılan bazı algoritmalar aşağıda gösterilmiştir:

- Hareketli Ortalama Yöntemleri
- Uzun Kısa Dönem Bellek ve Tekrarlayan Sinir Ağları
- Mevsimsel Otomatik Regresyon Entegre Hareketli Ortalama
- Eksponansiyel Düzleştirme Yöntemleri

3.1.6 Özellik seçimi ve özellik mühendisliği

Özellik seçimi ve özellik mühendisliği, kredi kartı dolandırıcılığı tespitinde kritik bir rol oynamaktadır. Özellik seçimi, veri kümesindeki en önemli özelliklerin belirlenmesi ve gereksiz veya zararlı özelliklerin çıkarılması işlemidir.

Özellik mühendisliği ise, mevcut özelliklerden yeni ve daha anlamlı özellikler oluşturmayı içerir. Yapay zekâ ile yapılan çalışmalarda veri setindeki özelliklerin kalitesi çok önemli olduğundan özellik seçimi ve özellik mühendisliği çok kritik bir öneme sahiptir. Bu alanda kullanılan bazı algoritmalar aşağıda gösterilmiştir:

- Ana Bileşenler Analizi
- Yinelemeli Özellik Elemesi
- Varyans Eşiği Belirleme
- En İyi K Özelliği Seçme
- Lasso Regresyonu
- Ağaç Temelli Özellik Seçimi
- Sıralı Özellik Seçimi

3.2 Yapay Zekâ ile Kredi Kartı Dolandırıcılığı Tespiti İçin Literatür Taraması

Yapay Zekâ ile kredi kartı dolandırıcılığı tespiti için literatür taraması yapılmıştır. Bu doğrultuda çeşitli bilimsel veri tabanları ve arama motorları kullanılarak kapsamlı bir araştırma gerçekleştirilmiştir. Google Scholar ve Semantic Scholar gibi elektronik arama motorları, yapay zekâ ve kredi kartı dolandırıcılığıyla ilgili anahtar kelimelerin farklı kombinasyonlarıyla taranmıştır. Bu kelimeler arasında 'kredi kartı dolandırıcılığı', 'dolandırıcılık tespiti', 'yapay zekâ algoritmaları', 'makine öğrenimi' ve 'derin öğrenme' gibi terimler bulunmaktadır.

Arama stratejisi, ilgili literatürde en güncel ve önemli çalışmaları belirlemek için kullanılmıştır. Elde edilen sonuçlar, çeşitli araştırma makaleleri, konferans bildirileri ve tezlerden oluşan geniş bir veri setini içermektedir. Bu veri seti, yapay zekâ ile kredi kartı dolandırıcılığı tespiti alanındaki en son gelişmeleri ve eğilimleri anlamak için analiz edilmiştir.

Aşağıda, yapılan literatür taraması sonucunda elde edilen bilgiyi içeren tablo yer almaktadır. Tablo, kullanılan algoritmaları, kullanım sıklıklarını ve literatürdeki çalışma sayısını göstermektedir.

Algoritma	Kullanım Sıklığı	Çalışma Sayısı
Lojistik Regresyon	Çok Yüksek	120
Karar Ağaçları	Çok Yüksek	106
Yapay Sinir Ağları	Çok Yüksek	95
Destek Vektör Makineleri	Yüksek	82
Rastgele Orman	Yüksek	68
Naif Bayes	Yüksek	59
K-En Yakın Komşu	Yüksek	54
Tek Sınıf Destek Vektör Makineleri	Orta	41
İzolasyon Ormanı	Orta	32
Hiyerarşik Kümeleme	Orta	26
Evrişimsel Sinir Ağları	Orta	21
Izgara Arama	Orta	16
Yerel Aykırı Gözlem Faktörü	Orta	12
Eğim Artırma Makineleri	Orta	11
Genetik Algoritmalar	Orta	11
Dikkat Tabanlı Uzun Kısa Dönem Bellek	Orta	10
Tekrarlayan Sinir Ağları	Düşük	7
Çok Katmanlı Algılayıcılar	Düşük	6
Uzun Kısa Dönem Bellek	Düşük	5
Bayes Eniyileme	Düşük	5
Oto Kodlayıcılar	Düşük	4
Aşırı Gradyan Artırma	Düşük	4

Temel Bileşen Analiz	Düşük	4
Uyarlamalı Artırma	Düşük	3
Değişkenlik Tabanlı Oto Kodlayıcılar	Düşük	3
Üretici Karşıt Ağlar	Düşük	3
Rastgele Arama	Düşük	3
Yinelemeli Özellik Elemesi	Düşük	3
Parçacık Sürü Optimizasyonu	Düşük	3
Genel Vektörler	Düşük	2
Kategorik Eğim Artırma Makineleri	Düşük	2
Kelime Gömme ve Belge Gömme	Düşük	2
Özellik Önemi	Düşük	2
Doğal Dil İşleme Modeli	Düşük	2
Zaman Serisi Analizi Modeli	Düşük	1
Görüntü Ağı	Düşük	1
En Küçük Mutlak Büzülme ve Seçim Operatörü	Düşük	1
Uzun Kısa Süreli Bellek ve Tekrarlayan Sinir Ağları	Düşük	1

Tablo 3.1: Kredi Kartı Dolandırıcılığının Tespiti İçin Kullanılan Algoritmalar

Bölüm 4

Kullanılan Veri Seti, Algoritmalar ve Performans Ölçütleri

4.1 Veri Seti

Bu çalışmada kullandığımız veri seti, Brüksel Libre Üniversitesi Makine Öğrenimi Grubu'nun dolandırıcılık tespiti konusundaki bir araştırması için, kullanılacak modellerin geliştirilmesi ve değerlendirilmesi amacıyla toplanmıştır.

Veri seti Avrupalı kart sahipleri tarafından Eylül 2013'te gerçek bir finansal kuruluşun kredi kartı işlem verilerden oluşmaktadır. Veri setine Kaggle platformundan kamuya açık şekilde erişilebilmektedir (Kaggle, 2022).

Veri setinde 284.807 işlemden oluşmaktadır, bu işlemlerden sadece 492'si dolandırıcılık yapılan işlemdir. Pozitif sınıf (dolandırıcılık) tüm işlemlerin %0.172'sini oluşturmaktadır. Aşağıdaki tablo veri setindeki işlemlerin özetini göstermektedir;

İşlem	**Sayı**
Normal	284.315
Dolandırıcılık	492
Toplam	284.807

Tablo 4.1: Veri Setindeki İşlem Sayıları

Veri seti içerisinde zaman, işlem tutarı ve işlem sınıfı gibi toplam 30 adet değişken bulunmaktadır. Bu değişkenlerin adı ve açıklaması gibi detaylar aşağıdaki tabloda gösterilmektedir;

Değişken	Veri Tipi	Açıklama
Time (Zaman)	int	İşlemler arasındaki süre
Amount (Tutar)	num	İşlem tutarı
Class (Sınıf)	int	Yanıt değişkeni, dolandırıcılık 1, normal 0
V1 - V28	num	İşlem için anonimleştirilmiş özellikler

Tablo 4.2: Veri Setinin Yapısı

Veri seti herhangi bir hatalı değer içermemektedir. Bu nedenle temizleme veya ön işleme gibi bir çalışmaya ihtiyaç duymamaktadır. Ancak dolandırıcılık işlem sayısının toplam veri içerisindeki oranının sadece %0.172 olması nedeniyle dengesizlik durumu söz konusudur. Bu nedenle makine öğrenmesi gibi durumlarda performans ölçütlerinde bazı istenmeyen sonuçlar elde edilmektedir. Bu çalışmada dengesizlik durumundan kurtulmak için sentetik veri üretilmesini sağlayan SMOTE yöntemi kullanılmıştır.

4.2 Algoritmalar

Kredi kartı dolandırıcılığını tespit etmek için yapılan literatür taraması, sıkça tercih edilen makine öğrenmesi algoritmalarını gün yüzüne çıkarmıştır. Bu algoritmalar, dolandırıcılık olaylarını etkili bir şekilde tanımlamak ve önlemek için titizlikle seçilmiştir. Çalışmamızda, bu algoritmaların detaylı incelemesi ve performans değerlendirmesi gerçekleştirilecektir. Aşağıda seçilen algoritmalar listelenmektedir:

- Destek Vektör Makineleri (SVM)
- Rastgele Orman (RF)
- Naive Bayes (NB)
- Lojistik Regresyon (LR)
- K-En Yakın Komşu (KNN)
- Karar Ağaçları (DT)

4.2.1 Destek Vektör Makineleri Algoritması

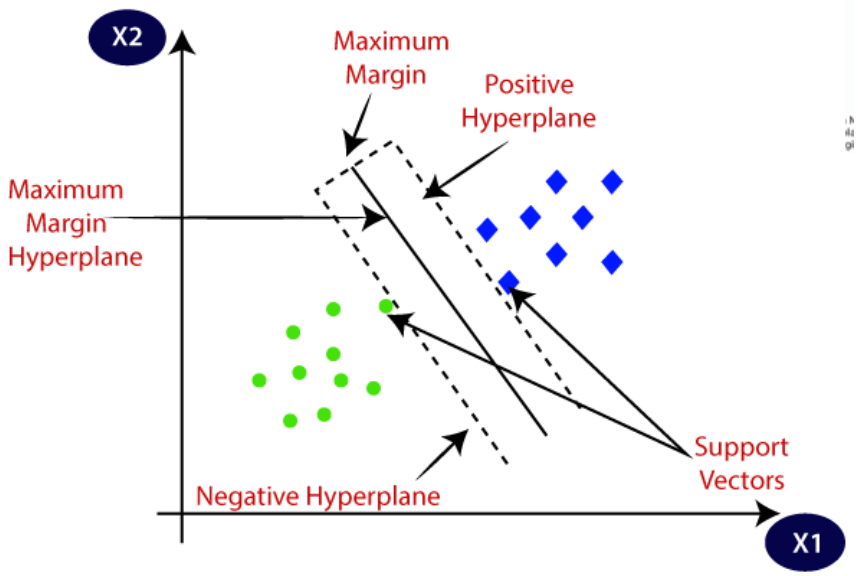
Destek Vektör Makineleri (SVM), sınıflandırma ve regresyon problemleri için kullanılan güçlü bir makine öğrenimi yöntemidir. Özellikle ikili sınıflandırma problemlerinde, SVM algoritması iki sınıf arasındaki en geniş marjı bulmaya çalışır ve bu marjın sınırlarındaki veri noktalarına "destek vektörleri" denir. Kredi kartı dolandırıcılığı tespiti gibi ikili sınıflandırma problemlerinde SVM'nin uygulanabilir bir yöntem olduğu bilinmektedir (Bhattacharyya, 2011).

SVM'nin gücü, doğrusal olarak ayrılabilir olmayan problemler için de başarılı bir şekilde kullanılabilmesinden gelmektedir. Bu durumda, SVM özellik alanını dönüştüren çekirdek yöntemini kullanarak veri noktalarını daha yüksek boyutlu bir alana eşlemek için bir çekirdek fonksiyonu kullanır. (Aaron Hertzmann, 2015).

SVM'nin uygulanmasında dikkat edilmesi gereken ön işleme adımları bulunmaktadır. Örneğin, eksik verilerin doldurulması, kategorik değişkenlerin kodlanması ve özelliklerin ölçeklendirilmesi gibi adımlar gerçekleştirilmelidir. Veri dengesizliği problemiyle başa çıkmak için ise Sentetik Azınlık Yüksek Örnekleme Tekniği (SMOTE) gibi veri artırma yöntemleri kullanılabilir (Chawla, Bowyer, Hall, & Kegelmeyer, 2002).

Model performansını artırmak için önemli özelliklerin seçilmesi gereklidir. Bu amaçla, LASSO, RFE veya istatistiksel testler gibi çeşitli özellik seçimi yöntemleri kullanılabilir (Demsar, 2006).

