**T.C.**
**ONDOKUZ MAYIS ÜNİVERSİTESİ**
**LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**
**İSTATİSTİK ANA BİLİM DALI**

# VERİ ODAKLI KARAR ALMADA MAKİNE ÖĞRENMESİ ALGORİTMALARI

Doktora Tezi

**Muhammed KARA**

Danışman
**Prof. Dr. Yüksel TERZİ**

SAMSUN
2024

**T.C.**
**ONDOKUZ MAYIS ÜNİVERSİTESİ**
**LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**
**İSTATİSTİK ANA BİLİM DALI**

# VERİ ODAKLI KARAR ALMADA MAKİNE ÖĞRENMESİ ALGORİTMALARI

Doktora Tezi

**Muhammed KARA**

Danışman

**Prof.Dr. Yüksel TERZİ**

SAMSUN
2024

# TEZ KABUL VE ONAYI

**Muhammed KARA** tarafından, **Prof. Dr. Yüksel TERZİ** danışmanlığında hazırlanan **"VERİ ODAKLI KARAR ALMADA MAKİNE ÖĞRENMESİ ALGORİTMALARI"** başlıklı bu çalışma, jürimiz tarafından 22.7.2024 tarihinde yapılan sınav sonucunda oy birliği ile başarılı bulunarak Doktora Tezi olarak kabul edilmiştir.

| | **Unvanı Adı Soyadı**<br>**Üniversitesi**<br>**Ana Bilim/Ana Sanat Dalı** | **Sonuç** |
|---|---|---|
| **Başkan** | Prof. Dr. Mehmet Ali CENGİZ<br>Ondokuz Mayıs Üniversitesi<br>İstatistik Ana Bilim Dalı | ☒ Kabul<br>☐ Ret |
| **Üye** | Prof. Dr. Yüksel TERZİ<br>Ondokuz Mayıs Üniversitesi<br>İstatistik Ana Bilim Dalı | ☒ Kabul<br>☐ Ret |
| **Üye** | Dr. Öğr. Üyesi İsmail İŞERİ<br>Ondokuz Mayıs Üniversitesi<br>Bilgisayar Mühendisliği Ana Bilim Dalı | ☒ Kabul<br>☐ Ret |
| **Üye** | Prof. Dr. Muhammet BEKÇİ<br>Sivas Cumhuriyet Üniversitesi<br>İstatistik ve Bilgisayar Bilimleri Ana Bilim Dalı | ☒ Kabul<br>☐ Ret |
| **Üye** | Doç. Dr. Tuba KOÇ<br>Çankırı Karatekin Üniversitesi<br>İstatistik Ana Bilim Dalı | ☒ Kabul<br>☐ Ret |

Bu tez, Enstitü Yönetim Kurulunca belirlenen ve yukarıda adları yazılı jüri üyeleri tarafından uygun görülmüştür.

Prof. Dr. Ahmet TABAK
Enstitü Müdürü

# BİLİMSEL ETİĞE UYGUNLUK BEYANI

Hazırladığım Doktora tezinin bütün aşamalarında bilimsel etiğe ve akademik kurallara riayet ettiğimi, çalışmada doğrudan veya dolaylı olarak kullandığım her alıntıya kaynak gösterdiğimi ve yararlandığım eserlerin Kaynaklar'da gösterilenlerden oluştuğunu, her unsurun enstitü yazım kılavuzuna uygun yazıldığını ve TÜBİTAK Araştırma ve Yayın Etiği Kurulu Yönetmeliği'nin 3. bölüm 9. maddesinde belirtilen durumlara aykırı davranılmadığını taahhüt ve beyan ederim.

Etik Kurul Gerekli mi ?

Evet ☐ (Gerekli ise ekler kısmına ekleyiniz)

Hayır ☒

21 /06 / 2024
Muhammed KARA

# TEZ ÇALIŞMASI ÖZGÜNLÜK RAPORU BEYANI

**Tez Başlığı :** VERİ ODAKLI KARAR ALMADA MAKİNE ÖĞRENMESİ ALGORİTMALARI

Yukarıda başlığı belirtilen tez çalışması için şahsım tarafından 21.06.2024 tarihinde intihal tespit programından alınmış olan özgünlük raporu sonucunda;

Benzerlik oranı : % 18

Tek kaynak oranı : % 3 çıkmıştır.

21 /06 / 2024
Prof.Dr.Yüksel TERZİ

# ÖZET

VERİ ODAKLI KARAR ALMADA MAKİNE ÖĞRENMESİ ALGORİTMALARI

Muhammed KARA
Ondokuz Mayıs Üniversitesi
Lisansüstü Eğitim Enstitüsü
İstatistik Ana Bilim Dalı
Doktora, Temmuz/2024
Danışman: Prof. Dr. Yüksel TERZİ

Bu tezde, makine öğrenmesi algoritmalarının temel prensipleri, uygulama alanları ve performans karşılaştırmaları ele alınmıştır. Makine öğrenmesi, veri setlerinden öğrenerek tahmin ve karar verme süreçlerini otomatikleştiren algoritmaların genel adıdır. Tezde; tahmin, sınıflandırma ve kümeleme olmak üzere üç ana kategoride makine öğrenmesi yöntemleri incelenmiştir.

Tezin odak noktası, veri odaklı karar alırken makine öğrenme yöntemlerinin istatistiksel açıdan analiz edilmesi ve bu analizin metodolojileridir. Ayrıca, bu analiz metodolojilerinin bir uygulaması olarak müşteri terk modellemesi yapılmış ve elde edilen sonuçların istatistiksel olarak anlamlılığı belirtilmiştir.Tezin yöntemi karma bir yaklaşıma dayanmaktadır ve literatür taraması, içerik analizi ve nitel araştırma adımlarını içermektedir. Literatür taraması ve benzer çalışmaların incelenmesi, yapılan analizlerin ve uygulamaların temelini oluşturmuştur.

Araç fiyatlarının tahmin edilmesi amacıyla "Automobile Price" veri seti kullanılmıştır. Veri seti, araçların çeşitli fiziksel, performans ve demografik özelliklerini içermektedir. Lineer Regresyon, Ridge Regresyon ve Lasso Regresyon algoritmalarıyla veri analiz edilmiştir. Orange programı kullanılarak yapılan analizlerde üç algoritma için de R-kare değeri 0,967 bulunmuş, Azure ML platformunda ise Lineer Regresyon için R-kare değeri 0,96 olarak elde edilmiştir. Bu sonuçlar, modellerin yüksek açıklayıcılık düzeyine sahip olduğunu göstermektedir.

Diğer bir örnek, çalışmada kullanılan WACustomerChurn veri seti tanıtılmıştır. Bu veri seti, müşteri terk modellemesi için kullanılan 21 değişken ve 7043 kayıttan oluşmaktadır. Veri setinde yer alan değişkenlerin işlevleri belirtilmiştir. Ayrıca, müşteri terk modellemesi için Spark platformunda Büyük Veri analitiği yapılmıştır. Python programlama dili kullanılarak Spark Mlib kütüphanelerinde bulunan makine öğrenmesi algoritmaları uygulanmış ve müşterilerin bizi terk etme durumları modellemeye çalışılmıştır. Bu çalışma, Büyük Veri analitiği ve müşteri terk modellemesi alanında yapılan araştırmalara yeni bir bakış açısı sunmayı amaçlamaktadır.

Son olarak, cilt kanseri lezyon görüntüleri makine öğrenimine hazır hale getirilmiştir. Inception V3 modellemeyle elde edilen sayısal veriler makine öğrenmesi algoritmalarıyla sınıflandırılmış ve kümelenmiştir. Bu iş için Orange programından yararlanılmıştır. Algoritmalardan elde edilen bulgular görsel olarak gösterilmiştir. Görseller açıklamalı olarak anlatılmıştır. Sonuç ve tartışma kısmında cilt kanseri görüntüleri teşhisinde algoritmaların başarısı belirtilip tartışılmıştır.

**Anahtar Sözcükler:** Büyük Veri, Makine Öğrenmesi, Tahmin, Sınıflandırma, Kümeleme

# ABSTRACT

MACHINE LEARNING ALGORITHMS IN DATA-DRIVEN DECISION MAKING

Muhammed KARA
Ondokuz Mayıs University
Institute of Graduate Studies
Department of Statistics
Ph.D., July/2024
Supervisor: Prof. Dr. Yüksel TERZİ

In this thesis, the fundamental principles, application areas, and performance comparisons of machine learning algorithms are discussed. Machine learning is a general term for algorithms that automate prediction and decision-making processes by learning from datasets. The thesis examines three main categories of machine learning methods: prediction, classification, and clustering.

The focus of the thesis is the statistical analysis of machine learning methods when making data-driven decisions and the methodologies of this analysis. Additionally, customer churn modeling was conducted as an application of these analysis methodologies, and the statistical significance of the results was noted. The methodology of the thesis is based on a mixed approach, including literature review, content analysis, and qualitative research steps. The review of literature and similar studies formed the foundation of the analyses and applications conducted.

The "automobile price" data set was used to estimate vehicle prices.. The dataset includes various physical, performance, and demographic features of cars. The data were analyzed using Linear Regression, Ridge Regression, and Lasso Regression algorithms. Analyses conducted with Orange software found an R-squared value of 0.967 for all three algorithms, while an R-squared value of 0.96 was obtained for Linear Regression on the Azure ML platform. These results indicate that the models have a high level of explanatory power. Additionally,

Another example is the WACustomerChurn dataset used in the study was introduced. This dataset consists of 21 variables and 7043 records used for customer churn modeling. The functions of the variables in the dataset were specified. big data analytics was performed on the Spark platform for customer churn modeling. Machine learning algorithms from the Spark Mlib libraries were applied using the Python programming language to model the situations in which customers might leave us. This study aims to offer a new perspective on research in the fields of big data analytics and customer churn modeling.

Finally, skin cancer lesion images were prepared for machine learning. Numerical data obtained through Inception V3 modeling were classified and clustered using machine learning algorithms. Orange software was utilized for this task. The findings from the algorithms were visually displayed, with explanations provided. In the results and discussion section, the success of the algorithms in diagnosing skin cancer images was discussed.

**Keywords:** Big Data, Machine Learning, Prediction, Classification, Clustering

# ÖN SÖZ VE TEŞEKKÜR

Bu çalışmamda, her türlü bilgi birikim ve tecrübesinden sınırsız yararlanma fırsatı bulduğum çok değerli danışmanım Prof. Dr. Yüksel TERZİ hocama çok teşekkür ediyorum.

Çalışmalarımın en başından beri desteklerini esirgemeyen, sürecin her safhasında bilgi, birikim ve tecrübesinden sınırsız bir şekilde yararlandığım Prof. Dr. Mehmet Ali CENGİZ hocama ve Dr. Öğretim Üyesi İsmail İŞERİ hocama çok teşekkür ediyorum.

Muhammed KARA

# İÇİNDEKİLER

# SİMGELER VE KISALTMALAR

| | |
|---|---|
| ANN | :Yapay Sinir Ağları (Artificial Neural Network) |
| CCM | :Araçlara ait silindir hacmi |
| CNN | :Konvolüsyonel Sinir Ağı (Convolutional Neural Network) |
| DBSCAN | :Gürültülü Uygulamaların Yoğunluk Tabanlı Mekânsal Kümelemesi (Density-Based Spatial Clustering of Applications with Noise) |
| DF | :Dermatofibroma Lezyonu |
| DVM | :Destek Vektör Makineleri (Support Vector Machines (SVMs) |
| GBM | :Gradyan Arttırma Makinesi (Gradient Boosting Machine) |
| GDELT | :Küresel veri tabanı projesi(Global Data on Events, Languages and Tone) |
| GMM | :Gaussian Karışım Modeli |
| GPU | :Grafik işlemci |
| HDFS | :Hadoop dosya sistemi |
| HPC | :Yüksek Performanslı Bilgi İşlem |
| IoT | :Nesnelerin interneti |
| KNN | :K En Yakın Komşu (K Nearest Neighbors) |
| LR | :Lojistik Regresyon |
| ML | :Machine Learning(Makine Öğrenmesi) |
| NV | :Nevus Lezyon, melanostik neviler. |
| SQL | :Yapısal sorgulama dili |
| VASC | :Vasküler Lezyon |
| YSA | :Yapay Sinir Ağları |

# ŞEKİLLER DİZİNİ

# TABLOLAR DİZİNİ

# 1. GİRİŞ

## 1.1. Genel Bilgiler

Makine öğrenmesi, modern teknolojinin en dinamik ve etkileyici alanlarından biri olarak öne çıkmaktadır. Veri hacminin hızla arttığı günümüzde, bu verilerden anlamlı ve değerli bilgiler elde etmek her zamankinden daha önemli hale gelmiştir. Makine öğrenmesi, veri setlerinden öğrenerek tahmin ve karar verme süreçlerini otomatikleştiren algoritmaların genel adıdır. Bu teknolojinin önemi, çeşitli sektörlerdeki geniş uygulama alanlarından kaynaklanmaktadır. Sağlık hizmetlerinden finansal modellere, perakende analizlerinden otonom araç teknolojilerine kadar birçok alanda makine öğrenmesi kullanılmaktadır.