Ön işleme ve özellik seçimi tamamlandıktan sonra, veri kümesi eğitim ve test setlerine ayrılır. SVM modeli eğitim seti üzerinde eğitilir ve test seti üzerinde değerlendirilir. Modelin performansını ölçmek için çeşitli ölçütler kullanılabilir, örneğin, doğruluk, kesinlik, duyarlılık ve F1 skoru. SVM modelinin hiperparametreleri optimize edilerek modelin performansı artırılabilir. Bu adımların doğru bir şekilde uygulanması, SVM'nin kredi kartı dolandırıcılığı gibi finansal dolandırıcılık tespiti gibi önemli uygulamalarda başarılı bir şekilde kullanılmasını sağlar.



Şekil 4.1: Destek Vektör Makineleri Algoritmasının Grafiği

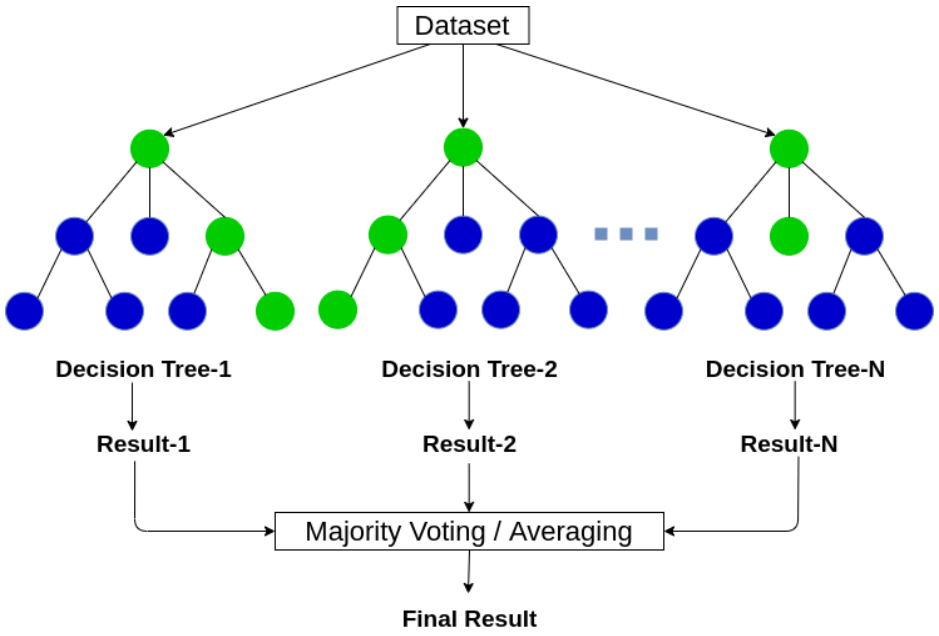
4.2.2 Rastgele Orman Algoritması

Rastgele Orman (RF), karar ağaçlarının birleşiminden oluşan bir topluluk öğrenme yöntemidir ve genellikle sınıflandırma ve regresyon problemleri için kullanılır. Rastgele Orman, her bir karar ağacının bağımsız olarak bir alt veri kümesi üzerinde eğitilmesi ve ardından tahminlerin birleştirilmesi yoluyla çalışır. Bu yöntem, karar ağaçlarının aşırı uydurmayı azaltmak için birbirinden bağımsız olarak eğitilmesi ve çeşitli alt veri kümeleri üzerinde çalışması nedeniyle oldukça etkili ve popülerdir (Louppe, 2015).

Rastgele Orman, her bir karar ağacının oluşturulması sırasında rastgele özellik seçimi yapar, bu da her bir ağacın birbirinden farklı olduğu anlamına gelir. Bu özellik, modelin çeşitliliğini artırır ve genelleştirme yeteneğini iyileştirir. Her ağacın tahminlerinin birleştirilmesiyle, Rastgele Orman modeli genellikle tek bir karar ağacından daha güçlü ve istikrarlı tahminler yapabilir.

Rastgele orman algoritmasının bir diğer avantajı, genellikle büyük özellik kümesi ile çalışabilmesidir. Çünkü her ağaç rastgele seçilen bir alt kümesiyle eğitilir. Ayrıca, veri kümesindeki eksik değerleri ve aykırı değerleri işleyebilme yeteneği sayesinde, genellikle veri temizliği gerektiren gerçek dünya veri setlerinde de etkilidir.

Rastgele Ormanın hiperparametreleri kolayca ayarlanabilir ve genellikle varsayılan parametrelerle bile iyi performans gösterebilir. Bununla birlikte, çapraz doğrulama gibi teknikler kullanılarak hiperparametrelerin optimize edilmesi modelin performansını daha da artırabilir.



Şekil 4.2: Rastgele Orman Algoritmasının Grafiği

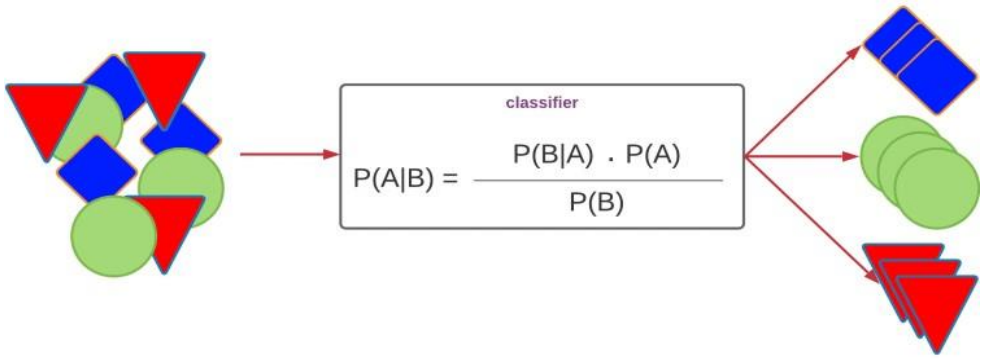
4.2.3 Naive Bayes Algoritması

Naif Bayes (NB), sınıflandırma problemleri için yaygın olarak kullanılan basit ancak güçlü bir olasılık temelli bir makine öğrenimi algoritmasıdır. Temel olarak Bayes Teoremi'ne dayanır ve bağımsızlık varsayımını yapar: tüm özellikler arasında birbirinden bağımsızdır. Bu varsayım, algoritmanın basitliğini artırır ve hesaplama karmaşıklığını azaltır, bu da büyük veri kümeleriyle etkili bir şekilde çalışmasını sağlar.

Naif Bayes, sınıflandırma problemlerindeki bir örneğin sınıfını belirlemek için Bayes Teoremi'ni kullanır. Bayes Teoremi'ne göre, bir örneğin sınıfını belirlemek için verilen özelliklerin olasılıklarını hesaplamak için sınıfın olasılığı kullanılır. Naif Bayes, her sınıfın olasılığını ve her sınıf için verilen özelliklerin olasılıklarını hesaplar ve en yüksek olasılığa sahip sınıfı tahmin eder (Uludağ, Gürsoy, 2020).

Naif Bayes, genellikle doğal dil işleme gibi metin sınıflandırma problemlerinde ve spam filtreleme gibi uygulamalarda başarılı bir şekilde kullanılır. Bu tür uygulamalarda, özellikler genellikle kelimelerin varlığı veya yokluğu gibi basit özelliklerdir ve Naif Bayes'in bağımsızlık varsayımı bu tür verilerle iyi çalışır.

Naif Bayes'in avantajları arasında basitlik, hız ve düşük bellek gereksinimleri bulunur. Bununla birlikte, bağımsızlık varsayımının gerçekte her zaman doğru olmadığı durumlarda performansı düşebilir. Eğitim veri setinde nadir veya hiç gözlemlenmemiş özellikler varsa, sıfır olasılık sorunuyla karşılaşabilir. Bu sorunla başa çıkmak için pürüzsüzleştirme teknikleri kullanılabilir (Uludağ, Gürsoy, 2020).



Şekil 4.3: Naive Bayes Algoritmasının Grafiği

4.2.4 Lojistik Regresyon Algoritması

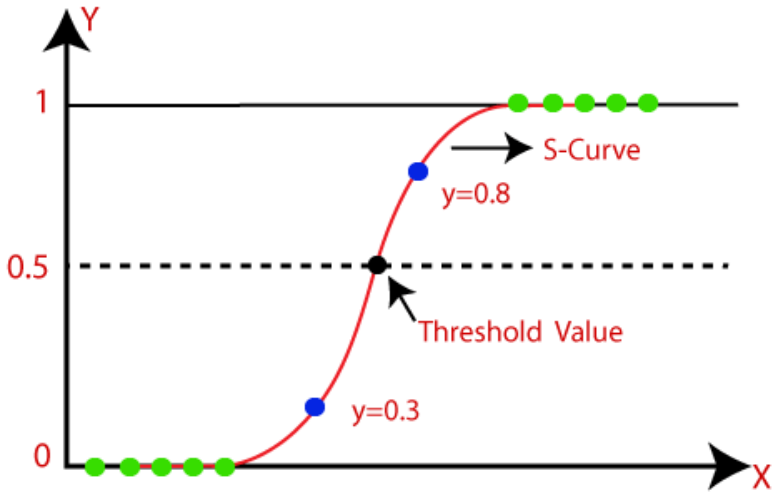
Kredi kartı dolandırıcılığının tespiti için Lojistik Regresyon, finansal kurumlar tarafından sıkça kullanılan bir yöntemdir. Bu algoritma, kredi kartı işlemlerinde dolandırıcılık olasılığını tahmin etmek için kullanılan istatistiksel bir modeldir. Lojistik Regresyon'un temel prensibi, bağımlı bir değişkenin (dolandırıcılık olasılığı gibi) bağımsız değişkenlerle (işlem tutarı, işlem zamanı, coğrafi konum gibi) ilişkisini modellemektir (Udjianto, 2006).

Lojistik regresyon algoritmasını daha iyi anlamak için örnek verebiliriz. Örneğin, bir müşterinin kredi kartıyla yapılan bir işlem yüksek bir tutara sahipse ve alışılmadık bir coğrafi konumdan gerçekleştirilmişse, bu işlemin dolandırıcılık olasılığı daha yüksek olabilir.

Lojistik Regresyon, bu tür desenleri tanıyarak ve belirli özelliklerin işlemin dolandırıcılık olasılığına etkisini değerlendirerek, her işlemin dolandırıcılık olasılığını tahmin edebilir.

Modelin eğitimi için kullanılan veri seti, geçmişteki dolandırıcılık vakalarını içerir ve normal işlemleri de içerebilir. Bu veri seti üzerinde, Lojistik Regresyon modeli eğitilir ve daha sonra gerçek zamanlı işlemleri analiz etmek için kullanılır.

Model, işlem verilerini alır, özelliklerini çıkarır ve dolandırıcılık olasılığını tahmin etmek için eğitilmiş parametreleri kullanarak bir skor üretir.



Şekil 4.4: Lojistik Regresyon Algoritmasının Grafiği

Lojistik Regresyon'un bu tür kullanımı, kredi kartı dolandırıcılığı gibi finansal suçların tespitinde önemli bir araç olabilir. Ancak, modelin doğruluğunu artırmak için sürekli olarak güncellenmesi ve geliştirilmesi gerekmektedir. Ayrıca, modelin yanlış pozitif ve yanlış negatif tahminlerinin etkileri dikkate alınmalı ve işlemlerin gerçekten dolandırıcılık içerip içermediğinin doğrulanması için insan denetimi yapılmalıdır.