Büyük veri, geleneksel veri işleme ve analiz araçları ile yönetilemeyecek kadar büyük, karmaşık ve çeşitli veri setlerini ifade eder. Büyük veri genellikle üç V kavramıyla tanımlanır: hacim (volume), çeşitlilik (variety) ve hız (velocity). Hacim, çok büyük miktarda veriyi işaret eder; çeşitlilik, farklı veri türlerini ve kaynakları ifade eder; hız ise bu verilerin hızlı bir şekilde üretilip işlenmesi gereğini vurgular.

Büyük veri; sosyal medya, sensör ağları, mobil cihazlar, web trafiği, coğrafi konum ve diğer birçok kaynaktan elde edilen verileri içerebilir. Bu verileri analiz etmek; desenleri, trendleri, müşteri davranışlarını ve diğer önemli bilgileri belirlemek için büyük veri analitiği teknikleri ve araçları kullanılır.

Büyük verinin sınıflandırılmasında ve kümelenmesinde kullanılan spark platformunun avantajlı yönleri son yıllarda yapılan çalışmalarda açıkça görülmektedir. Bu çalışmalara görüntü verilerinin de eklenmesi ve bu görüntülerin analizinde makine öğrenmesiyle sınıflandırma ve kümeleme algoritmalarının kullanılması büyük fırsatlar sunmaktadır.

Araç fiyat tahmini, otomotiv endüstrisi ve ikinci el araç piyasası için büyük önem taşımaktadır. Doğru fiyat tahminleri, satıcıların ve alıcıların bilinçli kararlar almasına yardımcı olabilir. Tezdeki ilk uygulamada, araç fiyatlarının tahmin edilmesi amacıyla "Automobile Price" veri seti kullanılmıştır. Veri seti, araçların çeşitli fiziksel, performans ve demografik özelliklerini içermektedir. Bağımlı değişken olarak "price" kullanılmıştır. Price değişkeni, aracın motor hacmi, silindir çapı, pistonun silindir içinde hareket ettiği mesafe, motorun sıkıştırma oranı, motorun beygir gücü, motorun maksimum devir hızı, şehir içi ve şehir dışı yakıt tüketimi, belirli bir araç modelinin

sigorta tazminat ödemelerine göre normalleştirilmiş kayıp oranı ve sigorta risk derecelendirmesi gibi özelliklerin kombinasyonu ile tahmin edilmiştir. Bağımsız değişkenler arasında aracın uzunluğu, genişliği, yüksekliği, aks mesafesi, boş ağırlığı, markası, yakıt türü, motor tipi ve silindir sayısı gibi özellikler bulunmaktadır. Bu özelliklerin kullanımıyla, Linear Regresyon, Ridge Regresyon ve Lasso Regresyon algoritmalarıyla veri analiz edilmiştir. Orange programı kullanılarak yapılan analizlerde üç algoritma için de R-kare değeri 0,967 bulunmuş, Azure ML platformunda ise Linear Regresyon için R-kare değeri 0,96 olarak elde edilmiştir. Bu sonuçlar, modellerin yüksek açıklayıcılık düzeyine sahip olduğunu göstermektedir.

Tezdeki ikinci uygulamada müşteri terk modelleri için makine öğrenmesi algoritmalarının kullanılması işlenmiştir. Büyük veri ve makine öğrenimi tekniklerinin kullanımıyla, endüstriler müşteri davranışlarını daha iyi anlayabilir ve bu anlayışı kullanarak daha etkili müşteri ilişkileri yönetimi stratejileri oluşturabilirler. Bu nedenle, müşteri terkini öngörme konusundaki bu tür çalışmalar, işletmeler için değerli bir kaynak niteliğindedir.

Son uygulama cilt kanseri, etkili tedavi ve hasta refahı için doğru ve zamanında teşhis gerektiren, artan bir insidans ile büyüyen bir küresel sağlık sorununu temsil etmektedir. İleri teknolojilerin ve makine öğrenimi tekniklerinin ortaya çıkmasıyla birlikte dermatoloji alanı önemli bir dönüşüm geçirmiştir. Bu durum, cilt kanseri lezyonlarının daha doğru ve verimli bir şekilde analiz edilmesini sağlamıştır.

Bu tezde, regresyon, sınıflandırma ve kümeleme algoritmalarının merceğinden cilt kanseri görüntü analizi dünyasını keşfetmek amaçlanmıştır. Bu uygulamaların ve teşhis doğruluğunu artırma potansiyellerinin kapsamlı bir karşılaştırma analizi yapılmıştır. Dünyada en sık görülen kanser türlerinden biri olan cilt kanseri, melanom, skuamöz hücreli karsinom ve bazal hücreli karsinom gibi çeşitli alt tipleri içerir. Bu lezyonların erken teşhis edilmesi ve özellikle iyi huylu ve kötü huylu vakalar arasında ayrım yapılması, hastalığın başarılı bir şekilde tedavi edilmesi ve yönetilmesinde çok önemli bir rol oynamaktadır. Bu amaçla, gelişmiş makine öğrenimi algoritmaları dermatologların cephaneliğinde güçlü araçlar olarak ortaya çıkmıştır. Bu bağlamda, özellikle konvolüsyonel sinir ağları (CNN'ler), destek vektör makineleri (DVM), karar ağaçları ve k-en yakın komşular (KNN) gibi sınıflandırma algoritmalarına odaklanarak; makine öğrenimi alanı araştırılmıştır. Bu algoritmalar cilt lezyonlarının doğru bir şekilde kategorize edilmesinde hayati bir rol oynamaktadır. Bu

algoritmaları duyarlılık, özgüllük ve genel doğruluklarına göre değerlendirerek; cilt kanserini teşhis etme gibi karmaşık bir görev için uygunluklarını belirlemeye çalışılmıştır. Tezde sınıflandırmanın ötesinde K-means, hiyerarşik kümeleme ve DBSCAN gibi kümeleme algoritmaları da belirtilmiştir. Bu algoritmalar, cilt kanseri görüntülerinden oluşan veri kümelerindeki örüntüleri tanımlama potansiyelleri açısından araştırılmakta ve sonuçta hastalığın karmaşıklığını anlamak için tamamlayıcı bir yaklaşım sunmaktadır. Kümeleme teknikleri, benzer cilt lezyonlarını gruplama fırsatı sunarak dermatologlara büyük görüntü veri kümelerindeki ortak özellikleri ve anormallikleri tanımlamada yardımcı olur.

### 1.2. Literatür Araştırması

Son yıllarda oylama sınıflandırıcıları, hastalıkların tanı ve teşhisinde daha hızlı hareket etmek adına sıklıkla kullanılan yöntemler olmuşlardır. Kumar ve ark. (2023), Wisconsin Üniversitesinden aldıkları verilerle, iyi huylu ve kötü huylu tümörün sınıflandırılması için çeşitli veri madenciliği tekniklerini kullanmışlardır. Bilgisayarlı Göğüs Tomografisi (BT) taramaları da birtakım hastalıkların tanısında oldukça önemli rol oynar. Hâlihazırda gündemde olan COVID-19 tanı ve teşhisinde BT görüntülerini işlemek ise oldukça pahalı bir iştir. Makine Öğrenimi Teknikleri bu zorluğun üstesinden gelme potansiyeline sahiptir.

“Predicting Customer Churn in the Telecommunication Industry Using Machine Learning Techniques" adlı çalışmada, büyük veri analitiği ve makine öğrenimi teknikleri kullanılarak telekomünikasyon endüstrisinde müşteri terkini öngörme üzerine odaklanılmıştır. Çalışma, müşteri terkini etkileyen faktörleri tanımlamak ve bu faktörleri kullanarak terk eden müşterileri belirlemek için çeşitli makine öğrenimi algoritmalarını değerlendirir (John et al., 2018).

"Customer Churn Prediction in Banking Industry Using Big Data Analytics" adlı makalede, bankacılık endüstrisinde büyük veri analitiği kullanılarak müşteri terkini öngörme üzerine odaklanılmıştır. Yazarlar, müşteri terkini etkileyen faktörleri belirlemek için çeşitli büyük veri tekniklerini kullanarak müşteri terkini tahmin etmek için makine öğrenimi modelleri geliştirmiştir (Smith ve Johnson, 2019).

"Predicting Customer Churn in E-commerce Industry Using Data Mining Techniques” adlı çalışma, e-ticaret endüstrisinde müşteri terkini öngörme üzerine odaklanır ve veri madenciliği tekniklerini kullanarak müşteri terkini etkileyen

faktörleri ve davranışları tanımlar. Çalışma, müşteri terkini azaltmak için öneriler sunar ve doğru tahminler elde etmek için farklı veri madenciliği algoritmalarının karşılaştırılmasını içerir (Lee ve Kim, 2020).

"Predicting Customer Churn in the Insurance Industry Using Big Data Analytics" adlı çalışmada, sigorta endüstrisinde müşteri terkini öngörme üzerine odaklanılmıştır. Yazarlar, büyük veri analitiği tekniklerini kullanarak müşteri terkini etkileyen faktörleri belirler ve makine öğrenimi modelleri kullanarak müşteri terkini tahmin etme yeteneğini değerlendirir (Brown ve White, 2021).

Sengupta (2016) büyük veriler için önyükleme yöntemleri ile ilgili bir çalışma yapmıştır. Khalifa (2017) yaptığı çalışmada Big Data hizmetlerini geliştirme konusuna bir fırsat olarak baktıklarını ifade etmiştir. Bu anlayışlar farklı veri madenciliği tekniklerinin kullanılmasını gerektirir. WEKA gibi olgun veri madenciliği araçları veya R programlama dili yıllardır geliştirilmektedir. Gelişmiş analitik yöntemlerle desteklenen çok sayıda veri madenciliği algoritmaları uygulanmıştır. Fakat bu olgun araçlar tek bir makinede çalışacak şekilde tasarlanmıştır.

2018 yılında Paulson tarafından yazılan tezde, Büyük Verinin artık yaygın olduğunu vurgulanmıştır. Büyük verilerin depolanması ve işlenmesi için yeni yöntemler geliştirmenin kritik bir ihtiyaç oluşturduğu ifade edilmiştir. Bu çalışmada ilk olarak, büyük ölçekli veri analizi yöntemleri incelenmiştir. Özellikle, popüler MapReduce(MR) modelini paralel ilişkisel veri tabanı yönetim sistemleriyle karşılaştırıp güçlü ve zayıf yönleri ampirik olarak analiz edilmiştir. İkinci katkıda, MapReduce gibi Büyük Veri ölçekleme yöntemlerinin ölçeklenebilir ve esnek bir bulut tabanlı varlık eşleştirme uygulamaları oluşturmak için nasıl kullanılabileceğini ve benzer uygulamaların gelecekteki gelişimi için bu çabadan hangi derslerin çıkarılabileceği belirtilmiştir.

2017’de Özcan tarafından yazılan tez, finansal piyasalarda yatırımcı türü ve duyarlılık oyununun rolüne odaklanmaktadır. İlk kısım varlık arasındaki etkileşimin etkisini araştırmaktadır. Farklı likidite tercihlerine sahip yatırımcıların varlık seçimlerini belirlemek için bir varlık arama modeli oluşturulmuştur. İkinci kısım, duygu temelli bir konu kümeleme algoritmasıyla Twitter'da trend olan konuları kullanarak gün içi döviz kurları tahmin edilmiştir.

Li (2016) tarafından yazılan tezde, analitik ve bilimsel algoritmalar gibi bilgi işlem yoğun iş yüklerinin, geleneksel Yüksek Performanslı Bilgi İşlem (HPC) ve Büyük Veri platformlarına yeni zorluklar getirdiği ifade edilmiştir. HPC platformu daha iyi performans ve güç verimliliği elde etmek için heterojen bilgi işlem kaynaklarını entegre eder. Bununla birlikte, heterojen mimariler şeffaf ivme ile kaynakların çekirdeklere ve hızlandırıcılara tahsis edilmesiyle ilgili zorluklarla karşı karşıyadır. MapReduce ve Apache Spark gibi popüler Büyük Veri platformları veri ve hesaplama dağıtımı ve bellek içi önbellekleme ile büyük veri zorluklarını giderir. Ayrıca HPC ve Büyük Veri platformlarındaki iş yüklerinin dinamikleri ve çeşitleri nedeniyle, heterojen kaynaklara karar vermek üzere gelen uygulamaları tahmin etmek zordur. Bu tezde, hem mikro düzeyde HPC Çip Üzerine Sistem (SoC) hem de makro düzeydeki Büyük Veri kümeleri için heterojen mimari kullanımı araştırılmıştır. Öngörülemeyen iş yüklerinin hızlandırılmasını destekleyen yeniden yapılandırılabilir mantıktan oluşan, çalışma zamanı yeniden programlanabilir heterojen bir SoC mimarisi olan "Transformer" önerilmiştir. Mimari, bir veya daha fazla hızlandırma fonksiyonunun çalışma zamanı somutlaştırılmasına izin verir ve böylece daha yüksek performans ve enerji verimliliği sağlar. Daha sonra, Apache Spark ile entegre GPU(Grafik işlemci) hızlandırmalı heterojen bir mimari olan "HeteroSpark" sunulmuştur. Bu, GPU'ların büyük işlem gücünü ve CPU(İşlemci)'ların ölçeklenebilirliğini ve hem veri hem de yoğun işlem gerektiren uygulamalar için sistem belleği kaynaklarını birleştirir. Son olarak, optimum performans için görevleri heterojen bilgi işlem kaynaklarına uyarlanabilir şekilde zamanlamak için düşük dinamik profil oluşturma teknikleri kullanan genel heterojen yük dengeleyici "Sparkling +" uygulanmıştır.

Zeng (2017) tarafından yazılan tezde, seyrek makine öğrenme modellerinin, yüksek boyutlu verilerin analizinde giderek daha popüler hale geldiği belirtilmiştir. Gelişmekte olan Büyük Veri dönemi ile ultra yüksek boyutlu, büyük ölçekli veri setleri genetik, genomik, biyomedikal görüntüleme, sosyal medya analizi ve yüksek frekans finansı gibi birçok alanda sürekli olarak toplanmaktadır. Bu tez, Büyük Veri analizini kolaylaştırmak için ölçeklenebilir seyrek makine öğrenme yöntemlerinin geliştirilmesine odaklanmaktadır.

2016'da Chen tarafından yazılan tez, statik ve akış kaynağının Büyük Veri analitiğine odaklanmaktadır. Statik kaynağın yerinde sorgulanması için filtreler ve ön

işlemeyen bir dilimleme tekniği geliştirir. Kaynak verilerinin yüksek alım hızında çevrimiçi işlenmesi için bir akış işleme çerçevesi sunar. İlk olarak, uygulama başlangıcından önce verilen sorguları sorgulamak için yeterli olsa da (ileri sorgulamalar), ikincisi, hedefleri önceden bilinmeyen sorgularla (geriye doğru sorgulamalar) ilgilenir. Son olarak, geniş çaplı yayılımlar üzerine veri madenciliğini araştırır ve kümelenme, sınıflandırma ve ilişkilendirme kuralları oluşturma gibi madencilik görevlerini etkin bir şekilde desteklerken, yüksek boyutsallığı azaltabilen, kaynağın geçici bir temsilini önerir ve zamansal temsil akışına da uygulanabilir. Önerilen teknikler; bilgisayar ağı verilerinden, hava modellerinden, okyanus modellerinden, uzak (uydu) görüntü verilerinden ve tarımsal karar verme tabanlı aracı simülasyonlardan yakalanan Büyük Veri kaynağına uygulanan yazılım prototipleri ile doğrulanmaktadır.

Princeton University 2017'de, Zhao tarafından yazılan tez, büyük veri analizine özgü modern istatistiksel prosedürlerin belirsizliğini değerlendirmek için yeni çıkarımsal yöntemler ve teori geliştirmektedir. Özellikle, büyük verilerin dört zorlu yönüne odaklanmıştır: büyük örnek boyutu, yüksek boyutsallık, heterojenlik ve karmaşıklık. Başlangıç olarak, kısmen büyük heterojen verilerin modellenmesi için doğrusal çerçeve düşünülmüştür. Ana hedef, her bir alt popülasyonun heterojenliğini araştırırken tüm alt popülasyonlarda ortak özellikler elde etmektir.

Carvalho (2017) Nesnelerin İnterneti ve evrimi ile Büyük Veri mimarilerinin veri depolaması gerekliliğinin nedeni ve zorlukları hakkında genel bir bakış sunmayı amaçlamaktadır. Odak noktası Nesnelerin İnterneti ve Büyük Veri'nin geleceği hakkında fikir vermektir.

Ghahramani vd., (2019)'da yayınlanan makalede, işlenmiş büyük ölçekli heterojen verilerin yüksek yüzdesi ve teknolojisinin, büyük veri sorununun en belirgin nedeni olduğunu belirtmişlerdir. Günümüzde veri genişletmenin sonuçları farklı alanlarda gösterilmektedir. Kullanıcıların içeriksel verileri, mühendislik ve işletme alanlarında olduğu kadar ulaşım, yere dayalı hizmetler ve reklam endüstrisi için de değerlidir. Cep telefonlarını örnek olarak alalım. Dünya çapında milyarlarca abonelik var ve sensör cihazları insanların etkileşimlerini dijitalleştiriyor. Cep telefonlarının ürettiği veri boyutu, gerçeklere dayalı ve gerçek zamanlı kararlar verme ihtiyacı, araştırmacıların karşılaştığı zorluklardır. Son zamanlarda, büyük miktarda veriyi işlemek ve analiz etmek için bulut bilişim tabanlı yeni teknolojiler ortaya çıkmıştır.

Bir telekomünikasyon şirketi ile yapılan işbirliği ile çağrı detay kayıtlarının analizi için bu teknolojiler kullanılmıştır. Bir keşifsel uzamsal veri analiz algoritması ve analiz sonuçları sunulmuştur. Farklı alanlara öncelik vermek, sıcak noktaları hızlı ve doğru bir şekilde tespit etmek hedeflenmiştir. Bu araştırma çalışmasının bulguları, kentsel planlama ve geliştirme ile telekomünikasyon altyapısının iyileştirilmesine yardımcı olabilir.

Hui Liu vd., (2019) yaptıkları çalışmada, iç mekandaki mobil robot pozisyonu için tahminin, robot navigasyon sisteminin güvenliğini ve çalışmasını etkileyen önemli bir faktör olduğunu ifade etmişlerdir. Mobil robot pozisyonu büyük verileri ile birleştirildiğinde, mobil robot pozisyonunu tahmin etmek daha güvenilir ve sağlam olacaktır. Bu yazıda, iç mekân mobil robotu için hibrit bir konum büyük veri tahmin modeli önerilmiştir: Apache Spark platformunda EWT-GB-DT-IEWT yöntemleri üzerinde durulmuştur. Önerilen pozisyon büyük veri tahmin modeli temel olarak, ayrışma önişleminden, büyük veri tahmin yönteminden ve yeniden yapılandırma sonrası işlemden oluşmaktadır. EWT (Ampirik Dalgacık Dönüşümü) yöntemi, başlangıç konum verilerinin altkümelere ayrıştırılması için kullanılır. GB (Gradient Boosting) ile geliştirilen DT yöntemi (Karar Ağacı), Apache Spark üzerindeki her bir alt satırı tahmin etmek için uygulanır. Farklı serilerin öngörme sonuçları, IEWT (Ters Ampirik Dalgacık Dönüşümü) sonrası işleme ile yeniden oluşturulur. Sonuçlar, önerilen hibrid konum büyük veri tahmin yönteminin konum yörüngesini doğru bir şekilde tahmin edebileceğini göstermektedir. Spark için önerilen konum büyük veri tahmin modeli, iç mekân robot navigasyon sisteminin güvenilirliğini ve istikrarını artırabilir.

Aftab ve Siddiqui (2018) tarafından yapılan çalışmada dünya teknolojisindeki dinamik değişikliklerle birlikte sosyal medya, bilgisayar ağları, nesnelerin interneti ve bulut bilişimin kullanımında artan bir büyüme ve benimsenme mevcuttur, denilmiştir. Araştırma deneyleri ayrıca toplanacak, yönetilecek ve analiz edilecek büyük miktarda veri üretmektedir. Bu büyük veri "Büyük Veri" olarak bilinir. Araştırma analistleri, hem yararlı hem de yararsız bilgileri içeren verilerde bir artış algıladılar. Yararlı bilgilerin çıkarılmasında, veri ambarı üretilen veri miktarının artmasıyla devam etmekte zorlanır. Paradigmada meydana gelen değişimlerle birlikte büyük veri analizi, karar verme açısından iş zekâsını destekleyen umut verici bir araştırma alanı olarak ortaya çıkmıştır. Bu makale büyük veri, büyük veri sorunları, büyük veri analitiği ve

büyük veri ambarı hakkında kapsamlı bir anket sunmaktadır. Ayrıca, büyük veri ve veri ambarının artırılması ihtiyacının, karar verme, yöntemlerin ve araştırma problemlerinin karşılaştırılması perspektifinde nasıl ortaya çıktığını da açıklamaktadır. Ayrıca büyük veriyi, veri ambarını ve zorluklarını destekleyen uygulamaları da inceler.

Shah tarafından 2019'da yapılan çalışmada, afet yönetimi süreçlerinin temel işlemlerini gerçekleştirmek için yeni ve daha etkili bir yaklaşım sunan BDA(Büyük veri analitiği) teknolojilerinin birleşmesi için yeni bir kavramsal bir çerçeve önerilmiştir. Çalışmada, bazı önerilen parametrelerle birleştirilen büyük veri teknolojisinin, afet yöneticilerine güncel ve yararlı bilgiler sağlamak için kriz verilerini üretmek, entegre etmek, işlemek ve analiz etmek için etkili bir şekilde kullanılabileceği üzerinde durulmuştur. Büyük veri kaynaklarının (yani, IoT tabanlı sensörler, sosyal medya, kalabalık kaynaklı çevrimiçi haritalama) entegrasyonu ve işlenmesi daha etkili ama aynı zamanda çok zorlu bir ortama yol açabilir. Bu tez, farklı türdeki büyük verilerin yönetime alınması ve konuyla ilgili literatür ışığında birleştirilmesi için kullanılabilecek yeni bir mimarinin hayata geçirilmesine katkıda bulunmaktadır. Bu tez çalışmasında, sosyal medya verilerinin kalitesi önerilen bir filtreleme mekanizması ile değerlendirilmiştir. Tez, konuşlandırılmış sistemde gerçekleştirilen tüm operasyonel adımların detaylarını gösteren uygulama modeline odaklanmaktadır. Önerilen uygulama modeli dört katmana ayrılmaktadır. Bunlar: 1) Veri Toplama, 2) Veri Yığma, 3) Veri Ön İşleme, 4) Büyük Veri Analitiği ve Servis Platformu. Spark Engine ve Hadoop Ecosystem ile donatılmış sistem platformu, verileri öngörülen algoritmalara göre işler. Uygulama, MapReduce mekanizmasıyla Hadoop ekosistemi kullanılarak gerçekleştirilir. MapReduce'un paralel oluşumu HDFS ile konuşlandırılmıştır. Apache Spark, gerçek zamanlı veri akışlarında daha güçlü işlemler için Hadoop ile birlikte kullanılmıştır. Hem çevrimiçi hem de çevrimdışı veri akışlarını destekleyen Spark Streaming, sistemde veri toplaması için dağıtılır. Uygulanan sistem, Hadoop üzerinden paralel veri işlemeden ve Apache Spark kullanarak gerçek zamanlı veri işlemeden faydalanmaktadır. Bu kombinasyon esnek ve etkili depolama, doğru parametre hesaplama ve hızlı sonuç üretmeyi sağlamıştır. Hadoop Ekosistemi ve Spark tabanlı analitik, IoT ve Twitter veri setleri için gerçek zamanlı ve çevrimdışı analizleri değerlendirmek üzere gerçekleştirilir. Önerilen çerçeve, gerçek zamanlı işlem gerektiren büyük veri kümelerinin işlenmesine

odaklanmıştır. Bu nedenle uygulanan sistem artan veri büyüklüğü dikkate alınarak veri işleme ve verim açısından değerlendirilmiştir. Veri filtreleme ve normalleştirme teknikleri, işlem süresini yeterince düşürmüş ve verimi artırmıştır. Çalışma, çeşitli şemaların performansını karşılaştırmak için Apache Spark'ın farklı durumlarıyla birlikte tek ve çift düğümlü MapReduce Hadoop küme örneklerini genel ve filtrelenmiş veri kümeleriyle değerlendirmiştir. Sistem verimliliğinin değerlendirilmesi, önerilen mimarinin performans üstünlüğünü gösteren işlem süresi ve verim açısından ölçülmüştür. Bu tez, akıllı şehirler ve afet yönetimi alanındaki gelecekteki satın alımlar için araştırmacılara ve endüstrilere referanslar sağlayabilir.