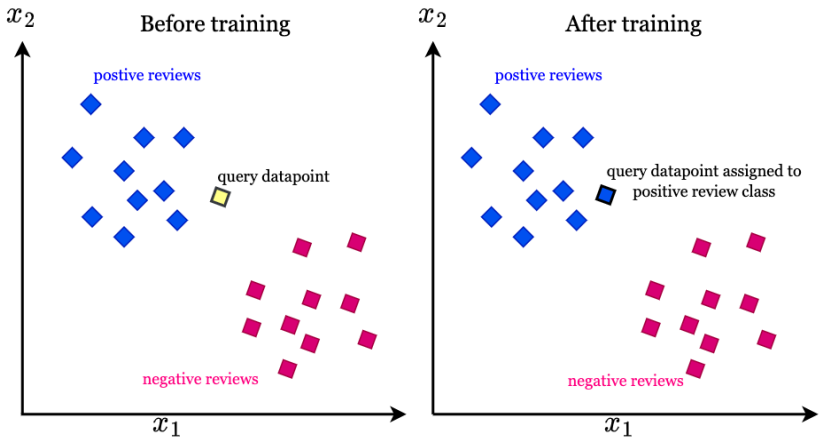
4.2.5 K-En Yakın Komşu Algoritması

K-En Yakın Komşu (KNN), kredi kartı dolandırıcılığı gibi sınıflandırma problemlerinde yaygın olarak kullanılan bir makine öğrenme algoritmasıdır. KNN, basit ve etkili bir algoritma olup, bir veri noktasını sınıflandırmak için yakınındaki komşularının sınıf etiketlerini dikkate alır.

Algoritmanın çalışma prensibi şu şekildedir: Veri noktası, özellik uzayında diğer veri noktalarına olan uzaklıkları dikkate alınarak sınıflandırılır. Bu uzaklıklar genellikle Öklid mesafesi veya Manhattan mesafesi gibi ölçümlerle hesaplanır. Daha sonra, veri noktasının komşularının sınıf etiketleri ve belirlenen bir K değeri (komşu sayısı) kullanılarak veri noktasının sınıfı tahmin edilir (Uğuz, Oral, 2019).

KNN'nin avantajlarından biri, basitliği ve doğruluğuyla bilinmesidir. Ayrıca, eğitim aşamasında karmaşık bir model oluşturulmasına gerek olmadığı için hızlı bir şekilde uygulanabilir. Ancak, KNN'nin dezavantajları da vardır. Özellikle, büyük veri kümelerinde hesaplama maliyeti artabilir ve veri setindeki gürültüye ve özellikler arasındaki ilişkisizliğe duyarlı olabilir (Zhang, 2010).

KNN'nin kredi kartı dolandırıcılığı tespiti gibi finansal suçlarla ilgili problemlerde kullanılması, özellikle veri setinin büyük olmadığı durumlarda etkili olabilir. Ancak, KNN'nin performansını artırmak için uygun bir K değeri seçilmesi ve veri ön işleme adımlarının doğru bir şekilde uygulanması önemlidir.



Şekil 4.5: K-En Yakın Komşu Algoritmasının Grafiği

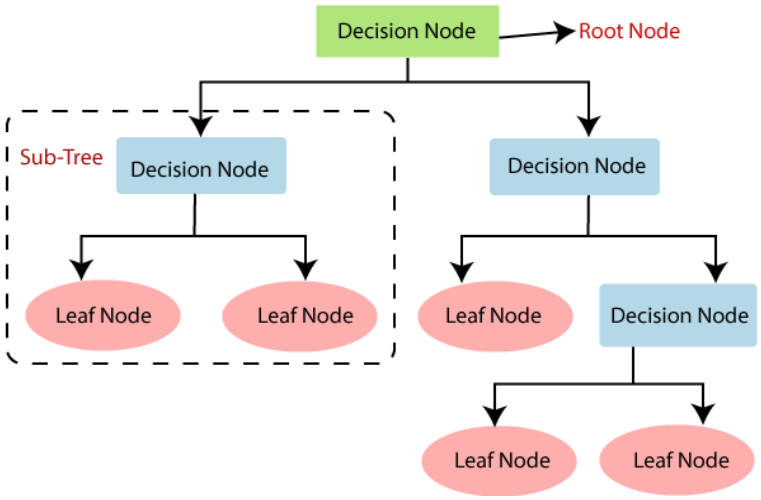
4.2.6 Karar Ağaçları Algoritması

Karar Ağaçları, kredi kartı dolandırıcılığı gibi sınıflandırma problemlerinde sıkça kullanılan bir makine öğrenme algoritmasıdır. Diğer denetimli öğrenme algoritmalarının aksine hem regresyon hem de sınıflandırma sorunlarının çözümü için kullanılmaktadır (Sahin., Duman, 2011).

Bu algoritma, veri setindeki özelliklerin değerlerine göre karar ağaçları oluşturarak sınıflandırma yapar. Karar ağaçları, adım adım bir dizi karar kuralını uygulayarak bir veri noktasını bir sınıfa atar. Veri setindeki özellikler, karar ağacının düğümlerine ve dallarına göre ayrılır. Her düğüm, bir özelliğin değerini test eder ve bu değere göre dallara ayırır. Bu işlem, belirlenmiş bir kriter fonksiyonu kullanılarak gerçekleştirilir.

Karar ağaçlarının avantajları arasında yüksek yorumlanabilirlik, kolaylıkla görselleştirilebilirlik ve farklı tipteki veri setlerine uyum sağlama yeteneği bulunur. Ayrıca, karar ağaçları veri ön işleme gereksinimini azaltabilir ve kategorik verilerle iyi çalışabilir. Karar ağaçlarının bazı dezavantajları da vardır. Özellikle, aşırı uyuma (overfitting) eğilimli olabilirler, yani eğitim veri setine aşırı derecede uyum sağlayarak genelleme yeteneklerini kaybedebilirler. (Yıldız, 2020).

Kredi kartı dolandırıcılığı tespiti gibi problemlerde karar ağaçları, özellikle belirli özelliklerin ve karar kriterlerinin anlaşılabilirliği önemli olduğunda tercih edilebilir. Ancak, modelin aşırı uyuma eğilimini kontrol etmek için uygun düğüm derinliği ve düğüm bölme kriteri gibi hiperparametrelerin dikkatlice ayarlanması gereklidir.



Şekil 4.6: Karar Ağaçları Algoritmasının Grafiği

4.3 Performans Ölçütleri

Performans ölçütleri, makine öğrenimi modellerinin etkinliğini değerlendirmek ve karşılaştırmak için kullanılan önemli ölçütlerdir. Bu ölçütler, bir modelin tahmin yeteneğini, doğruluğunu ve güvenilirliğini değerlendirirken önemli bir rol oynar. Bu ölçütlerin anlaşılması ve doğru şekilde yorumlanması, makine öğrenimi modellerinin geliştirilmesi ve optimize edilmesi açısından hayati öneme sahiptir. Aşağıdaki listede bu çalışmada kullandığımız performans ölçütleri gösterilmektedir;

- Doğruluk
- Kesinlik
- Duyarlılık
- F1 Skoru

4.3.1 Doğruluk

Doğruluk ölçütü, bir tahminin veya modelin ne kadar doğru olduğunu değerlendirmek için kullanılan bir ölçüttür. Genellikle makine öğrenimi ve istatistiksel analiz gibi alanlarda kullanılır. Doğruluk ölçütü, tahminlerin gerçek değerlerle ne kadar iyi eşleştiğini ifade eder (Powers, 2011).

Doğruluk ölçütünün formülü, doğru tahmin edilen örnek sayısının toplam örnek sayısına bölünmesiyle elde edilir. Bu formül genellikle yüzde olarak ifade edilir. Aşağıdaki formülü kullanarak doğruluk ölçütünü hesaplayabilirsiniz:

$$Doğruluk = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

- TP (True Positive), gerçek pozitifleri temsil eder. Yani, pozitif olarak tahmin edilen örneklerin gerçekten pozitif olduğu durumları ifade eder.
- TN (True Negative), gerçek negatifleri temsil eder. Yani, negatif olarak tahmin edilen örneklerin gerçekten negatif olduğu durumları ifade eder.
- FP (False Positive), yanlış pozitifleri temsil eder. Yani, negatif olan örneklerin pozitif olarak yanlış bir şekilde tahmin edildiği durumları ifade eder.
- FN (False Negative), yanlış negatifleri temsil eder. Yani, pozitif olan örneklerin negatif olarak yanlış bir şekilde tahmin edildiği durumları ifade eder.

Doğruluk ölçütü, doğru sınıflandırılan örneklerin toplam örnek sayısına oranını verir. Yani, doğru pozitif ve doğru negatif sınıflandırmaların toplam sayısını, tüm örneklerin toplam sayısına böler.

Doğruluk ölçütü, modelin performansını anlamak için kullanışlı bir ölçüttür. Ancak bazı durumlarda doğruluk tek başına yeterli olmayabilir. Özellikle dengesiz sınıf dağılımları veya yanlış sınıflandırma maliyetleri gibi durumlarda, başka ölçütler de dikkate alınmalıdır. Bu durumda, Duyarlılık, Kesinlik, F1 puanı gibi diğer performans ölçütleri de kullanılabilir.

4.3.2 Kesinlik

Kesinlik, bir sınıflandırma modelinin gerçek negatifleri doğru bir şekilde tanımlama yeteneğini ölçen bir performans metriğidir. Kesinlik, yanlış pozitifleri (FP) en aza indirgemeye odaklanır (Powers, 2011).

$$Kesinlik = \frac{TN}{(TN + FP)}$$

- TN (True Negative), gerçek negatifleri temsil eder. Yani, negatif olarak tahmin edilen örneklerin gerçekten negatif olduğu durumları ifade eder.

- FP (False Positive), yanlış pozitifleri temsil eder. Yani, negatif olan örneklerin pozitif olarak yanlış bir şekilde tahmin edildiği durumları ifade eder.

Kesinlik, modelin gerçek negatifleri doğru bir şekilde sınıflandırma yeteneğini gösterir. Yüksek bir Kesinlik değeri, modelin negatif sınıfı doğru bir şekilde tanımlama yeteneğinin yüksek olduğunu gösterir. Düşük bir Kesinlik değeri ise yanlış pozitiflerin (FP) daha fazla olduğunu ve negatif sınıfın yanlış bir şekilde pozitif olarak sınıflandırıldığını gösterebilir.

4.3.3 Duyarlılık

Duyarlılık, bir sınıflandırma modelinin pozitif olarak tahmin ettiği örneklerin gerçekten pozitif olma oranını ölçen bir performans metriğidir. Duyarlılık, yanlış pozitifleri (FP) en aza indirgemeye odaklanır (Powers, 2011).

$$Duyarlılık = \frac{TP}{(TP + EP)}$$

- TP (True Positive), gerçek pozitifleri temsil eder. Yani, pozitif olarak tahmin edilen örneklerin gerçekten pozitif olduğu durumları ifade eder.
- FP (False Positive), yanlış pozitifleri temsil eder. Yani, negatif olan örneklerin pozitif olarak yanlış bir şekilde tahmin edildiği durumları ifade eder.

Duyarlılık, modelin pozitif tahminlerinin doğruluğunu gösterir. Yüksek bir Duyarlılık değeri, modelin pozitif tahminlerinin çoğunlukla doğru olduğunu gösterir. Düşük bir Duyarlılık değeri ise yanlış pozitiflerin (FP) daha fazla olduğunu ve pozitif tahminlerin gerçek pozitiflere kıyasla daha az doğru olduğunu gösterebilir.

4.3.4 F1 Skoru

F1 skoru, bir sınıflandırma modelinin doğruluğunu ölçmek için kullanılan bir ölçüttür ve hem kesinlik hem de duyarlılık ölçütlerinin birleşimidir. F1 skoru, sınıflandırma modelinin hem yanlış pozitiflerle (FP) hem de yanlış negatiflerle (FN) başa çıkma yeteneğini dengeler (Powers, 2011).

$$F1 = \ 2x\frac{Kesinlik\ x\ Duyarlılık}{Kesinlik\ +\ Duyarlılık}$$

Burada precision, doğru pozitiflerin (TP) toplam pozitif tahminlerin oranını, recall ise doğru pozitiflerin toplam gerçek pozitiflerin oranını temsil eder. F1 skoru hem precision hem de recall yüksek olduğunda en yüksek değere ulaşır ve bu da modelin sınıflandırma performansının dengeli olduğunu gösterir.

F1 skoru, dengesiz sınıf dağılımlarında ve yanlış sınıflandırma maliyetlerinin olduğu durumlarda doğruluk ölçütünden daha güvenilir bir performans metriği olarak kabul edilir.

Bölüm 5

Seçilen Algoritmaların Uygulanması

Makine öğrenmesi algoritmaları, kredi kartı dolandırıcılığının tespitinde önemli bir rol oynamaktadır. Bu başlık altında, kredi kartı dolandırıcılığının tespiti için belirlenen algoritmaların pratik uygulamaları detaylı bir şekilde ele alınacaktır. Her bir algoritma için, veri hazırlığı, model eğitimi, hiperparametre ayarlaması ve performans değerlendirmesi gibi adımlar incelenecek ve finansal güvenliğin sağlanması ve dolandırıcılık vakalarının sınırlanması amacıyla nasıl kullanılabilecekleri üzerine odaklanılacaktır.

5.1 Destek Vektör Makineleri

Destek Vektör Makineleri (SVM) algoritması, yüksek boyutlu ve karmaşık veri setlerinde etkili sınıflandırma görevlerini gerçekleştirebilme yeteneğiyle öne çıkar. SVM, veri noktalarını bir kenara yerleştirerek iki sınıf arasında optimal bir hiper düzlem bulmaya çalışır. Bu şekilde, veri noktalarını sınıflara ayırmak için bir karar sınırı belirlenir ve sınıflar arasındaki maksimum marj elde edilmeye çalışılır. Bu özellikleriyle SVM, doğrusal ve doğrusal olmayan sınıflandırma problemlerinde başarılı sonuçlar elde etmektedir (Bhattacharyya, 2011).

SVM modelini uygulamadan önce, veri setinin doğru bir şekilde hazırlanması kritik öneme sahiptir. Bu adımda, veri seti özellikler ve etiketler olarak iki bölüme ayrılır. Özellikler, dolandırıcılık tespiti için kullanılacak olan veri noktalarının özelliklerini temsil ederken, etiketler bu veri noktalarının doğru sınıfını belirtir. Veri seti üzerinde yapılan bu ayrım, makine öğrenimi modelinin doğru şekilde eğitilmesini sağlar.

```python
import pandas as pd
import matplotlib.pyplot as plt
from imblearn.over_sampling import SMOTE
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVC

# Veri seti yüklendi.
data = pd.read_csv("data.csv")

# Özellikler ve Etiketler ayrıldı.
features = data.drop('Class', axis=1)
labels = data['Class']
```

Şekil 5.1: Destek Vektör Makineleri Özelliklere ve Etiketlere Ayırma Kodu

SVM gibi birçok makine öğrenimi modeli, özellikler arasındaki farklılıkları dengelemek için özellikleri ölçeklendirmeyi gerektirir. Özelliklerin ölçeklendirilmesi işlemi, özellik değerlerini aynı aralığa getirerek modelin daha dengeli bir şekilde öğrenmesini sağlar. Bu adım, modelin daha istikrarlı ve güvenilir sonuçlar üretmesine yardımcı olur (Bhattacharyya, 2011).

Dolandırıcılık tespiti durumunda, verilerin çok az bir kısmı gerçekten dolandırıcılık olduğundan genellikle veri setinde dengesizlik sorunu ile karşılaşılmaktadır. Dengesizlik yanlış sonuçlar elde etmeye ve performans ölçütlerinin hatalı olmasına neden olmaktadır. Azınlık sınıfını (dolandırıcılık işlemleri) artırmak için Sentetik Azınlık Yüksek Örnekleme Tekniği (SMOTE) kullanılmaktadır. Bu teknik, azınlık sınıfındaki veri noktalarını sentetik olarak üreterek veri setini dengeler (Chawla, Bowyer, Hall, & Kegelmeyer, 2002).

```python
# Özellikleri ölçeklendirme
scaler = StandardScaler()
scaled_features = scaler.fit_transform(features)

# Sınıf dengesizliği nedeniyle, azınlık sınıfını artırarak
# veri setini dengelemek için SMOTE tekniği kullanıldı.
smote = SMOTE(sampling_strategy='minority')
resampled_features, resampled_labels = smote.fit_resample(scaled_features, labels)
```

Şekil 5.2: Destek Vektör Makineleri Ölçeklendirme ve SMOTE Kodu

Veri setinin eğitim ve test verilerine ayrılması işlemi, modelin doğruluğunun değerlendirilmesi için önemlidir. Bu adımda, veri seti %80 eğitim ve %20 test verisi olarak ayrılır. Eğitim verisi, modelin parametrelerini öğrenmek için kullanılırken, test verisi ise modelin performansını değerlendirmek için ayrılır. Ayrıca, veri setinin ayrılması sırasında random_state parametresi belirlenerek, her seferinde aynı rastgele ayrımın yapılması sağlanır.

SVM modeli, Scikit-learn kütüphanesinin SVC sınıflandırıcı fonksiyonu kullanılarak eğitilir. SVC fonksiyonu parametresiz olarak kullanılarak model öğrenme sürecini gerçekleştirir. SVC fonksiyonu, varsayılan olarak C=1.0, kernel='rbf' ve class_weight=None gibi değerlerle çalışır. Bu parametreler, modelin karmaşıklığını ve sınıf dengesizliğini kontrol etmek için kullanılabilir. Eğitim sürecinde, model veri setindeki örnekleri kullanarak öğrenir ve sınıflandırma hiper düzlemini belirler.

```
# Veri kümesini eğitim ve test alt kümelere ayrıldı.
# Test kümesi boyutu %20, rastgele durum (random_state) ise 42 olarak belirlendi.
train_features, test_features, train_labels, test_labels = train_test_split(
    resampled_features, resampled_labels, test_size=0.20, random_state=42
)

# Destek Vektör Makineleri (SVM) modeli oluşturuldu.
svm_model = SVC()

# Model eğitildi.
svm_model.fit(train_features, train_labels)

# Eğitilmiş model kullanılarak test verileri için tahminler yapıldı.
predicted_labels = svm_model.predict(test_features)
```

Şekil 5.3: Destek Vektör Makineleri Veri Ayırma ve Model Eğitimi Kodu

Eğitim tamamlandıktan sonra, model test verisi üzerinde değerlendirilir ve performans ölçütleri hesaplanır. Bu ölçütler arasında doğruluk (accuracy), kesinlik (precision), duyarlılık (recall) ve F1 skoru bulunur. Bu ölçütler, modelin ne kadar iyi performans gösterdiğini belirlemek için kullanılır.

```
# Doğruluk, Kesinlik, Duyarlılık ve F1 Skoru değerleri hesaplandı.
accuracy = accuracy_score(test_labels, predicted_labels)
precision = precision_score(test_labels, predicted_labels)
recall = recall_score(test_labels, predicted_labels)
f1 = f1_score(test_labels, predicted_labels)
```

Şekil 5.4: Destek Vektör Makineleri Performans Ölçütleri Kodu

Tüm bu adımların ardından destek vektör algoritması modelinin performansı aşağıdaki tabloda gösterilmektedir;

Doğruluk	Kesinlik	Duyarlılık	F1 Skoru
0,980224398994	0,984781645850	0,975603762988	0,980171220497

Tablo 5.1: Destek Vektör Makineleri Performans Ölçütleri

5.2 Rastgele Orman

Rastgele Orman (Random Forest) algoritması, topluluk öğrenme (ensemble learning) yaklaşımını kullanan güçlü bir sınıflandırma ve regresyon algoritmasıdır. Bu algoritma, birden fazla karar ağacını kullanarak bir model oluşturur ve her bir ağacın tahminlerini bir araya getirerek daha doğru sonuçlar elde eder. Her karar ağacı, rastgele seçilen özellikler üzerinde eğitilir ve birbirinden bağımsızdır. Sonuçlar, çoğunluk oyu alınarak final tahmin belirlenir (Louppe, 2015).

Modelin eğitilmesine geçmeden önce veri setini ayarlamak gerekir. Veri seti önce özellikler ve etiketler olarak iki bölüme ayrılır. Özellikler, dolandırıcılık tespiti için kullanılacak olan veri noktalarının özelliklerini temsil ederken, etiketler bu veri noktalarının doğru sınıfını belirtir.

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
from sklearn.preprocessing import StandardScaler
from imblearn.over_sampling import SMOTE
from sklearn.ensemble import RandomForestClassifier

# Veri seti yüklendi.
data = pd.read_csv("data.csv")

# Özellikler ve Etiketler ayrıldı.
features = data.drop('Class', axis=1)
labels = data['Class']
```

Şekil 5.5: Rastgele Orman Özelliklere ve Etiketlere Ayırma Kodu

Özelliklerin ölçeklendirilmesi işlemi, özellik değerlerini aynı aralığa getirerek modelin daha dengeli bir şekilde öğrenmesini sağlar. Bu adımda, StandardScaler kullanılarak özellikler ölçeklendirilir. Ayrıca, sınıf dengesizliği nedeniyle azınlık sınıfını (dolandırıcılık işlemleri) artırmak için SMOTE tekniği kullanılır.

```
# Özellikleri ölçeklendirme
scaler = StandardScaler()
scaled_features = scaler.fit_transform(features)

# Sınıf dengesizliği nedeniyle, azınlık sınıfını artırarak
# veri setini dengelemek için SMOTE tekniği kullanıldı.
smote = SMOTE(sampling_strategy='minority')
resampled_features, resampled_labels = smote.fit_resample(scaled_features, labels)
```

Şekil 5.6: Rastgele Orman Ölçeklendirme ve SMOTE Kodu

Veri setinin eğitim ve test olarak ayrılması, makine öğrenimi modelinin güvenilir bir şekilde değerlendirilmesi için kritik öneme sahiptir. Bu aşamada, veri seti genellikle %80 eğitim ve %20 test verisi olarak ayrılır. Eğitim verisi, modelin parametrelerini öğrenmek ve içsel yapılarını ayarlamak için kullanılırken, test verisi modelin gerçek dünya verileri üzerinde ne kadar iyi performans göstereceğini değerlendirmek için ayrılır. Veri setinin bu şekilde ayrılması, modelin aşırı uyum (overfitting) veya eksik uyum (underfitting) gibi genelleme sorunlarından kaçınmasını sağlar.

Eğitim ve test verilerinin ayrılması sırasında dikkat edilmesi gereken bir diğer önemli nokta, random_state parametresinin belirlenmesidir. Bu parametre, veri setinin rastgele olarak bölünmesini kontrol eder ve her seferinde aynı ayrımın yapılmasını sağlar. Bu, sonuçların tekrarlanabilirliğini ve karşılaştırılabilirliğini sağlar.

Modelin eğitilmesi aşamasında, seçilen algoritmanın uygun parametrelerle belirlenmiş veri seti üzerinde eğitilmesi gerekmektedir. Rastgele Orman algoritması için ağaç sayısı (n_estimators) gibi parametreler belirlenmelidir. Eğitim süreci, eğitim verisi üzerinde yapılan iteratif bir optimizasyon sürecini içerir. Model, eğitim veri setindeki örnekleri kullanarak öğrenir ve veri setinin içindeki desenleri tanımak için kendini ayarlar.