Khan tarafından 2017'de yazılan çalışmada gerçek yaşam uyarlamalı sinyal için çevrimiçi doğrusal olmayan öğrenme araştırılmış, büyük veri içeren işlem ve makine öğrenme uygulamaları, etkili algoritmalar için yeni çözümler sunulmuştur. Ayrıca önerilen çevrimiçi doğrusal olmayan öğrenme modellerinin performansı, uzmanların çeşitli yöntemleri ve güçlendirme kavramı ile geliştirilmiştir. Önerilen algoritmalar, gerçek yaşam koşullarında önemli ölçüde azaltılmış hesaplama karmaşıklığı ve depolama gereksinimi ile en son teknolojik yöntemler üzerinde önemli bir performans kazancı elde etmiştir.

Çelik tarafından 2018'de yazılan tezde, artık büyük veri kavramının sanal ve yerel ortamlarda karşımıza çıkan ve bizim ürettiğimiz verilerin çokluğundan dolayı hayatımızın vazgeçilmezi haline geldiği ifade edilmiştir. Eğer bu veriler uygun yöntem ve tekniklerle işlenirse hayatımızda anlamlı kolaylıklar sağlayacaktır. Çelik'in bu çalışmasında 1979'dan günümüze kadar dünyadaki olayları tutan bir veri tabanı olan GDELT' vurgu yapılmıştır. Bu veri tabanındaki bilgiler SQL sorgu ile alınarak dünyadaki çatışma durumu araştırılmıştır. Sadece dünyadaki çatışma durumları değil Türkiye'deki ve Ukrayna'daki protesto ve gerilimler de analiz edilmiştir.

Zebari'nin (2023) "Ülkelere Göre Sosyal Medyada Büyük Veri Kullanarak Covid-19 Gündeminin R Programı İle Analizi" isimli doktora tezinde; twitter'da covid19 konusu ile ilgili Irak'tan sağlanan mesajlar ile bunun yanısıra seçilen diğer birkaç ülkeden atılan tweetler belirlenmiş ve bu tweetler arasındaki ilişkiler incelenmiştir. İnceleme sırasında Büyük Veri, İçerik, Korelasyon, Frekans ve Çapraz Gecikmeli Korelasyon Analizlerinden yararlanılmıştır. Tüm bu analizlerin sonunda; covid19 tweetlerinin herkesin yazdığı genel tweetlere oranla daha düşük oranda tercih

edildiği görülmüştür. Bunu yanı sıra covid19 konulu tweet ve retweetlerin diğer tweet ve retweetlere göre daha yüksek sayıda olduğu sonucuna varılmıştır.

# 2. MATERYAL VE YÖNTEM

## 2.1. Büyük Veri

Büyük veri (Big Data), genellikle geleneksel veri yönetim araçları ile işlenemeyen büyük hacimde, yüksek çeşitlilikte ve yüksek hızda veri akışlarını ifade eder. Bu veri, genellikle yapısal olmayan veya yapısal veri türlerini içerir ve çeşitli kaynaklardan gelir. Örneğin sensörler, sosyal medya, web trafiği, mobil cihazlar ve diğer dijital kaynaklar…

Büyük veri, işletmelerin ve kuruluşların veri tabanlı kararlar almasına, yeni fırsatları keşfetmesine ve rekabet avantajı elde etmesine olanak tanır. Ancak, bu büyük veri kütlelerini etkili bir şekilde işlemek için özel veri depolama, analiz ve işleme araçları gerekebilir.

### 2.1.1. Büyük Veri Bileşenleri

Büyük veri, "3V" olarak adlandırılan üç temel özelliği ile tanımlanır:

Hacim (Volume): Geleneksel yöntemlerle saklanamayacak kadar büyük olan verinin boyutudur. Büyük veri, milyarlarca veya trilyonlarca kayıttan oluşabilir ve geleneksel veri tabanları ile işlenmekte zorlanabilir.

Çeşitlilik (Variety): Büyük veri setlerinde bulunan ve çeşitli kaynaklardan, farklı yapısal veya yapısal olmayan veri türlerinden (metin, resim, ses ve video gibi) ve çeşitli veri formatlarından oluşan verinin çeşitliliğidir.

Hız (Velocity): Büyük veri hızlı bir şekilde üretilir, işlenir ve analiz edilir. Bu veri, gerçek zamanlı veya yakın gerçek zamanlı analiz gerektirebilir. Büyük veri setlerindeki veri hacminin hızla artmaya devam etmesidir.

Büyük veri ayrıca aşağıdaki bileşenleri de içerir.

Değer (Value): Çeşitli organizasyonlar için önemli hale gelen veriden elde edilen anlamdır. Değer üretmeyen veri anlamsızdır.

Doğruluk (Accuracy): Büyük veriye özgü teknolojilerle verinin doğruluğunun ve analize uygunluğunun sağlanması gereklidir.

## 2.2. Büyük Veri Analitiği

Büyük veri analitiği (Big Data Analytics), verilerde yapılan işlemlerle ilgili yapılacakları belirlemek üzere büyük hacimdeki işlenmemiş verilerin yönelimlerini

meydana çıkarma aşamalarını gösterir. Bu aşamalar, sınıflandırma, regresyon ve kümeleme gibi farklı istatistiksel yöntemleri içine alır. Büyük veri analitiği, veri bilimi, istatistik, makine öğrenimi ve veri madenciliği gibi alanlardan gelen teknikleri ve araçları kullanarak bu veri setlerinden anlamlı desenler, trendler, ilişkiler ve fırsatlar bulmayı amaçlar.

Büyük veri analitiği, işletmelerin karar verme süreçlerini iyileştirmek, müşteri davranışlarını anlamak, pazarlama stratejilerini optimize etmek, operasyonel verimliliği artırmak ve rekabet avantajı elde etmek için kullanılır. Örnek uygulamalar arasında müşteri segmentasyonu, tahmin analizi, dolandırıcılık tespiti, talep tahmini ve kişiselleştirilmiş öneri sistemleri bulunmaktadır.

Büyük veri kavramı ve uygulanan analitik yöntemler son 25 yıldır ele alınmaktadır. Özellikle akıllı alışveriş mağazaları, internet destekli alışveriş sistemlerinin artmasıyla büyük veri kullanımı ve analizi karar verme süreçlerinde etkin bir rol oynamaya başlamıştır. Bu denli büyük verilerin saklanması ve işleme alınması için farklı projeler geliştirilmiştir.

Veri bilimi ile uğraşan profesyoneller internet, algılayıcı, akıllı cihazlar ile üretilen ve elde edilen bu devasa veriyi karmaşıklıktan kurtarıp karar verme ve topluma katkı sunabilecek süreçlere dahil edebilmenin yollarını aramaktadırlar. İşte bu yollardan en önemlisi büyük veri analitiğinde makine öğrenmesi yöntemlerinin kullanılması olmuştur.

### 2.2.1. Büyük Veri Teknolojileri

#### 2.2.1.1. Bulut Ortamı

Bir internet ağı üzerinden sunulan, online olarak ağa ulaşabilen yetkilendirilmiş kullanıcılara açık olan geniş kapsamlı verilerin kullanıma sunulduğu ortamlardır. Bulut ortamı, internet üzerinden sunulan ve pay-per-use (kullanım başına ödeme) modeliyle çalışan bilişim kaynaklarını sağlayan bir hizmettir. Büyük veri işleme ve depolama için bulut bilişim hizmetleri, büyük miktarda veriyi depolamak, işlemek ve analiz etmek için ölçeklenebilir ve esnek bir altyapı sağlar. Önde gelen bulut sağlayıcıları, büyük veri teknolojilerini bulut ortamında sunmak için hizmetler sunmaktadır.

Bulut tabanlı büyük veri çözümleri, maliyet etkinliği, ölçeklenebilirlik, esneklik ve erişilebilirlik gibi avantajlar sağlar. Bu nedenle, birçok işletme ve kuruluş, büyük

veri analitiği projelerini bulut ortamında yürütmeyi tercih etmektedir. Bu yaklaşım, işletmelere büyük veri işleme ve analizi için gerekli altyapıyı sağlamanın yanı sıra, yatırım maliyetlerini azaltma ve işletme sürekliliğini artırma avantajı sunar.

#### 2.2.1.2. Google Dosya Sistemi

Google Dosya Sistemi (Google File System-GFS) Google tarafından geliştirilen ve büyük ölçekli dağıtık dosya depolama sistemi olan bir dosya sistemi teknolojisidir. Bu sistem, büyük ölçekli veri depolama ihtiyaçlarını karşılamak üzere tasarlanmıştır ve Google'ın arama, reklam ve diğer hizmetlerinde kullanılmaktadır. Google Dosya Sistemi, yüksek oranda paralel erişim, dayanıklılık, yüksek hızlı veri aktarımı ve otomatik veri yedekleme gibi özellikler sunar. Bu, Google'ın geniş ölçekteki veri depolama ve işleme ihtiyaçlarını karşılamasına yardımcı olur.

#### 2.2.1.3. Hadoop

Java programlama dili ile geliştirilmiştir. Büyük veri işlemek için kullanılan bir araçtır. Cloudera mimarı Apache Software Foundation başkanı Doug Cutting tarafından geliştirilmiştir. Farklı ortamlardan gelen verilerin hızla artmasıyla bilinen veri işleme yöntemlerinin yetersizliğinin ortaya çıkması sonucu bir ihtiyaç olarak ortaya çıkarılmıştır. Yüzde yüz açık kaynaklı veri depolama ve işleme biçiminin öncüsüdür. Geçmişten günümüze kadar kullanılagelen hiç de ucuz olmayan sistemlere nazaran Hadoop, büyük verilerin dağıtık olarak paralel bir şekilde işlemeyi düşük bir maliyetle yaparak büyük veri analitiğinde sunucu, saklama ve işleme alanlarının verimli bir şekilde kullanılmasını gerçekleştirir. Değişik özelliklere sahip veri türleri gerek yapılandırılmış gerek yapılandırılmamış olsun başka veri depolarından gelmiş olsalar bile Hadoop bunları şemaları olmadan depolama özelliğine sahiptir.

Hadoop Dağıtık Dosya Sistemi (Hadoop Distributed File System-HDFS) sıradan sunucu kümelerinde yer alan büyük veri işlemlerini yapabilmek için kullanılan dağıtık bir dosya sistemidir. Hadoop MapReduce büyük verilerde uygulamalarda yapılan analizlerdeki iş yükünün birden fazla iş istasyonuna yönlendirilip iş yükünün paylaştırılmasına imkân sağlayan açık kaynaklı bir kütüphane olarak ifade edilebilir. Hadoop, farklı veri analitiği sistemlerine göre bir çatı görevi üstlenir. Hadoop ile birlikte, Euro (veri serileştirme sistemi), Cassandra (yüksek erişilebilirlikli, ölçeklenebilir NoSQL veritabanı), HBase (büyük veri üzerinde iş zekâsı sistemi, ölçeklenebilir, dağıtılmış NoSQL veritabanı), Hive (büyük veri üzerinde iş zekâsı

sistemi), Mahout (ölçeklenebilir yapay öğrenme ve veri madenciliği kütüphanesi), Pig (paralel hesaplamalar için yüksek seviye veri akış dili ve yürütme kütüphanesi), ZooKeeper (dağıtık uygulamalar için yüksek ölçekli koordinasyon uygulaması) gibi veri analitiğinde büyük katkıları olan yapılar geliştirilmektedir.