```
# Veri kümesini eğitim ve test alt kümelere ayrıldı.
# Test kümesi boyutu %20, rastgele durum (random_state) ise 42 olarak belirlendi.
train_features, test_features, train_labels, test_labels = train_test_split(
    resampled_features, resampled_labels, test_size=0.20, random_state=42
)

# Rastgele Orman modeli oluşturuldu.
random_forest_model = RandomForestClassifier(n_estimators=100, random_state=42)

# Model eğitildi.
random_forest_model.fit(train_features, train_labels)

# Eğitilmiş model kullanılarak test verileri için tahminler yapıldı.
predicted_labels = random_forest_model.predict(test_features)
```

Şekil 5.7: Rastgele Orman Veri Ayırma ve Model Eğitimi Kodu

Eğitim tamamlandıktan sonra, model test verisi üzerinde değerlendirilir ve performans ölçütleri hesaplanır. Bu ölçütler, modelin ne kadar iyi performans gösterdiğini belirlemek için kullanılır. Aşağıda sırayla ölçütlerin hesaplanması ve ölçütlerin tablosu gösterilmektedir:

```
# Doğruluk, Kesinlik, Duyarlılık ve F1 Skoru değerleri hesaplandı.
accuracy = accuracy_score(test_labels, predicted_labels)
precision = precision_score(test_labels, predicted_labels)
recall = recall_score(test_labels, predicted_labels)
f1 = f1_score(test_labels, predicted_labels)
```

Şekil 5.8: Rastgele Orman Performans Ölçütleri Kodu

Doğruluk	Kesinlik	Duyarlılık	F1 Skoru
0,999903276296	0,999806973520	1	0,999903477444

Tablo 5.2: Rastgele Orman Performans Ölçütleri

5.3 Naive Bayes

Naive Bayes (NB) sınıflandırma algoritması, olasılık teorisine dayalı bir makine öğrenimi yöntemidir. Bu algoritma, Bayes teoremini temel alarak bir örneğin belirli bir sınıfa ait olma olasılığını hesaplar. "Naive" terimi, bu algoritmanın her bir özelliğin birbirinden bağımsız olduğunu varsayma ile ilgilidir, yani özellikler arasında ilişki olmadığını düşünür. Bu varsayım, modelin basitleştirilmesini sağlar ve hesaplama açısından avantaj sağlar (Uludağ, Gürsoy, 2020).

Veri setinin hazırlanması aşamasında, özellikler (features) ve etiketler (labels) olarak ayrılması gerekmektedir. Özellikler, örneğin sınıflandırılması için kullanılacak olan veri noktalarının özelliklerini temsil ederken, etiketler bu veri noktalarının doğru sınıfını belirtir. Bu ayrım, modelin doğru şekilde eğitilmesini ve test edilmesini sağlar.

```
import pandas as pd
import matplotlib.pyplot as plt
from imblearn.over_sampling import SMOTE
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
from sklearn.preprocessing import StandardScaler
from sklearn.naive_bayes import GaussianNB

# Veri seti yüklendi.
data = pd.read_csv("data.csv")

# Özellikler ve Etiketler ayrıldı.
features = data.drop('Class', axis=1)
labels = data['Class']
```

Şekil 5.9: Naive Bayes Özelliklere ve Etiketlere Ayırma Kodu

Özellikler arasındaki ilişkisizlik varsayımını korumak için özelliklerin ölçeklendirilmesi önemlidir. Bu adım, modelin daha doğru sonuçlar üretmesini sağlar. Dolandırıcılık tespiti gibi durumlarda, genellikle azınlık sınıfının (dolandırıcılık işlemleri) sayısı az olduğundan dengesizlik sorunuyla karşılaşılır. Bu dengesizliği gidermek için, Sentetik Azınlık Yüksek Örnekleme Tekniği (SMOTE) gibi teknikler kullanılabilir.

```
# Özellikleri ölçeklendirme
scaler = StandardScaler()
scaled_features = scaler.fit_transform(features)

# Sınıf dengesizliği nedeniyle, azınlık sınıfını artırarak
# veri setini dengelemek için SMOTE tekniği kullanıldı.
smote = SMOTE(sampling_strategy='minority')
resampled_features, resampled_labels = smote.fit_resample(scaled_features, labels)
```

Şekil 5.10: Naive Bayes Ölçeklendirme ve SMOTE Kodu

Veri seti, genellikle %80 eğitim ve %20 test verisi olarak ayrılır. Eğitim verisi modelin parametrelerini öğrenmek ve içsel yapılarını ayarlamak için kullanılırken, test verisi modelin gerçek dünya verileri üzerinde ne kadar iyi performans göstereceğini değerlendirmek için ayrılır.

Daha sonra, Naive Bayes modeli oluşturulur. Bu model, veri setinin özelliklerinin dağılımını varsayarak öğrenme işlemini gerçekleştirir. Gaussian Naive Bayes yöntemi, özelliklerin normal dağılıma sahip olduğunu varsayar ve bu doğrultuda özelliklerin ortalama ve varyansını tahmin ederek modeli eğitir (Uludağ, Gürsoy, 2020).

Model eğitim aşamasında, eğitim verisi kullanılarak model parametreleri öğrenilir. Bu parametreler, özelliklerin ortalama ve varyansı gibi modelin özniteliklerini temsil eder. Eğitilmiş model, test verisi üzerinde kullanılarak önceden belirlenmemiş örnekler için sınıflandırma yapar. Sonuç olarak, her bir test örneği için modelin tahmin ettiği sınıflar elde edilir. Bu tahminler, modelin performansını değerlendirmek için kullanılır ve gerçek etiketlerle karşılaştırılarak modelin doğruluğu hesaplanır.

```
# Veri kümesini eğitim ve test alt kümelere ayrıldı.
# Test kümesi boyutu %20, rastgele durum (random_state) ise 42 olarak belirlendi.
train_features, test_features, train_labels, test_labels = train_test_split(
    resampled_features, resampled_labels, test_size=0.20, random_state=42
)

# Naive Bayes modeli oluşturuldu (Gaussian Naive Bayes kullanıldı).
naive_bayes_model = GaussianNB()

# Model eğitildi.
naive_bayes_model.fit(train_features, train_labels)

# Eğitilmiş model kullanılarak test verileri için tahminler yapıldı.
predicted_labels = naive_bayes_model.predict(test_features)
```

Şekil 5.11: Naive Bayes Veri Ayırma ve Model Eğitimi Kodu

Eğitim tamamlandıktan sonra, model test verisi üzerinde değerlendirilir ve performans ölçütleri hesaplanır. Bu ölçütler, modelin sınıflandırma yeteneğini değerlendirmek için kullanılır ve modelin gerçek dünya verileri üzerinde ne kadar iyi performans gösterdiğini gösterir.

```
# Doğruluk, Kesinlik, Duyarlılık ve F1 Skoru değerleri hesaplandı.
accuracy = accuracy_score(test_labels, predicted_labels)
precision = precision_score(test_labels, predicted_labels)
recall = recall_score(test_labels, predicted_labels)
f1 = f1_score(test_labels, predicted_labels)
```

Şekil 5.12: Naive Bayes Performans Ölçütleri Kodu

Tüm bu adımların ardından naive bayes modelinin performansı aşağıdaki tabloda gösterilmektedir;

Doğruluk	Kesinlik	Duyarlılık	F1 Skoru
0,913845558623	0,971440562795	0,853113591688	0,908440174934

Tablo 5.3: Naive Bayes Performans Ölçütleri

5.4 Lojistik Regresyon

Lojistik Regresyon girdi değişkenleri ile çıktı değişkeni arasındaki ilişkiyi modellemek için bir doğrusal regresyon modelini kullanır ve çıktıyı [0, 1] aralığına sıkıştırmak için bir logit dönüşümü uygular. Bu sayede çıktıyı bir olasılık değeri olarak yorumlamak mümkün olur. LR'nin temel varsayımlarından biri, girdi değişkenleri ile çıktı arasındaki ilişkinin doğrusal olduğudur, yani LR, bağımsız değişkenlerin ağırlıklı toplamının doğrusal bir fonksiyonu olarak çıktıyı modellemektedir (Udjianto, 2006).

LR'nin eğitimi sırasında, model parametreleri (katsayılar) veri seti üzerindeki gözlemlere uyum sağlamak için en uygun şekilde ayarlanır. Bu genellikle maksimum olabilirlik tahminine dayalı bir optimizasyon süreci ile gerçekleştirilir. Model, belirli bir eğitim veri setine uyum sağladıktan sonra, test veri setindeki gözlemleri tahmin etmek için kullanılabilir (Udjianto, 2006).

Veri setinin hazırlanması aşamasında, özellikler (features) ve etiketler (labels) olarak ayrılması gerekmektedir. Özellikler, modelin eğitilmesi ve sınıflandırma işlemi için kullanılacak olan girdi değişkenlerini temsil eder. Bu değişkenler, veri noktalarının özelliklerini belirtir ve genellikle veri setinin sütunlarına karşılık gelir. Etiketler ise, modelin öğrenmek istediği hedef değişkeni ifade eder. Sınıflandırma problemlerinde etiketler genellikle iki sınıftan oluşur: pozitif ve negatif sınıflar. Kredi kartı dolandırıcılığının tespitinde pozitif etiket dolandırıcılığı belirtirken, negatif etiket dolandırıcılık olmayan işlemleri temsil eder.

```python
import pandas as pd
import matplotlib.pyplot as plt
from imblearn.over_sampling import SMOTE
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression

# Veri seti yüklendi.
data = pd.read_csv("data.csv")

# Özellikler ve Etiketler ayrıldı.
features = data.drop('Class', axis=1)
labels = data['Class']
```

Şekil 5.13: Lojistik Regresyon Özelliklere ve Etiketlere Ayırma Kodu

Veri setinin dengesizliği nedeniyle bazı teknikler uygulanmıştır. Özelliklerin ölçeklendirilmesi, modelin her bir özelliği aynı ölçekte ele almasını sağlar. Ölçeklendirme işlemi, modelin daha dengeli ve tutarlı bir şekilde öğrenmesine yardımcı olur. SMOTE (Sentetik Azınlık Yüksek Örnekleme Tekniği), azınlık sınıfındaki nadir örnekleri artırmak için kullanılan bir tekniktir. Dolandırıcılık gibi nadir olayları tanımlayan azınlık sınıfının örneklerini artırmak, modelin nadir olayları daha iyi tanımlamasına ve dengesizlik sorununun üstesinden gelmesine yardımcı olur.

```python
# Özellikleri ölçeklendirme
scaler = StandardScaler()
scaled_features = scaler.fit_transform(features)

# Sınıf dengesizliği nedeniyle, azınlık sınıfını artırarak
# veri setini dengelemek için SMOTE tekniği kullanıldı.
smote = SMOTE(sampling_strategy='minority')
resampled_features, resampled_labels = smote.fit_resample(scaled_features, labels)
```

Şekil 5.14: Lojistik Regresyon Ölçeklendirme ve SMOTE Kodu

Veri setinin hazırlanması aşamasından sonra lojistik regresyon modelinin eğitim aşaması başlar. İlk olarak, veri seti eğitim ve test alt kümelerine ayrılır. Bu işlem, modelin eğitim verileri üzerinde öğrenmesini sağlayacak ve daha sonra test verileri üzerinde performansını değerlendirecektir. Eğitim için kullanılacak örnekler genellikle veri setinin büyük bir yüzdesini oluştururken, test için kullanılacak örnekler genellikle daha küçük bir yüzdedir. Veri setinin belirli bir oranda (genellikle %80 eğitim ve %20 test) eğitim ve test alt kümelerine ayrılması sağlanmaktadır.

Veri setinin eğitim ve test alt kümelerine ayrılmasının ardından, lojistik regresyon modeli oluşturulur. Lojistik regresyon, bağımlı değişkenin kategorik olduğu durumlarda kullanılan bir regresyon modelidir. Bu model, bağımsız değişkenler arasındaki ilişkiyi kullanarak bir olayın olasılığını tahmin etmeye çalışır. Model oluşturulurken, genellikle Scikit-learn kütüphanesinin LogisticRegression sınıfı kullanılır.

Oluşturulan lojistik regresyon modeli, eğitim veri seti üzerinde eğitilir. Bu adımda, model veri setindeki örnekleri kullanarak bağımlı değişkenin olasılığını tahmin etmeyi öğrenir. Lojistik regresyon modeli, eğitim süreci boyunca veri setindeki örnekler arasındaki ilişkiyi belirler ve model parametrelerini ayarlar.

```
# Veri kümesini eğitim ve test alt kümelere ayrıldı.
# Test kümesi boyutu %20, rastgele durum (random_state) ise 42 olarak belirlendi.
train_features, test_features, train_labels, test_labels = train_test_split(
    resampled_features, resampled_labels, test_size=0.20, random_state=42
)

# Lojistik Regresyon modeli oluşturuldu.
logistic_regression_model = LogisticRegression()

# Model eğitildi.
logistic_regression_model.fit(train_features, train_labels)

# Eğitilmiş model kullanılarak test verileri için tahminler yapıldı.
predicted_labels = logistic_regression_model.predict(test_features)
```

Şekil 5.15: Lojistik Regresyon Veri Ayırma ve Model Eğitimi Kodu

Eğitim tamamlandıktan sonra, model test verisi üzerinde değerlendirilir ve performans ölçütleri hesaplanır. Bu ölçütler, modelin sınıflandırma yeteneğini değerlendirmek için kullanılır ve modelin gerçek dünya verileri üzerinde ne kadar iyi performans gösterdiğini gösterir.

```
# Doğruluk, Kesinlik, Duyarlılık ve F1 Skoru değerleri hesaplandı.
accuracy = accuracy_score(test_labels, predicted_labels)
precision = precision_score(test_labels, predicted_labels)
recall = recall_score(test_labels, predicted_labels)
f1 = f1_score(test_labels, predicted_labels)
```

Şekil 5.16: Lojistik Regresyon Performans Ölçütleri Kodu

Tüm bu adımların ardından lojistik regresyon modelinin performansı aşağıdaki tabloda gösterilmektedir;

Doğruluk	Kesinlik	Duyarlılık	F1 Skoru
0,948578161546	0,973934000742	0,922037349059	0,947275415630

Tablo 5.4: Lojistik Regresyon Performans Ölçütleri

5.5 K-En Yakın Komşu

K-En Yakın Komşu (KNN), temel bir sınıflandırma ve regresyon algoritmasıdır. Bu algoritma, bir veri noktasını etiketlemek veya bir değer tahmin etmek için çevresindeki K en yakın komşusunun etiketlerine veya değerlerine dayanır. Temel prensibi, benzerlik ölçüsü kullanarak bir veri noktasının komşularını belirlemek ve bu komşuların etiketlerini veya değerlerini kullanarak hedef değişkeni tahmin etmektir (Uğuz S., Oral Ç. , 2019).

KNN algoritması çalışırken, öncelikle bir veri noktasının komşularını belirlemek için bir benzerlik metriği hesaplanır. Genellikle kullanılan benzerlik metrikleri arasında Euclidean mesafe, Manhattan mesafe ve Minkowski mesafe bulunur. Daha sonra, belirlenen K değeri kadar en yakın komşu seçilir. Sınıflandırma durumunda, bu komşuların çoğunluğunun sınıf etiketi verilen veri noktasının sınıf etiketi olarak tahmin edilir. Regresyon durumunda ise, komşuların hedef değişken değerlerinin ortalaması alınarak tahmin yapılır (Zhang, 2010).

Veri setinin hazırlanması aşamasında, öncelikle kullanılacak olan özellikler (features) ve bu özelliklere karşılık gelen etiketler (labels) belirlenmelidir. K-En Yakın Komşu (KNN) algoritması, veri noktalarının benzerlik ölçüsüne dayanarak sınıflandırma ve regresyon yapar. Bu nedenle, kullanılacak özelliklerin ve etiketlerin doğru bir şekilde belirlenmesi önemlidir.

```
import pandas as pd
import matplotlib.pyplot as plt
from imblearn.over_sampling import SMOTE
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier

# Veri seti yüklendi.
data = pd.read_csv("data.csv")

# Özellikler ve Etiketler ayrıldı.
features = data.drop('Class', axis=1)
labels = data['Class']

# Özellikleri ölçeklendirme
scaler = StandardScaler()
scaled_features = scaler.fit_transform(features)
```

Şekil 5.17: K-En Yakın Komşu Özelliklere ve Etiketlere Ayırma Kodu

Özelliklerin ölçeklendirilmesi, farklı özellikler arasındaki ölçek farklılıklarını ortadan kaldırarak, KNN algoritmasının doğru bir şekilde çalışmasını sağlar. Ölçek farklılıkları, mesafe hesaplamalarını yanıltabilir ve algoritmanın yanlış sonuçlar üretmesine neden olabilir. Özellikle, KNN algoritması mesafe tabanlı bir algoritma olduğu için, özelliklerin ölçeklendirilmesi önemlidir. Ölçeklendirme işlemi genellikle özellik değerlerini belirli bir aralığa ölçeklendirerek yapılır, bu da algoritmanın daha tutarlı ve doğru sonuçlar üretmesini sağlar. KNN gibi algoritmaların, dengesiz veri setlerindeki azınlık sınıfını yanlış sınıflandırma eğiliminde olabileceği bilinmektedir. Bu nedenle, azınlık sınıfını artırmak ve veri setini daha dengeli hale getirmek için SMOTE tekniği kullanılır.

```
# Özellikleri ölçeklendirme
scaler = StandardScaler()
scaled_features = scaler.fit_transform(features)

# Sınıf dengesizliği nedeniyle, azınlık sınıfını artırarak
# veri setini dengelemek için SMOTE tekniği kullanıldı.
smote = SMOTE(sampling_strategy='minority')
resampled_features, resampled_labels = smote.fit_resample(scaled_features, labels)
```

Şekil 5.18: K-En Yakın Komşu Ölçeklendirme ve SMOTE Ayırma Kodu

KNN algoritması, eğitim aşamasında herhangi bir model oluşturma veya parametre ayarlama işlemi gerektirmez. Bunun yerine, Scikit-learn kütüphanesinin KNeighborsClassifier sınıfı kullanılarak bir KNN modeli oluşturulur.

Bu sınıf, KNN algoritmasının temel prensiplerini uygular ve eğitim verilerini bellekte saklar. KNeighborsClassifier sınıfı, eğitim verilerini içeren bir yapı oluşturmak için eğitim veri kümesini alır. Bu yapı, KNN algoritmasının sınıflandırma işlemini gerçekleştirmek için kullanılır. Bu aşamada, eğitim verileri belleğe yüklenir ve sınıflandırma için hazır hale getirilir. Ancak, eğitim verileri sadece bellekte saklanır ve bir model oluşturma süreci gerçekleştirilmez.

KNN algoritması, sınıflandırma işlemini gerçekleştirirken, bir test veri noktasının etrafındaki K en yakın komşuyu belirler. Bu komşuların sınıfları dikkate alınarak, test veri noktası belirli bir sınıfa atanır. Bu süreç, KNeighborsClassifier sınıfının predict() yöntemi kullanılarak gerçekleştirilir.

```
# Veri kümesini eğitim ve test alt kümelere ayrıldı.
# Test kümesi boyutu %20, rastgele durum (random_state) ise 42 olarak belirlendi.
train_features, test_features, train_labels, test_labels = train_test_split(
    resampled_features, resampled_labels, test_size=0.20, random_state=42
)

# K-Nearest Neighbors (KNN) modeli oluşturuldu.
knn_model = KNeighborsClassifier()

# Model eğitildi.
knn_model.fit(train_features, train_labels)

# Eğitilmiş model kullanılarak test verileri için tahminler yapıldı.
predicted_labels = knn_model.predict(test_features)
```

Şekil 5.19: K-En Yakın Komşu Veri Ayırma ve Model Eğitimi Kodu

Eğitim tamamlandıktan sonra, model test verisi üzerinde değerlendirilir ve performans ölçütleri hesaplanır. Bu ölçütler arasında doğruluk (accuracy), kesinlik (precision), duyarlılık (recall) ve F1 skoru bulunur. Bu ölçütler, modelin ne kadar iyi performans gösterdiğini belirlemek için kullanılır.

```
# Doğruluk, Kesinlik, Duyarlılık ve F1 Skoru değerleri hesaplandı.
accuracy = accuracy_score(test_labels, predicted_labels)
precision = precision_score(test_labels, predicted_labels)
recall = recall_score(test_labels, predicted_labels)
f1 = f1_score(test_labels, predicted_labels)
```

Şekil 5.20: K-En Yakın Komşu Performans Ölçütleri Kodu

Tüm bu adımların ardından k-en yakın komşu (KNN) modelinin performansı aşağıdaki tabloda gösterilmektedir;

Doğruluk	Kesinlik	Duyarlılık	F1 Skoru
0,998927246188	0,997863322708	1	0,998930518786

Tablo 5.5: K-En Yakın Komşu Performans Ölçütleri

5.6 Karar Ağaçları

Algoritmanın temelinde bulunan prensip, veri setindeki özelliklerin değerlerine göre en iyi bölünmeyi bulmaktır. En iyi bölünme, veri setini en homojen alt gruplara bölen bir özellik ve eşik değeri kombinasyonudur. Karar Ağaçları, bu bölünmeleri yaparken genellikle bilgi kazancı, Gini impurity veya hata azaltma gibi ölçütleri kullanır. Böylece, her bir dalda temsil edilen veri noktalarının benzerliklerini ve farklılıklarını değerlendirir ve en iyi bölünmeyi seçer (Sahin, Duman, 2011).

Karar Ağaçları, sonuç olarak bir ağaç yapısı oluşturur. Bu yapıda, kök düğümden başlayarak her bir iç düğüm bir özelliği temsil eder ve dallar ise bu özelliğin değerlerine göre bölünmeyi gösterir.

Yaprak düğümler ise sonuç sınıflarını veya regresyon tahminlerini içerir. Bu şekilde, yeni bir veri noktası geldiğinde, Karar Ağacı bu yapının yardımıyla sınıflandırma veya tahmin işlemini gerçekleştirir.

Veri setinin hazırlanması aşamasında, öncelikle özellikler (features) ve etiketler (labels) olarak ayrılması gerekmektedir. Karar Ağaçları algoritması için veri setinin bu şekilde hazırlanması önemlidir çünkü algoritma, özelliklerin değerlerine göre karar yapısı oluştururken, bu yapının hangi sınıfı veya değeri tahmin etmesi gerektiğini öğrenir.

Özellikler ve etiketlerin ayrılması işlemi genellikle pandas kütüphanesi kullanılarak gerçekleştirilir. Veri seti bir DataFrame olarak yüklenir ve DataFrame üzerinde gerekli işlemler yapılarak özellikler ve etiketler ayrılır.

```python
import pandas as pd
import matplotlib.pyplot as plt
from imblearn.over_sampling import SMOTE
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
from sklearn.preprocessing import StandardScaler
from sklearn.tree import DecisionTreeClassifier

# Veri seti yüklendi.
data = pd.read_csv("data.csv")

# Özellikler ve Etiketler ayrıldı.
features = data.drop('Class', axis=1)
labels = data['Class']
```

Şekil 5.21: Karar Ağaçları Özelliklere ve Etiketlere Ayırma Kodu

Diğer modellerde olduğu gibi bu modelde de veri dengesizliğiyle başa çıkmak ve daha doğru sonuçlar almak için özelliklik ölçeklendirmesi ve SMOTE yöntemleri kullanılmaktadır.

Özelliklerin ölçeklendirilmesi genellikle sklearn.preprocessing.StandardScaler gibi kütüphaneler kullanılarak yapılırken, SMOTE tekniği için imbalanced-learn kütüphanesi içindeki imblearn.over_sampling.SMOTE fonksiyonu kullanılır. Bu fonksiyon, azınlık sınıfındaki örnekleri sentetik olarak üreterek veri setini dengeler.