#### 2.2.1.4. Map Reduce

Veri kümelerindeki işleyişin anlaşılmasını ve çözülmesini sağlayan bir araçtır. 2004 yılında Google tanımlamıştır. Bu sayede binden fazla düğüm tarafından iş yükleri paylaşılmış oldu. Map-Reduce Eşle-İndirge mantığına dayanır. Yani büyük bir veri analizini daha küçük parçalara bölebilir. Bunu SQL araçlarıyla bile yapabilir. Büyük veri kümelerini daha küçük parçalara ayırarak dağıtıp böylece birden fazla iş biriminin veri analiz yükünü paylaşarak sistemi rahatlatmayı hedef alır. Çünkü bu büyük veri kümeleri birden fazla bilgisayarın üstesinden gelemeyeceği zorluktadır. Burada veri kümeleri iki aşamada işlem görür. Map aşaması; isim ve değer ikililerini sisteme girdi olarak alır ve işlemi gerçekleştirir. Girdideki isim ile birlikte üretilen çıktıyı sonuç listesi olarak verir. Reduce aşaması, Map aşamasındaki çıktıları birleştirir ve hepsini tek bir sonuca çevirir. Böylece girdilere ait işlenmiş çıktıları tek bir sonuç listesine indirgemiş olur.

#### 2.2.1.5. Spark MLib

MLlib, Spark'ın makine öğrenimi (ML) kitaplığıdır. Amacı, pratik makine öğrenimini ölçeklenebilir ve kolay hale getirmektir. Yüksek düzeyde, aşağıdaki gibi araçlar sağlar:

*ML Algoritmaları:* Sınıflandırma, regresyon, kümeleme ve işbirlikçi filtreleme gibi yaygın öğrenme algoritmaları

*Özellikleştirme:* Özellik çıkarma, dönüştürme, boyutluluk azaltma ve seçme

*Pipelines:* ML Pipelines oluşturmak, değerlendirmek ve ayarlamak için araçlar

*Kalıcılık:* Algoritmaları, modelleri ve Ardışık Hatları kaydetme ve yükleme

*Yardımcı programlar:* Lineer cebir, istatistik, veri işleme vb.

MLlib birçok algoritma ve yardımcı program içerir. ML algoritmaları şunları içerir:

- *Sınıflandırma:* Lojistik regresyon, NaiveBayes,

- *Regresyon:* Genelleştirilmiş doğrusal regresyon, hayatta kalma regresyonu,
- Karar ağaçları: Rastgele ormanlar ve gradyanla güçlendirilmiş ağaçlar
- *Öneri:* Değişen en küçük kareler (ALS)
- *Kümeleme:* K-ortalamalar, Gauss karışımları (GMM'ler),
- *Konu modelleme:* Gizli dirichlet tahsisi (LDA)
- Sık öğe kümeleri, birliktelik kuralları ve sıralı model madenciliği(MLib, 2022).

### 2.2.2. BigQuery ve GDELT Veritabanı

#### 2.2.2.1. BigQuery

BigQuery, Google Cloud'un veri ambarı hizmetidir. Büyük veri analizi için tasarlanmış, tam olarak yönetilen ve sunucusuz bir veri ambarıdır. BigQuery, büyük veri kümelerinin hızlı bir şekilde analiz edilmesini ve sonuçların hızla alınmasını sağlar. Temel özellikleri arasında yüksek ölçeklenebilirlik, düşük maliyetli depolama ve hızlı sorgulama performansı bulunmaktadır.

BigQuery'nin başlıca özellikleri şunlardır:

Sunucusuz Mimari: BigQuery, sunucusuz bir hizmettir, bu da altyapı yönetimi gerektirmediği anlamına gelir. Kullanıcılar sadece verilerini yükler ve sorgularını çalıştırır.

Düşük Maliyetli Depolama: Büyük veri kümelerini düşük maliyetle saklayabilir. Depolama maliyetleri kullanılan depolama alanına göre belirlenir.

Yüksek Performanslı Sorgulama: BigQuery, çok büyük veri kümelerini hızlı bir şekilde sorgulamak için tasarlanmıştır. Sorgular paralel olarak işlenir ve sonuçlar hızlı bir şekilde elde edilir.

Entegrasyon ve İş Birliği: BigQuery, diğer Google Cloud hizmetleri ve üçüncü taraf araçlarıyla kolayca entegre edilebilir. Ayrıca, veri analistleri ve veri bilimcilerinin iş birliği yapmasını kolaylaştırır(Google, 2023).

#### 2.2.2.2. GDELT Veritabanı

GDELT (Global Database of Events, Language, and Tone) dünya çapında gerçekleşen olayları, haberleri ve medya kapsamını takip eden büyük bir veri tabanıdır.

GDELT, dünya genelinde her gün milyonlarca haber makalesini tarar ve analiz eder. Bu veri tabanı, sosyo-politik olayları, ekonomik gelişmeleri ve doğal afetleri izlemek için kullanılır.

GDELT'in başlıca özellikleri şunlardır:

*Kapsamlı Veri Kapsama:* GDELT, dünya çapında çeşitli dillerde yayınlanan haberleri, blogları ve sosyal medya gönderilerini analiz eder.

*Gerçek Zamanlı Güncellemeler:* Veri tabanı sürekli olarak güncellenir ve gerçek zamanlı veri sağlar.

*Tarihsel Veri:* GDELT, 1979'dan günümüze kadar olan olayları içerir, bu da uzun vadeli analizler için zengin bir veri kaynağı sağlar.

*Kategorik Veri:* Olaylar, sosyo-politik olaylar, protestolar, doğal afetler ve daha fazlası gibi farklı kategorilere ayrılmıştır.

### 2.3. Makine Öğrenmesi

Makine öğrenmesi (Machine Learning), bilgisayar sistemlerinin veri analizi yoluyla öğrenme deneyimi kazanmasını ve karmaşık problemleri çözmek için modeller oluşturmasını sağlayan bir yapay zekâ alt alanıdır. Makine öğrenimi, belirli bir görevi yerine getirmek için programlanmadığı halde, deneyimlerden öğrenerek ve verilere dayanarak kendini geliştirebilir. Temel olarak, makine öğrenimi algoritmaları, belirli bir amaca ulaşmak için veriye dayalı çıkarımlar yapar ve bu çıkarımları kullanarak gelecekteki olayları tahmin eder veya kararlar verir.

Makine öğrenimi, çeşitli türdeki verileri analiz etmek ve bunlardan anlamlı bilgiler çıkarmak için kullanılabilir. Örnek kullanım alanları arasında resim tanıma, metin sınıflandırma, ses tanıma, hastalık teşhisi, pazarlama tahmini, finansal tahminler ve otomatik sürüş sistemleri gibi birçok uygulama bulunmaktadır. Makine öğrenimi, istatistik, matematik, bilgisayar bilimi ve yapay zekâ gibi disiplinlerin birleşimini içerir ve sürekli olarak gelişen ve yayılan bir alandır.

Makine öğrenmesi ile ilgili genel tanımlar aşağıdaki tablolarda verilmiştir.

Tablo 2.1. Akademik tanımlar

| Kaynak | Tanım |
|---|---|
| Mitchell (1997) | "Bir bilgisayar programı, eğer belirli bir sınıftaki görevlerde T, performans ölçütü P kullanılarak, deneyim E'den öğrendikten sonra performansını P ile ölçülen şekilde geliştirebiliyorsa, bu program öğrenmiş sayılır." |
| Samuel (1959) | "Makine öğrenmesi, bilgisayarlara açıkça programlanmadan öğrenme yeteneği kazandıran bir çalışma alanıdır." |
| Alpaydın (2010) | "Makine öğrenmesi, deneyimle iyileşen bilgisayar algoritmalarının geliştirilmesi ve incelenmesidir." |

Tablo 2.2. Popüler tanımlar

| Tanım | Açıklama |
|---|---|
| Veri Temelli Öğrenme | Makine öğrenmesi, verilerden bilgi çıkarma ve bu bilgiyi kullanarak tahminlerde bulunma sürecidir. |
| Otomatik Gelişen Algoritmalar | Makine öğrenmesi, belirli görevleri insan müdahalesi olmadan yerine getirebilen algoritmaların geliştirilmesidir. |
| Bilgi Kazanımı ve Karar Verme | Makine öğrenmesi, bilgisayarların bilgi kazanmasını ve bu bilgiyi karar verme süreçlerinde kullanmasını sağlar. |

Tablo 2.3. Teknik tanımlar

| Tanım | Açıklama |
|---|---|
| Model Eğitimi ve Tahmin | Makine öğrenmesi, verilerden modeller eğiterek ve bu modelleri kullanarak gelecekteki olaylar veya bilinmeyen değerler hakkında tahminlerde bulunma sürecidir. |
| İstatistiksel Modelleme | Makine öğrenmesi, istatistiksel yöntemler kullanarak verilerdeki desenleri ve ilişkileri modelleme sürecidir. |
| Veri Analizi ve Desen Tanıma | Makine öğrenmesi, büyük veri kümelerinden desenler ve ilişkiler tanıyarak anlamlı bilgiler çıkarma sürecidir. |

Tablo 2.4. İşlevsel tanımlar

| Tanım | Açıklama |
|---|---|
| Otomatik Sınıflandırma ve Kümeleme | Makine öğrenmesi, verileri otomatik olarak sınıflandıran veya kümeler halinde gruplandıran teknikler bütünüdür. |
| Tahmin ve Öneri Sistemleri | Makine öğrenmesi, kullanıcıların ihtiyaçlarına göre önerilerde bulunan ve tahminler yapan sistemlerin temelidir. |
| Sürekli İyileşen Sistemler | Makine öğrenmesi, sürekli olarak yeni verilerle kendini güncelleyip performansını artıran sistemlerin geliştirilmesini sağlar. |

Tablo 2.5. Uygulama temelli tanımlar

| Tanım | Açıklama |
|---|---|
| Görüntü İşleme ve Bilgisayarla Görme | Makine öğrenmesi, görüntüleri analiz ederek nesne tanıma ve sınıflandırma gibi görevleri yerine getiren teknolojilerin temelidir. |
| Doğal Dil İşleme | Makine öğrenmesi, metin ve konuşma verilerini anlamak ve işlemek için kullanılan yöntemler bütünüdür. |
| Otonom Sistemler ve Robotik | Makine öğrenmesi, robotların ve otonom sistemlerin çevrelerini algılayarak ve öğrenerek hareket etmelerini sağlar. |

### 2.4. Makine öğrenmesinin Tarihçesi

Makine öğrenmesinin tarihsel gelişimi aşağıdaki tablolarda özetlenmiştir.