```python
# Özellikleri ölçeklendirme
scaler = StandardScaler()
scaled_features = scaler.fit_transform(features)

# Sınıf dengesizliği nedeniyle, azınlık sınıfını artırarak
# veri setini dengelemek için SMOTE tekniği kullanıldı.
smote = SMOTE(sampling_strategy='minority')
resampled_features, resampled_labels = smote.fit_resample(scaled_features, labels)
```

Şekil 5.22: Karar Ağaçları Ölçeklendirme ve SMOTE Kodu

Eğitim sürecinde, karar ağaçları algoritması veri setindeki örnekleri kullanarak ağaç yapısını oluşturur. Bu ağaç yapısı, veri setinin içerdiği karmaşıklığı ve ilişkileri temsil eder. Model, eğitim verisi üzerinde bu ağaç yapısını oluşturarak öğrenir ve sonuçları tahmin etmek için bu yapıyı kullanır. Sklearn kütüphanesinin DecisionTreeClassifier sınıfı kullanılarak Karar Ağaçları modeli oluşturulur. Bu sınıf, veri setindeki örneklerin özelliklerine göre bir ağaç yapısı oluşturur ve sınıflandırma işlemini gerçekleştirir.

```
# Veri kümesini eğitim ve test alt kümelere ayrıldı.
# Test kümesi boyutu %20, rastgele durum (random_state) ise 42 olarak belirlendi.
train_features, test_features, train_labels, test_labels = train_test_split(
    resampled_features, resampled_labels, test_size=0.20, random_state=42
)

# Karar Ağacı modeli oluşturuldu.
decision_tree = DecisionTreeClassifier()

# Model eğitildi.
decision_tree.fit(train_features, train_labels)

# Eğitilmiş model kullanılarak test verileri için tahminler yapıldı.
predicted_labels = decision_tree.predict(test_features)
```

Şekil 5.23: Karar Ağaçları Veri Ayırma ve Model Eğitimi Kodu

Eğitim tamamlandıktan sonra sonuçlar test edilir ve performans ölçütlerinin değerleri belli olur. Modelin ne kadar iyi çalıştığını anlamak için bu ölçütlerin birlikte değerlendirilmesi önemlidir. Çalışmamızda Sklearn kütüphanesinin metrics modülü kullanılarak bu performans ölçütleri hesaplanmaktadır. Bu metrikler, modelin gerçek dünya verileri üzerinde ne kadar başarılı olduğunu değerlendirmek için önemlidir.

```
# Doğruluk, Kesinlik, Duyarlılık ve F1 Skoru değerleri hesaplandı.
accuracy = accuracy_score(test_labels, predicted_labels)
precision = precision_score(test_labels, predicted_labels)
recall = recall_score(test_labels, predicted_labels)
f1 = f1_score(test_labels, predicted_labels)
```

Şekil 5.24: Karar Ağaçları Performans Ölçütleri Kodu

Tüm bu adımların ardından karar ağaçları modelinin performansı aşağıdaki tabloda gösterilmektedir;

Doğruluk	Kesinlik	Duyarlılık	F1 Skoru
0,998338110898	0,997564181197	0,999122437518	0,998342701309

Tablo 5.6: Karar Ağaçları Performans Ölçütleri

Bölüm 6

Bulgular

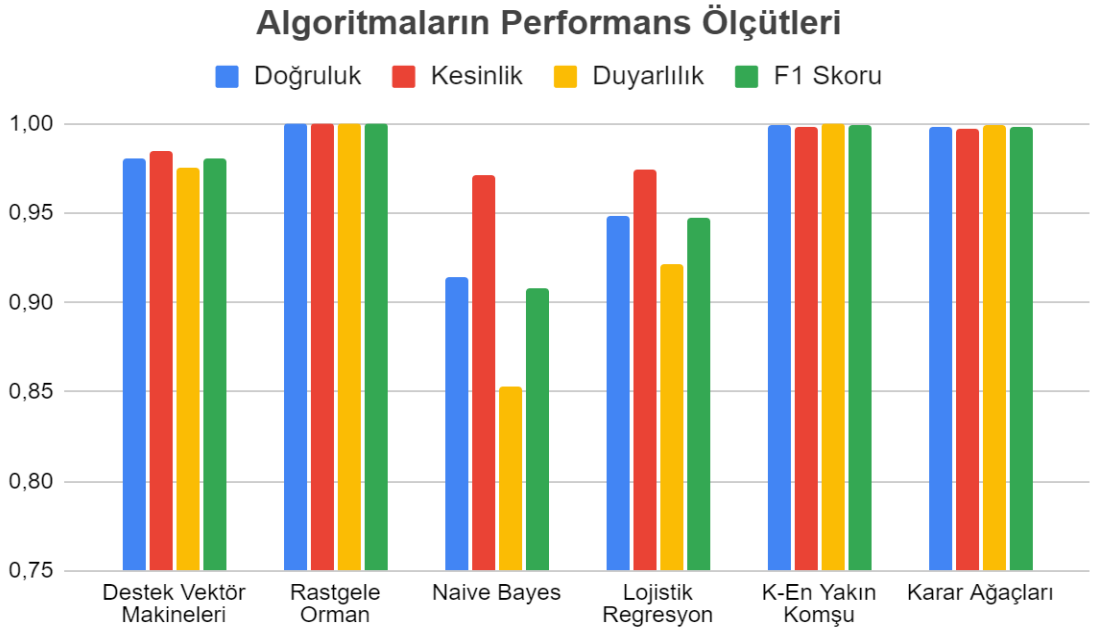
Bu çalışma, kredi kartı dolandırıcılığı tespiti için çeşitli makine öğrenme algoritmalarının performansını değerlendirmektedir. Destek Vektör Makineleri, Rastgele Orman, Naive Bayes, Lojistik Regresyon, K-En Yakın Komşu ve Karar Ağaçları algoritmalarının kredi kartı dolandırıcılığı tespiti üzerindeki etkinliği ve performansı incelenmiştir. Her bir algoritmanın güçlü ve zayıf yönleri belirlenerek, kredi kartı dolandırıcılığını tespit etme sürecinde hangi algoritmanın en etkili olduğunu belirlemek amaçlanmıştır. Algoritmaların performansı, gerçek finansal verilerle oluşturulan kredi kartı dolandırıcılığı veri seti üzerinde incelenmiştir. Veri dengesizliğini ele almak için, azınlık sınıfındaki dolandırıcılık vakalarını artırmak için Synthetic Minority Over-sampling Technique (SMOTE) gibi teknikler kullanılmıştır.

Yapılan çalışma neticesinde elde edilen algoritmalarının performans ölçütlerini gösteren tablo ve şekil aşağıda gösterilmektedir:

Algoritma	**Doğruluk**	**Kesinlik**	**Duyarlılık**	**F1 Skoru**
Destek Vektör Makineleri	0,9802	0,9848	0,9756	0,9802
Rastgele Orman	0,9999	0,9998	1	0,9999
Naive Bayes	0,9138	0,9714	0,8531	0,9084
Lojistik Regresyon	0,9486	0,9739	0,9220	0,9473
K-En Yakın Komşu	0,9989	0,9979	1	0,9989
Karar Ağaçları	0,9983	0,9976	0,9991	0,9983

Tablo 6.1: Algoritmaların Performans Ölçütleri

Çalışmada incelenen farklı makine öğrenme algoritmalarının birkaç ortak noktası bulunmaktadır. Öncelikle, Destek Vektör Makineleri (SVM), Rastgele Orman (RF), Naive Bayes (NB), Lojistik Regresyon (LR), K-En Yakın Komşu (KNN) ve Karar Ağaçları (DT) gibi algoritmaların hepsi denetimli öğrenme yaklaşımını benimser. Bu, veri setindeki özelliklerle hedef değişken arasındaki ilişkiyi anlamak ve yeni verilere genelleme yapmak için eğitilmiş verilerle çalıştıkları anlamına gelir. Ayrıca, her bir algoritmanın temel amacı, sahtekarlık işlemlerini gerçek işlemlerden ayırt etme yeteneğini maksimize etmektir.



Şekil 6.1: Algoritmaların Performans Ölçütlerinin Grafiği

Çalışmada incelenen farklı makine öğrenme algoritmalarının birkaç ortak noktası bulunmaktadır. Öncelikle, Destek Vektör Makineleri (SVM), Rastgele Orman (RF), Naive Bayes (NB), Lojistik Regresyon (LR), K-En Yakın Komşu (KNN) ve Karar Ağaçları (DT) gibi algoritmaların hepsi denetimli öğrenme yaklaşımını benimser. Bu, veri setindeki özelliklerle hedef değişken arasındaki ilişkiyi anlamak ve yeni verilere genelleme yapmak için eğitilmiş verilerle çalıştıkları anlamına gelir. Ayrıca, her bir algoritmanın temel amacı, sahtekarlık işlemlerini gerçek işlemlerden ayırt etme yeteneğini maksimize etmektir.

Her bir algoritmanın kendine özgü avantajları ve dezavantajları vardır. Doğru model seçimi, belirli bir uygulamanın gereksinimlerine, kullanılabilir kaynaklara ve veri setinin özelliklerine bağlıdır.

Destek Vektör Makineleri (SVM), veri noktalarını iki sınıfa ayıran en iyi ayrım hiperdüzlemi (sınırlayıcı) bulmaya çalışan güçlü bir sınıflandırma algoritmasıdır. Bu algoritma, doğrusal ve doğrusal olmayan sınıflandırma problemlerine uygulanabilir ve genellikle karmaşık karar sınırları çizebilme yeteneği ile bilinir. Özellikle yüksek boyutlu veri kümeleriyle iyi çalışabilir ve veri setlerindeki karmaşıklıkları anlamak için etkili bir araç olabilir. Ancak, SVM'nin eğitim süresi ve hesaplama gücü diğer bazı algoritmalara kıyasla daha fazla olabilir, özellikle de büyük veri setleri ile çalışırken bu durum daha belirgin hale gelebilir. Yine de doğru parametrelerle ayarlandığında ve uygun şekilde uygulandığında, SVM genellikle yüksek doğruluk ve güvenilirlik sağlayabilir.

Rastgele Orman, birden çok karar ağacını birleştirerek daha sağlam bir model oluşturan bir topluluk öğrenme tekniğidir. Her bir ağacın ayrı ayrı eğitilmesi ve sonuçlarının birleştirilmesi, aşırı öğrenme eğilimini azaltır ve genel performansı artırır. Bu, Rastgele Orman'ı özellikle aşırı uyum riski taşıyan diğer modellere kıyasla daha istikrarlı hale getirir. Ancak, algoritmanın karmaşıklığı ve hesaplama gücü, özellikle büyük veri kümeleriyle çalışırken dikkate alınması gereken önemli faktörlerdir. Karar Ağaçlarına benzer şekilde, Rastgele Orman da veri setindeki özelliklerin ve ilişkilerin daha net bir şekilde anlaşılmasını sağlar. Bununla birlikte, karar ağaçlarına kıyasla daha karmaşık bir yapıya sahip olması, Rastgele Orman'ın daha fazla hesaplama kaynağı gerektirmesine neden olabilir.

Naive Bayes, basit bir olasılık temeline dayanan ve özellikler arasındaki bağımsızlığı varsayan bir sınıflandırma algoritmasıdır. Bu algoritma, sınıflandırma işlemi sırasında her bir özelliğin sınıfı üzerindeki etkisini bağımsız olarak değerlendirir ve bu etkileri bir araya getirerek sonuç üretir. Naive Bayes, özellikle metin sınıflandırma gibi alanlarda başarılı sonuçlar verirken, daha karmaşık ilişkileri modelleme konusunda sınırlı olabilir. Diğer algoritmalara kıyasla eğitim süresi genellikle daha kısadır ve küçük boyutlu veri setlerinde iyi performans gösterebilir. Ancak, Naive Bayes'in bağımsızlık varsayımı bazı durumlarda gerçek verilere tam olarak uymayabilir ve bu da performansını etkileyebilir.

Lojistik Regresyon, bağımlı değişkenin kategorik olduğu durumlarda kullanılan bir regresyon modelidir. Bu algoritma, doğrusal bir sınırlayıcı kullanarak sınıflandırma yapar ve veri noktalarını iki veya daha fazla sınıfa ayırır. Lojistik Regresyon, özellikle yüksek boyutlu veri kümelerinde ve büyük veri setlerinde hızlı eğitim ve tahmin süreleri sağlar. Modelin açıklanabilirliği ve hesaplama maliyetinin düşük olması gibi avantajları bulunmaktadır. Ancak, doğrusal sınırlayıcılar karmaşık ve doğrusal olmayan ilişkileri modellemede zorlanabilir, bu da Lojistik Regresyon'un bazı durumlarda diğer algoritmalarla karşılaştırıldığında daha az esnek olmasına neden olabilir.

"K-En Yakın Komşu (KNN) algoritması, basit ve anlaşılır bir sınıflandırma yöntemi sunarken, özellikle küçük veri setlerinde iyi performans gösterebilir. Ancak, diğer algoritmalara kıyasla hesaplama maliyeti daha yüksektir, çünkü tahmin yapmak için tüm veri setini bellekte saklar ve her tahminde tüm veri noktalarını yeniden hesaplamak zorundadır. Ayrıca, KNN'nin sınıflandırma süreci, veri setindeki gürültüye ve dengesiz sınıf dağılımlarına duyarlı olabilir. Diğer yandan, KNN'nin doğruluğu ve performansı, komşu sayısı gibi belirli parametrelere bağlı olarak değişebilir. Bu nedenle, KNN algoritması veri setinin özelliklerine ve problem gereksinimlerine bağlı olarak tercih edilebilir bir seçenek olabilir.

Karar Ağaçları, veri özelliklerini ve hedef değişken arasındaki ilişkileri açık bir şekilde modelleyen açıklanabilir bir sınıflandırma algoritmasıdır. Bu özelliği, karar ağaçlarının sonuçlarını yorumlama ve anlama kolaylığı sağlar. Ancak, karar ağaçları genellikle aşırı öğrenme eğilimindedir ve gürültülü verilere karşı duyarlıdır. Bununla birlikte, bu aşırı öğrenme eğilimi, ensemble yöntemleri gibi tekniklerle düzeltilebilir ve genellikle karar ağaçlarına dayalı modellerin performansını artırır. Karar Ağaçları algoritması, diğer bazı karmaşık algoritmalarla karşılaştırıldığında daha az hesaplama kaynağı gerektirir ve büyük veri kümeleriyle bile iyi çalışabilir. Ancak, karmaşıklığı artırdıkça, karar ağaçlarının eğitim süresi ve modelin genel karmaşıklığı da artabilir.

Bu bulgular, kredi kartı dolandırıcılığı tespiti için kullanılan çeşitli makine öğrenme algoritmalarının performansını değerlendirirken her bir algoritmanın güçlü ve zayıf yönlerini ortaya koymaktadır. Seçim yaparken, uygulamanın gereksinimlerine, kullanılabilir kaynaklara, veri setinin özelliklerine ve spesifik hedeflere dikkat edilmelidir.

Bölüm 7

Sonuç

Bu çalışmada kredi kartı dolandırıcılığının tespiti için yapay zekâ teknolojisinden yararlanmak adına makine öğrenme algoritmalarının çalışma prensibi ve verimliliği değerlendirmiştir. Kullanılan algoritmalar arasında destek vektör makineleri, rastgele orman, naive bayes, lojistik regresyon, k-en yakın komşu ve karar ağaçları bulunmaktadır.

Destek Vektör Makineleri (SVM) modeli, %98.02 doğruluk, %98.48 kesinlik, %97.56 duyarlılık ve %98.02 F1 skoru elde etmiştir. SVM, veri noktalarını bir hiperdüzlem ile ayırarak sınıflandırma yapar. Bu model, yüksek boyutlu verilerle iyi çalışabilir ve karmaşık karar sınırları çizebilir. Ancak, eğitim süresi ve hesaplama gücü daha fazladır.

Rastgele Orman modeli, %99.99 doğruluk, %99.98 kesinlik, %100 duyarlılık ve %99.99 F1 skoru ile en yüksek performansı sergilemiştir. Rastgele Orman, birden fazla karar ağacını birleştirerek daha güçlü ve genelleştirilebilir bir model oluşturur. Bu sayede, aşırı uyum riski azalır ve performans artar. Ancak, algoritma karar ağaçlarına kıyasla daha karmaşıktır ve daha fazla hesaplama kaynağı gerektirir.

Naive Bayes modeli, %91.38 doğruluk, %97.14 kesinlik, %85.31 duyarlılık ve %90.84 F1 skoru ile diğer modellere kıyasla daha düşük bir performans göstermiştir. Naive Bayes, basit bir olasılık temeline dayanan bir sınıflandırma algoritmasıdır. Model, özellikler arasındaki bağımsızlığı varsayar, bu da bazı durumlarda gerçek verilere uymayabilir.

Lojistik Regresyon modeli, %94.86 doğruluk, %97.39 kesinlik, %92.20 duyarlılık ve %94.73 F1 skoru ile orta düzeyde bir performans sergilemiştir. Lojistik Regresyon, bağımlı değişkenin kategorik olduğu durumlarda kullanılan bir regresyon modelidir. Doğrusal bir sınırlayıcı kullanarak sınıflandırma yapar.

K-En Yakın Komşu modeli, %99.89 doğruluk, %99.79 kesinlik, %100 duyarlılık ve %99.89 F1 skoru ile yüksek bir performans göstermiştir. K-En Yakın Komşu, bir örneğin sınıfını belirlemek için komşularının etiketlerini kullanır. Bu model, veri setindeki örneklerin benzerliklerine dayanarak sınıflandırma yapar.

Karar Ağaçları modeli, %99.83 doğruluk, %99.76 kesinlik, %99.91 duyarlılık ve %99.83 F1 skoru ile yüksek bir performans sergilemiştir. Karar Ağaçları, veri özelliklerini ve hedef değişken arasındaki ilişkileri açık bir şekilde modelleyen açıklanabilir bir sınıflandırma algoritmasıdır. Ancak, aşırı öğrenme eğilimindedir ve gürültülü verilere karşı duyarlıdır.

Destek Vektör Makineleri, Rastgele Orman ve Karar Ağaçları gibi algoritmaların kredi kartı dolandırıcılığı tespitindeki performansı öne çıkmaktadır. Özellikle, veri setindeki belirleyici özelliklerin tanımlanması ve hızlı karar alma gereksinimlerinde bu algoritmaların etkin olduğu belirlenmiştir. Destek Vektör Makineleri, Rastgele Orman ve Karar Ağaçları, veri setindeki önemli özellikleri belirleme ve sıralama konusunda etkili bir araç olarak öne çıkmaktadır. Bu algoritmaların karar mekanizmaları, diğerlerine kıyasla daha şeffaf ve anlaşılabilir olma eğilimindedir.

Bu sonuçlar, farklı makine öğrenme algoritmalarının kredi kartı dolandırıcılığı tespiti konusundaki etkinliğini değerlendirirken, her bir algoritmanın güçlü ve zayıf yönlerini vurgulamaktadır. Seçim yaparken, uygulamanın gereksinimlerine, kullanılabilir kaynaklara, veri setinin özelliklerine ve spesifik hedeflere dikkat edilmelidir. Bu nedenle algoritma seçimi ayrıntılı bir analiz gerektirir ve seçilen modelin performansını sürekli olarak izlemek ve gerekirse müdahale etmek gerekmektedir.

Bu değişik bakış açısı, kredi kartı dolandırıcılığının tespitinde daha etkili ve özel yaklaşımların geliştirilmesine yol açabilir, böylece farklı algoritmaların bir arada kullanıldığı ve karşılaştırıldığı yeni araştırmalar teşvik edilmektedir. İleriye dönük çalışmalar, bu algoritmaların daha da geliştirilmesi ve özelleştirilmesi üzerine odaklanabilir, dolayısıyla dolandırıcılık tespitinde daha etkin bir yaklaşım sağlanabilir. Bu araştırma, finansal kurumlar için maliyet tasarrufu sağlamanın yanı sıra tüketicilere daha güvenli bir alışveriş deneyimi sunma potansiyeline sahiptir.

Kaynaklar

Aaron Hertzmann, D. J. (2015). Machine Learning and Data Mining. *Machine Learning and Data Mining,* s. 115-117.

Adeel, M., & Hussain, S. (2017). ATM Skimming and Its Effects on Banking Industry: A Case Study of Pakistan. *International Journal of Academic Research in Accounting, Finance, and Management Sciences,* s. 2-5.

Ahmad, A., & Saini, D. (2019). Machine Learning Based Approach for Credit Card Fraud Detection. *International Conference on Computational Intelligence and Knowledge Economy (ICCIKE,)* s. 1-6. IEEE.

Awoyemi J. O., Adetunmbi A. O., Oluwadare S. A., Credit card fraud detection using machine learning techniques: A comparative analysis, *International Conference on Computing Networking and Informatics (ICCNI),* s 1-9, (2017).

Bhattacharyya, S. J. (2011). Data mining for credit card fraud: A comparative study. *Decision Support Systems,* s. 602-613.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research,* s. 321-357.

Demsar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research,* 1-30.

Jin, M., Gao, L., & Liu, Y. (2021). Multi-factor Authentication for Financial Applications: A Survey. *Journal of Network and Computer Applications,* s. 187-189.

Kaggle, 2022. Credit Card Fraud Detection Dataset. Credit Card Fraud Detection (kaggle.com)

Louppe, G. (2015). Understanding Random Forests: From Theory to Practice.

Phua, C., Lee, V., Smith, K., & Gayler, R. (2010). A Comprehensive Survey of Data Mining-based Fraud Detection Research. *The Computer Journal*, s. 2-6.

Powers, D. M. (2011). Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation. *Journal of Machine Learning Technologies*, s 37–63.

Sahin Y., Duman E., Detecting Credit Card Fraud by Decision Trees and Support Vector Machines, *Proceedings of International Multi-Conference of Engineers and Computer Scientists*, s 1: 1-6, (2011).

Santos, A., & Maynard, S. (2018). Financial Fraud Prevention and Anti-Money Laundering Compliance: The Role of Artificial Intelligence. *European Business Organization Law Review*, s. 29-56.

Shen, J., & Hsiao, K. F. (2021). Fraud Detection and Prevention with Big Data Analytics and Machine Learning. *Information & Management*, s. 58.

Udjianto, A. W. (2006). Statistical methods for credit card fraud detection. *Section on Quality and Productivity*, s 34-39.

Uğuz S., Oral O., PV Güç Santrallerinden Elde Edilecek Enerjinin Makine Öğrenmesi Metotları Kullanılarak Tahmin Edilmesi, *International Journal of Engineering Research and Development*, s 769-779, 2019.

Uludağ O., Gürsoy A., On the Financial Situation Analysis with KNN and Naive Bayes Classification Algorithms, *Journal of the Institute of Science and Technology*, s 2881-2888, 2020

Yero, S. I. (2018). Types and Patterns of Fraud in the Banking Industry. *International Journal of Scientific and Research Publications*, s. 8-10.

Yıldız B., Applying Decision Tree Techniques to Classify European Football Teams, *Journal of Soft Computing and Artificial Intelligence*, s 86-91, 2020.

Zhang S., KNN-CF Approach: Incorporating Certainty Factor to kNN Classification, *IEEE Intelligent Informatics Bulletin*, 2010.

.