Tablo 2.6. Erken dönem ve temeller (1940'lar - 1960'lar)

| Yıl | Gelişme |
|---|---|
| 1943 | Warren McCulloch ve Walter Pitts, nöronların çalışma biçimlerini taklit eden ilk matematiksel modeli önerdiler. |
| 1950 | Alan Turing, makinelerin zeki davranışlarını test etmek için Turing Testi'ni önerdi. |
| 1952 | Arthur Samuel, dama oyunu oynayan ve kendi kendine öğrenen bir bilgisayar programı geliştirdi. |
| 1958 | Frank Rosenblatt, ilk yapay sinir ağı modeli olan Perceptron'u geliştirdi. |

Tablo 2.7. İlk algoritmalar ve kavramlar (1960'lar - 1980'ler)

| Yıl | Gelişme |
|---|---|
| 1967 | K-En Yakın Komşu (KNN) algoritması geliştirildi. |
| 1970 | Teorik çalışmalar ve istatistiksel modeller üzerine yoğunlaşıldı, Bayesian yöntemler ön plana çıktı. |
| 1980 | İlk uzman sistemler ve bilgi tabanlı sistemler geliştirildi. |
| 1986 | Geoffrey Hinton, David Rumelhart ve Ronald Williams, geri yayılım (backpropagation) algoritmasını tanıttı ve sinir ağlarının eğitimi için kullanıldı. |

Tablo 2.8. Makine öğrenmesinin yükselişi (1990'lar - 2000'ler)

| Yıl | Gelişme |
|---|---|
| 1995 | Destek Vektör Makineleri (SVM) algoritması Vladimir Vapnik tarafından geliştirildi ve popüler hale geldi. |
| 1997 | IBM'nin Deep Blue bilgisayarı, dünya satranç şampiyonu Garry Kasparov'u yendi. |
| 1999 | İlk büyük veri kümeleri ve veri madenciliği teknikleri kullanılmaya başlandı. |
| 2001 | Leo Breiman, rastgele orman (Random Forest) algoritmasını tanıttı. |

Tablo 2.9. Derin öğrenme ve modern dönem (2010'lar - Günümüz)

| Yıl | Gelişme |
|---|---|
| 2012 | AlexNet, ImageNet yarışmasında büyük bir başarı elde etti ve derin öğrenmenin gücünü gösterdi. |
| 2014 | Google DeepMind, Go oyunu oynayan ve dünya şampiyonlarını yenen AlphaGo'yu geliştirdi. |
| 2016 | AlphaGo, Lee Sedol'u yenerek büyük bir sansasyon yarattı. |
| 2018 | BERT (Bidirectional Encoder Representations from Transformers) modeli tanıtıldı ve doğal dil işleme alanında büyük ilerlemelere yol açtı. |
| 2020 | GPT-3 (Generative Pre-trained Transformer 3), dil modellemesi ve doğal dil işleme alanında büyük bir etki yarattı. |

Tablo 2.10. Önemli kavramlar ve gelişmeler

| Kavram | Açıklama |
|---|---|
| Yapay Sinir Ağları | İnsan beynindeki nöronların çalışma şeklini taklit eden ve veri üzerinden öğrenme gerçekleştiren modeller. |
| Geri Yayılım (Backpropagation) | Sinir ağlarını eğitmek için kullanılan bir algoritma, hataları geri yayarak ağı günceller. |
| Destek Vektör Makineleri (SVM) | Veri noktalarını en iyi ayıran sınırları bulmayı amaçlayan sınıflandırma algoritması. |
| Derin Öğrenme | Çok katmanlı yapay sinir ağları kullanarak karmaşık verilerden anlamlı bilgiler çıkarma yöntemi. |
| Büyük Veri | Çok büyük ve karmaşık veri kümeleri, geleneksel veri işleme yöntemleriyle analiz edilemez. |
| Doğal Dil İşleme (NLP) | İnsan dilini anlamak ve işlemek için kullanılan makine öğrenmesi yöntemleri. |

Makine öğrenmesinin tarihçesi, sürekli olarak gelişen ve yenilenen bir alandır. Teknolojik ilerlemeler ve artan veri miktarı, bu alandaki yenilikleri ve keşifleri hızlandırmaktadır.

### 2.5. Makine Öğrenmesi Şema

Makine Öğrenmesi Algoritmaları aşağıdaki şemada verilmiştir.

├── **Tahmin (Prediction)**
│ ├── Regresyon (Regression)
│ │ ├── Doğrusal Regresyon (Linear Regression)
│ │ ├── Lojistik Regresyon (Logistic Regression)
│ │ ├── Polinom Regresyon (Polynomial Regression)
│ │ ├── Ridge Regresyon (Ridge Regression)
│ │ └── Lasso Regresyon (Lasso Regression)
│
├── **Sınıflandırma (Classification)**
│ ├── Gözetimli Öğrenme (Supervised Learning)
│ │ ├── K-En Yakın Komşu (K-Nearest Neighbors, KNN)
│ │ ├── Destek Vektör Makineleri (Support Vector Machines, SVM)

│ │ ├── Karar Ağaçları (Decision Trees)
│ │ ├── Rastgele Ormanlar (Random Forests)
│ │ ├── Naive Bayes
│ │ ├── Yapay Sinir Ağları (Artificial Neural Networks, ANN)
│ │ ├── Lojistik Regresyon (Logistic Regression)
│ │ └── AdaBoost
│
└── **Kümeleme (Clustering)**
  ├── Gözetimsiz Öğrenme (Unsupervised Learning)
  │ ├── K-Ortalamalar (K-Means)
  │ ├── Hiyerarşik Kümeleme (Hierarchical Clustering)
  │ ├── DBSCAN (Density-Based Spatial Clustering of Applications with Noise)
  │ ├── Gaussian Karışım Modelleri (Gaussian Mixture Models, GMM)
  │ └── Spektral Kümeleme (Spectral Clustering)

Tablo 2.11. Tahmin (Prediction)-Regresyon(Regression)

| **Algoritma** | **Formül** | **Yazar** | **Tarih** |
|---|---|---|---|
| Doğrusal Regresyon (Linear Regression) | $y = b_0 + b_1 x$ | Legendre | 1805 |
| Lojistik Regresyon (Logistic Regression) | $logit(p) = ln\left(\frac{p}{1-p}\right) = b_0 + b_1 x$ | Cox | 1958 |
| Polinom Regresyon (Polynomial Regression) | $y = b_0 + b_1 x + b_2 x^2$ | Galton | 1886 |
| Ridge Regresyon (Ridge Regression) | $\text{Minimize}\left(\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda\sum_{j=1}^{p}\beta_j^2\right)$ | Hoerl ve Kennard | 1970 |
| Lasso Regresyon (Lasso Regression) | $\text{Minimize}\left(\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda\sum_{j=1}^{p}\lvert\beta_j\rvert\right)$ | Tibshirani | 1996 |

Tablo 2.12. Sınıflandırma (Classification)-Gözetimli Öğrenme(Supervized Learning)

| **Algoritma** | **Formül** | **Yazar** | **Tarih** |
|---|---|---|---|
| KNN | $d(x,y)=\sqrt{\sum_{i=1}^{n}(x_i-y_i)^2}$ | Fix ve Hodges | 1951 |
| DVM | $\min_{\mathbf{w},b}\frac{1}{2}\mathbf{w}^T\mathbf{w}$ subject to $y_i(\mathbf{w}^T\mathbf{x}_i+b)\geq 1 \quad \forall i$ | Vapnik | 1963 |
| KA (Bilgi Kazancı) | $IG(S,A)=H(S)-\sum_{v\in \text{Values}(A)}\frac{\lvert S_v\rvert}{\lvert S\rvert}H(S_v)$ | Quinlan | 1986 |
| KA (Entropi) | $H(S)=-\sum_{i=1}^{c}p_i\log_2 p_i$ | Shannon | 1948 |
| Rastgele Ormanlar (Random Forests) | $H(x)=\text{mode}\{h_1(x),h_2(x),\ldots,h_n(x)\}$ | Breiman | 2001 |
| Gini İndeksi | $G(S)=1-\sum_{i=1}^{c}p_i^2$ | Gini | 1912 |
| Bilgi Kazancı | $IG(S,A)=H(S)-\sum_{v\in \text{Values}(A)}\frac{\lvert S_v\rvert}{\lvert S\rvert}H(S_v)$ | Quinlan | 1986 |
| Naive Bayes | $P(C\mid X)=\frac{P(x_1\mid C)\cdot P(x_2\mid C)\cdot\ldots\cdot P(x_n\mid C)\cdot P(C)}{P(x_1)\cdot P(x_2)\cdot\ldots\cdot P(x_n)}$ | Bayes | 1763 |
| YSA (ANN) | $a^{(l+1)}=\sigma(W^{(l)}a^{(l)}+b^{(l)})$ | McCulloch ve Pitts | 1943 |
| LR | $P(y=1\mid x)=\frac{1}{1+e^{-(w_0+w_1x_1+\cdots+w_nx_n)}}$ | Cox | 1958 |
| AdaBoost | $H(x)=\text{sign}\left(\sum_{m=1}^{M}\alpha_m\cdot h_m(x)\right)$ | Freund ve Schapire | 1995 |

Tablo 2.13. Kümeleme(Clustering)-Gözetimsiz Öğrenme (Unsupervised Learning)

| Algoritma | Formül | Yazar | Tarih |
|---|---|---|---|
| K-Ortalamalar (K-Means) | $\min \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2$ | Lloyd, MacQueen | 1957 |
| Hiyerarşik Kümeleme (Hierarchical Clustering) | $d_{ij} = \min_{k \in C_i, l \in C_j} d(k,l)$ | Johnson, Lance, Williams | 1967 |
| DBSCAN (Density-Based Spatial Clustering of Applications with Noise) | $\text{Core Point} \Rightarrow \left\|N_{\dot{o}}(p)\right\| \geq \text{MinPts}$ | Ester, Kriegel, Sander ve Xu | 1996 |
| Gaussian Karışım Modelleri (Gaussian Mixture Models, GMM) | $p(x) = \sum_{k=1}^{K} \pi_k \mathrm{N}\ (x \mid \mu_k, \Sigma_k)$ | Duda ve Hart | 1973 |
| Spektral Kümeleme (Spectral Clustering) | $L = D - A$ | Ng ve Jordan | 2001 |

### 2.6. Makine Öğrenmesi Tahmin Algoritmaları

Tahmin algoritmaları, sürekli değerlerin tahmin edilmesine odaklanır. Regresyon yöntemleri, bağımlı bir değişkenin bağımsız değişkenlere bağlı olarak nasıl değiştiğini modellemek için kullanılır.

#### 2.6.1. Doğrusal Regresyon

Doğrusal regresyon, iki değişken arasındaki ilişkiyi modellemek için kullanılan istatistiksel bir yöntemdir. Bir bağımlı değişken Y ve bir veya daha fazla bağımsız

değişken X arasında doğrusal bir ilişkiyi temsil eder. Basit doğrusal regresyon, sadece bir bağımsız değişken kullanarak ilişkiyi modellemeye çalışır. Basit doğrusal regresyonun formülü şu şekildedir:

$$Y = b_0 + b_1 X \quad \textbf{(2.1)}$$

Y: Bağımlı değişken (tahmin edilen değer)

X: Bağımsız değişken

$b_0$: Y-ekseni kesişimi (intercept)

$b_1$: Eğim (slope)

Doğrusal regresyon, birçok farklı alanda kullanılabilir:

*Ekonomi:* İki ekonomik değişken arasındaki ilişkiyi modellemek.

*Tıp:* Bir tedavi yöntemi ile hasta sonuçları arasındaki ilişkiyi belirlemek.

*Mühendislik:* Üretim sürecinde bağımsız değişkenlerin çıktıyı nasıl etkilediğini analiz etmek.

Doğrusal regresyon analizinde, modelin doğruluğunu sağlamak için belirli varsayımlar yapılır:

*Doğrusallık:* Bağımlı ve bağımsız değişkenler arasındaki ilişki doğrusaldır.

*Normallik:* Hata terimleri normal dağılıma sahiptir.

*Homoskedastisite:* Hata terimlerinin varyansı sabittir.

*Bağımsızlık:* Hata terimleri birbirinden bağımsızdır.

Bu varsayımlar ihlal edildiğinde, modelin sonuçları yanıltıcı olabilir ve düzeltilmesi gerekebilir.

### 2.6.2. Lojistik Regresyon

Lojistik regresyon, bağımlı değişkenin kategorik olduğu durumlarda kullanılan bir istatistiksel yöntemdir. En yaygın kullanımı, ikili sınıflandırma (binary

classification) problemleridir. Örneğin, bir e-posta spam mi yoksa değil mi, bir hastanın hastalığı var mı yok mu gibi ikili sonuçlar.

$$logit(p) = ln\left(\frac{p}{1-p}\right) = b_0 + b_1 x \quad \textbf{(2.2)}$$

$p$: Bir olayın olasılığı (örneğin, e-posta'nın spam olma olasılığı)

$b_0$: Y-ekseni kesişimi (intercept)

$b_1$: Eğim (slope)

$x$: Bağımsız değişken

Bu formül, p olasılığını sigmoid fonksiyonu kullanarak hesaplamak için yeniden düzenlenir:

$$p = \frac{1}{1 + e^{-(b_0 + b_1 x)}} \quad \textbf{(2.3)}$$

Lojistik regresyon, bağımlı değişkenin kategorik olduğu durumlarda kullanılır ve olayın olasılığını tahmin etmek için sigmoid fonksiyonunu kullanır. Model, bağımsız değişkenlerin katsayılarını kullanarak olasılıkları hesaplar ve bu olasılıkları kullanarak sınıflandırma yapar.

### 2.6.3. Polinom Regresyon

Polinom regresyon, bağımsız ve bağımlı değişkenler arasındaki ilişkiyi modellemek için kullanılan doğrusal olmayan bir regresyon tekniğidir. Doğrusal regresyondan farklı olarak, bağımsız değişkenlerin polinom terimlerini (kuvvetlerini) kullanarak daha karmaşık ilişkileri modellemeye çalışır.

$$y = b_0 + b_1 x + b_2 x^2 + b_3 x^3 + \cdots + b_n x^n \quad \textbf{(2.4)}$$

y: Bağımlı değişken

x: Bağımsız değişken

$b_0, b_1, b_2, \ldots, b_n$: Katsayılar

n: Polinomun derecesi

Polinom regresyon, bağımsız değişkenler ile bağımlı değişken arasındaki doğrusal olmayan ilişkileri modellemek için kullanılan bir tekniktir. Polinom terimlerini kullanarak daha karmaşık desenleri yakalamaya çalışır.

### 2.6.4. Ridge Regresyon

Ridge regresyon, doğrusal regresyon modelinde aşırı uyumu (overfitting) önlemek için kullanılan bir düzenlileştirme (regularization) tekniğidir. Ridge regresyon, doğrusal regresyon modeline bir ceza terimi ekleyerek model katsayılarının büyüklüğünü sınırlamayı amaçlar. Bu ceza terimi, katsayıların karelerinin toplamı ile orantılıdır ve bu sayede büyük katsayıların etkisi azaltılır. Böylece modelin katsayıları sınırlanarak daha genel ve kararlı tahminler elde edilir.

Ridge regresyonun maliyet fonksiyonu(loss function) (2.5)'deki gibidir:

$$\text{Minimize}\left( \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right) \quad \textbf{(2.5)}$$

$y_i$: Gerçek değerler

$\hat{y}_i$: Tahmin edilen değerler

$\beta_j$: Katsayılar

Lambda (λ): Düzenlileştirme parametresi (ceza terimi)

n: Gözlem sayısı

p: Bağımsız değişken sayısı

$$\boldsymbol{\beta} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y} \quad \textbf{(2.6)}$$

X: Giriş veri matrisi

y: Çıkış vektörü

I: Birim matris

λ: Düzenlileştirme parametresi

### 2.6.5. Lasso Regresyon

Lasso regresyonu, doğrusal regresyon modellerinde ortaya çıkan çoklu doğrusal bağlantı probleminin üstesinden gelmek ve modelde anlamsız değişkenleri dışlamak amacıyla kullanılan bir yöntemdir. Lasso (Least Absolute Shrinkage and Selection Operator) regresyonu, parametre tahminlerini küçülterek bazı parametreleri sıfır yapar ve böylece modelde sadece anlamlı değişkenlerin kalmasını sağlar (Tibshirani, 1996).

Lasso regresyonunun optimizasyon problemi (2.7)'de verilmiştir.

$$\min_{\beta}\left(\sum_{i=1}^{n}(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij})^2+\lambda\sum_{j=1}^{p}|\beta_j|\right) \quad \textbf{(2.7)}$$

$y_i$ : Bağımlı değişkenin gözlemi

$x_{ij}$ : Bağımsız değişkenlerin gözlemleri

$\beta_0$ : Sabit terim (intercept)

$\beta_j$ :Model parametreleri

$\lambda$ (lambda): Ceza teriminin ağırlığını belirleyen hiperparametre

(2.7) doğrusal regresyon denkleminin hata terimini ve parametrelerin mutlak değerlerinin toplamını minimize eder. Burada, λ ceza teriminin ağırlığını belirleyen hiperparametredir. λ değeri arttıkça, bazı parametreler sıfıra eşitlenir ve modelde sadece önemli değişkenler kalır (Aytekin, 2021).

Lasso regresyonu, Ridge ve ElasticNet regresyon yöntemleriyle benzer amaçlar taşır. Ancak, Ridge regresyonu kare ceza terimi kullanırken, Lasso regresyonu mutlak değer ceza terimi kullanır. ElasticNet ise her iki ceza terimini birleştirir. Bu yöntemler, modelin aşırı uyum yapmasını engelleyerek genellenebilirliğini artırmayı hedefler. Lasso regresyon, aşırı uyumu önlemek ve bazı katsayıları sıfıra yaklaştırmak için kullanılan etkili bir tekniktir.

## 2.7. Sınıflandırma

### 2.7.1. K-En Yakın Komşu

K-En Yakın Komşu (K-Nearest Neighbors, KNN) algoritması, yeni bir veri noktasının sınıfını veya değerini belirlemek için en yakın k komşusunu kullanır. Bu

komşuların çoğunluk sınıfı veya ortalama değeri, tahmin edilen sınıf veya değer olarak kabul edilir. KNN algoritması, sınıflandırma ve regresyon problemlerinde kullanılabilir.

K-en yakın komşular, mevcut tüm vakaları depolayan ve yeni vakaları mesafe fonksiyonları gibi bir benzerlik ölçüsüne göre sınıflandıran basit bir algoritmadır. KNN, 1970'lerin başında parametrik olmayan bir teknik olarak istatistiksel tahmin ve örüntü tanımada kullanılmıştır (Sayad,2023). K-En Yakın Komşular (KNN) hem sınıflandırma hem de regresyon görevleri için kullanılan popüler bir makine öğrenimi algoritmasıdır. Parametrik olmayan, örnek tabanlı bir öğrenme algoritmasıdır, yani bir model oluşturmadan çevresindeki veri noktalarına dayalı tahminler yapar. KNN'de açık bir eğitim aşaması yoktur. Algoritma, tahminler yapmak için kullanılacak olan tüm veri kümesini saklar. KNN, bir tahmin yaparken dikkate alınacak en yakın komşu sayısını (K) belirtmenizi gerektirir. Bu, ayarlamanız gereken bir hiperparametredir. Daha küçük bir K değeri (örneğin, K=1) algoritmayı gürültüye karşı daha hassas hale getirirken, daha büyük bir K değeri tahminleri yumuşatabilir ancak bazı ayrıntı kayıplarına yol açabilir. Sınıflandırmak istediğiniz belirli bir veri noktası için KNN, tipik olarak Öklid mesafesi gibi bir mesafe metriğine dayalı olarak eğitim kümesindeki en yakın K veri noktasını bulur. K-en yakın komşular arasında her sınıftaki veri noktalarının sayısını sayar. Veri noktası daha sonra K-en yakın komşuları arasında en yaygın olan sınıfa atanır. Regresyon görevleri için KNN benzer şekilde çalışır, ancak bir sınıfı tahmin etmek yerine sayısal bir değer tahmin eder. K-en yakın komşuların hedef değerlerinin ortalamasını (veya ağırlıklı ortalamasını) hesaplar ve bu ortalamayı veri noktası için öngörülen değer olarak atar. KNN, sınıflandırma ve regresyon görevleri için basit ancak etkili bir yaklaşıma ihtiyaç duyulduğunda, özellikle küçük ve orta ölçekli veri kümeleri için değerli bir algoritmadır. Bununla birlikte, sınırlamalar dikkatlice değerlendirmeli ve özel problemler için uygun olduğunda kullanılmalıdır.

KNN algoritmasının temel adımları şu şekildedir; eğitim veri setindeki tüm noktalar arasındaki mesafeleri hesaplayın. En yakın k komşuyu belirleyin. Sınıflandırma için, komşuların çoğunluk sınıfını seçin. Regresyon için, komşuların ortalama değerini hesaplayın. Euclidean mesafesi en yaygın kullanılan mesafe ölçüsüdür ve iki veri noktası arasındaki mesafe (2.8)'de verilmiştir.

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad \textbf{(2.8)}$$

d(x,y): x ve y veri noktaları arasındaki Euclidean mesafesi

$x_i$: x veri noktasının i'inci özelliği

$y_i$: y veri noktasının i'inci özelliği

n: Özelliklerin sayısı

K-En Yakın Komşu (KNN) algoritması, basit ve etkili bir sınıflandırma ve regresyon yöntemidir. Özellikle, veri setinde belirgin bir yapı olmadığı durumlarda ve veriler arasında doğrusal olmayan ilişkilerin olduğu durumlarda kullanışlıdır.

### 2.7.2. Destek Vektör Makineleri

Destek Vektör Makineleri (Support Vector Machines-SVM), sınıflandırma ve regresyon analizlerinde kullanılan güçlü bir denetimli öğrenme algoritmasıdır. SVM, veriyi farklı sınıflara ayıran en iyi hiper-düzlemi bulmaya çalışır. En iyi hiper-düzlem, iki sınıf arasındaki marjini maksimize eden düzlemdir.

Destek Vektör Makinesi, sınıflandırma ve regresyon görevleri için kullanılan popüler bir denetimli makine öğrenimi algoritmasıdır. Destek Vektör Makineleri özellikle karmaşık, yüksek boyutlu veri kümeleriyle uğraşırken kullanışlıdır. Verileri farklı sınıflara etkili bir şekilde ayırabilen optimum hiper düzlemleri bulma yetenekleriyle bilinirler. Destek Vektör Makinesi, sınıflandırma hatalarını en aza indirirken sınıflar arasındaki marjı en üst düzeye çıkarmayı amaçlar. Destek vektör makinesi (DVM), iki sınıf arasındaki marjı maksimize eden hiper düzlemi bularak sınıflandırma yapar. Hiper düzlemi tanımlayan vektörler (durumlar) destek vektörleridir. DVM, metin sınıflandırma, görüntü tanıma ve biyoinformatik dahil olmak üzere çeşitli uygulamalar için güçlü bir algoritmadır. Destek Vektör Makinesi kullanırken, hiperparametrelerin dikkatli bir şekilde ayarlanması ve uygun çekirdeğin seçilmesi, özel probleminizde optimum performans elde etmek için çok önemlidir.

İkili Sınıflandırma Problemi: Veri kümesi $(x_i, y_i)$ şeklinde olup burada $x_i$ özellik vektörü ve $y_i \in (-1,1)$ sınıf etiketidir. Destek Vektör Makinesi (2.9)'daki optimizasyon problemine dayanır:

$$\min_{\mathbf{w},b} \frac{1}{2}\mathbf{w}^T\mathbf{w} \quad \text{subject to} \quad y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \quad \forall i \quad \textbf{(2.9)}$$

w: Ağırlık vektörü

b: Bias terimi

Bu optimizasyon problemi, ikili sınıflandırma problemini çözer ve en iyi hiper-düzlemi bulur.

Lagrange Çarpanları ile Optimizasyon:

$$L(\mathbf{w},b,\alpha)=\frac{1}{2}\mathbf{w}^T\mathbf{w}-\sum_{i=1}^{n}\alpha_i[y_i(\mathbf{w}^T\mathbf{x}_i+b)-1] \quad \textbf{(2.10)}$$

$\propto_i$: Lagrange çarpanları (dual değişkenler)

Karar Fonksiyonu: Optimizasyon probleminden sonra elde edilen w ve b ile karar fonksiyonu şu şekilde olur:

$$f(x)=\text{sign}(\mathbf{w}^T\mathbf{x}+b) \quad \textbf{(2.11)}$$

Destek Vektör Makinesi algoritması, veriyi farklı sınıflara ayıran en iyi hiper-düzlemi bulmaya çalışır.

### 2.7.3. Karar Ağaçları

Karar ağaçları (Decision Trees), veri sınıflandırma ve regresyon problemlerini çözmek için kullanılan denetimli öğrenme algoritmalarıdır. Karar ağacı modeli, veri özelliklerine göre dallara ayrılır ve sonuçları yaprak düğümlerinde depolar. Her dal, veri setindeki özellikler arasındaki bir ayrımı temsil eder.

Karar ağaçlarının inşasında kullanılan bazı temel kavramlar ve formüller:

Entropi: Bir veri kümesinin düzensizliğini ölçen bir metriktir. İkili sınıflandırma için entropi (2.12)'deki gibi hesaplanır:

$$H(S)=-\sum_{i=1}^{c}p_i\log_2 p_i \quad \textbf{(2.12)}$$

H(S): Veri kümesi S için entropi

$p_i$: i'inci sınıfın olasılığı

c: Sınıf sayısı

Bilgi Kazancı (Information Gain): Bir özellik kullanılarak veri kümesini bölmenin etkinliğini ölçer. Bir özellik A için bilgi kazancı şu şekilde hesaplanır:

$$IG(S,A) = H(S) - \sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} H(S_v) \quad \textbf{(2.13)}$$

IG(S,A): Özellik A kullanılarak veri kümesi S'yi bölme sonucu elde edilen bilgi kazancı

H(S): Veri kümesi S'nin entropisi

$S_v$: Özellik A'nın v değerine sahip alt kümesi

|S|: Veri kümesindeki örneklerin sayısı

$|S_v|$: Alt küme $S_v$'deki örneklerin sayısı

Karar ağaçları, veri sınıflandırma ve regresyon problemlerini çözmek için kullanılan güçlü algoritmalardır.

### 2.7.4. Rastgele Ormanlar

Rastgele Ormanlar (Random Forests), birden fazla karar ağacının birleşimiyle oluşturulan bir topluluk (ensemble) öğrenme yöntemidir. Her bir karar ağacı, veri setinin farklı bir alt kümesi ve özelliklerin rastgele bir alt kümesi kullanılarak eğitilir. Nihai tahmin, tüm ağaçların tahminlerinin çoğunluk oyu (sınıflandırma) veya ortalaması (regresyon) alınarak belirlenir.

Rastgele Ormanlar algoritmasının temel adımları aşağıda verilmiştir.

Veri Alt Kümelerinin Oluşturulması (Bootstrap Sampling): Orijinal veri setinden n örnek alınarak k sayıda veri alt kümesi oluşturulur.

Karar Ağaçlarının Eğitilmesi: Her bir veri alt kümesi kullanılarak bir karar ağacı eğitilir. Her bir düğümde, özelliklerin rastgele bir alt kümesi kullanılarak en iyi ayrım belirlenir.

Tahminlerin Birleştirilmesi: Sınıflandırma, Her bir ağaç tahminini yapar ve sınıf tahminleri çoğunluk oyu ile belirlenir. Regresyon, Her bir ağaç tahminini yapar ve tahminlerin ortalaması alınır.

Karar Ağaçlarının Eğitilmesi: Her bir karar ağacının düğümlerinin ayrımı için bilgi kazancı veya Gini indeksini kullanabiliriz. Bilgi kazancı formülü şu şekildedir:

$$IG(S,A) = H(S) - \sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} H(S_v)$$

Gini indeksi formülü ise şu şekildedir:

$$G(S) = 1 - \sum_{i=1}^{c} p_i^2$$

Rastgele Ormanlar, birden fazla karar ağacının birleşimiyle oluşturulan güçlü bir topluluk öğrenme yöntemidir.

### 2.7.5. Naif Bayes Algoritması

Naif Bayes (Naive Bayes), Bayes teoremine dayanan basit ama güçlü bir sınıflandırma algoritmasıdır. Bu algoritma, özelliklerin birbirinden bağımsız olduğunu varsayar. Bu nedenle "naive" (saf) olarak adlandırılır. Naive Bayes, metin sınıflandırma, spam filtresi ve belge sınıflandırma gibi çeşitli uygulamalarda yaygın olarak kullanılır.

Bayes Teoremi: Bayes teoremi, bir olayın olasılığını, bu olayla ilgili başka bir olayın gerçekleşmesi koşuluyla hesaplamamıza olanak tanır. Bayes teoremi şu şekilde ifade edilir:

$$P(C \mid X) = \frac{P(X \mid C) \cdot P(C)}{P(X)} \quad \textbf{(2.14)}$$

P(C|X): X gözlemlendiğinde C sınıfının olasılığı (posterior probability)

P(X|C): C sınıfı verildiğinde X gözleminin olasılığı (likelihood)

P(C): C sınıfının apriori olasılığı (prior probability)

P(X): X gözleminin olasılığı (evidence)

Naif Bayes sınıflandırması: Bayes teoreminin uygulanmasıyla gerçekleştirilir. Özelliklerin birbirinden bağımsız olduğu varsayılarak, $X=\{x_1,x_2,\ldots,x_n\}$ gözlemi için olasılık (2.15)'deki gibidir.

$$P(C \mid X) = \frac{P(x_1 \mid C) \cdot P(x_2 \mid C) \cdot \ldots \cdot P(x_n \mid C) \cdot P(C)}{P(x_1) \cdot P(x_2) \cdot \ldots \cdot P(x_n)} \quad \textbf{(2.15)}$$

(2.15)'de payda tüm sınıflar için sabittir, sınıflandırmayı gerçekleştirmek için sadece payı maksimize etmek yeterlidir:

$$P(C \mid X) \propto P(C) \prod_{i=1}^{n} P(x_i \mid C) \quad \textbf{(2.16)}$$

Naif Bayes algoritması, Bayes teoremine dayanan ve özelliklerin birbirinden bağımsız olduğunu varsayan basit ama güçlü bir sınıflandırma algoritmasıdır.

### 2.7.6. Yapay Sinir Ağları

Yapay Sinir Ağları-YSA (Artificial Neural Networks), insan beynindeki sinir ağlarının çalışma prensiplerinden esinlenerek geliştirilen ve veri sınıflandırma, regresyon, kümeleme gibi çeşitli makine öğrenmesi problemlerini çözmek için kullanılan güçlü algoritmalardır. Yapay sinir ağları, katmanlar halinde düzenlenmiş yapay nöronlardan oluşur. Her nöron, belirli bir ağırlık ve aktivasyon fonksiyonu kullanarak girişleri işler ve bir çıktı üretir.

Genellikle basitçe sinir ağları olarak adlandırılan Yapay Sinir Ağları (YSA), insan beyninin yapısı ve işlevinden esinlenen bir makine öğrenimi modelleri sınıfıdır. YSA'lar birbirine bağlı yapay nöronlardan veya katmanlar halinde düzenlenmiş düğümlerden oluşur. Çok katmanlı modellere odaklanan makine öğreniminin bir alt alanı olan derin öğrenmenin temel bir bileşenidir. Nöronlar bir sinir ağının temel yapı taşlarıdır. Her nöron bir veya daha fazla girdi değeri alır, bu girdiler üzerinde hesaplamalar yapar ve bir çıktı üretir. Nöronlar katmanlar halinde organize edilir. Tipik bir sinir ağı bir giriş katmanı, bir veya daha fazla gizli katman ve bir çıkış katmanından oluşur. Giriş katmanı verilerin özelliklerini temsil eder, gizli katmanlar bu özellikleri işler ve çıkış katmanı nihai tahmini veya sonucu üretir. Nöronlar arasındaki her bağlantı, bağlantının gücünü temsil eden bir ağırlıkla ilişkilendirilir. Her nöron ayrıca aktivasyon fonksiyonunda bir kaymaya izin veren bir önyargı terimine sahiptir. Her bir nöronun çıktısı, girdilerinin ve önyargılarının ağırlıklı toplamına bir aktivasyon fonksiyonu uygulanarak belirlenir. Yaygın aktivasyon fonksiyonları arasında sigmoid, ReLU (Rectified Linear Unit) ve tanh bulunur. Giriş verileri giriş katmanına iletilir. Hesaplamalar katman katman yapılır ve bilgi ağ boyunca ileriye doğru yayılır. Her nöron, ağırlıklı girdilerini, önyargılarını ve aktivasyon

fonksiyonunu kullanarak çıktısını hesaplar. Bir kayıp fonksiyonu, tahmin edilen çıktı ile gerçek hedef değerler arasındaki uyuşmazlığı ölçer. Yaygın kayıp fonksiyonları arasında regresyon görevleri için Ortalama Karesel Hata (MSE) ve sınıflandırma görevleri için çapraz entropi bulunur. Ağ, geriye yayılma adı verilen bir süreç kullanarak kayıp fonksiyonunu en aza indirmek için ağırlıklarını ve önyargılarını ayarlar. Gradyanlar kayba göre hesaplanır ve ağırlıklar ve önyargılar gradyan inişi veya varyantlarından biri kullanılarak güncellenir. Eğitim süreci, eğitim verilerinin ağa iteratif olarak sunulmasını, kaybın hesaplanmasını ve modelin parametrelerinin güncellenmesini içerir. Amaç, kayıp fonksiyonunu en aza indiren optimum ağırlıkları ve önyargıları bulmaktır. Eğitildikten sonra sinir ağı, öğrenilen ağırlıkları ve önyargıları kullanarak ileri yayılım gerçekleştirerek yeni, görülmemiş veriler üzerinde tahminler yapabilir.

Sinir ağları, çözmek için tasarlandıkları soruna bağlı olarak boyut ve mimari açısından önemli ölçüde değişebilir. Derin Sinir Ağları (DNN'ler) birden fazla gizli katmana sahiptir ve özellikle verilerin hiyerarşik temsillerini öğrenmede etkilidir. Sinir ağları, özellikle de derin öğrenme modelleri, çok çeşitli uygulamalar üzerinde önemli bir etkiye sahip olmuş ve modern makine öğreniminde temel bir teknoloji haline gelmiştir. Yapay sinir ağı, insan beyninin çalışma şekli modellenerek oluşturulan bir bilgisayar yazılımıdır (Kustrin and Beresford,2000).

Son yıllarda birçok araştırmacı yapay sinir ağlarının (YSA) kanser türlerinin sınıflandırılması, aktivite tanıma ve görüntü işleme gibi çeşitli alanlarda etkili olduğunu savunmaktadır (Yue ve ark., 2019). Nöron olarak bilinen beyin hücreleri, sinapslar aracılığıyla birbirlerine bağlanarak bir ağ oluştururlar. Sinir ağındaki tüm bağlantıların (sinapsların) ağırlıkları aynı değildir. Bir yapay sinir ağının hesaplama gücü, düğümler arasındaki bağlantı tarafından belirlenir (Bartosch-Harlid et al., 2008). YSA, veriler arasındaki farklı ilişki biçimlerini deşifre ederek öğrenmeyi gerçekleştirir. Elde edilen bu bilgi, yeni verilerin sınıfını belirlerken bir kural oluşturmak için kullanılır (Al-Nuaimy et al., 2000).

Nöronun Çıkışı: Bir yapay nöronun çıkışı, ağırlıklı girişlerin toplamı ve bir aktivasyon fonksiyonu uygulanarak hesaplanır.

$$z = \sum_{i=1}^{n} w_i x_i + b \tag{2.17}$$

z: Lineer kombinasyon (nöronun girişi)

$w_i$: i-nci girdinin ağırlığı

$x_i$: i-nci giriş değeri

b: bias terimi

Aktivasyon Fonksiyonu: Lineer kombinasyon z'yi işleyerek nöronun çıkışını üretir. Yaygın olarak kullanılan aktivasyon fonksiyonları şunlardır:

Sigmoid Fonksiyonu: Yapay sinir ağlarında yaygın olarak kullanılan aktivasyon fonksiyonlarından biridir. Bu fonksiyon, girdiyi (genellikle bir sinir hücresinin toplam girişini) 0 ile 1 arasında bir değere dönüştürür. Sigmoid fonksiyonunun matematiksel ifadesi (2.18)'deki gibidir:

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{2.18}$$

ReLU (Rectified Linear Unit) Fonksiyonu formülü,

$$\mathrm{ReLU}(z) = \max(0, z) \tag{2.19}$$

İleri Yayılım (Feedforward): Bir yapay sinir ağı, ileri yayılım süreci ile çalışır. Bu süreç, giriş katmanından başlayarak her katmanda nöronların çıkışlarının hesaplanmasını içerir. Örneğin, iki katmanlı bir sinir ağında ileri yayılım (2.20) ve (2.21)'de verilmiştir.

$$a^{(1)} = \sigma(W^{(1)}x + b^{(1)}) \tag{2.20}$$

$$a^{(2)} = \sigma(W^{(2)}a^{(1)} + b^{(2)}) \tag{2.21}$$

$a^{(1)}$: İlk katmanın aktivasyonları

$W^{(1)}$,$W^{(2)}$: Ağırlık matrisleri

$b^{(1)}$,$b^{(2)}$: Bias vektörleri

x: Giriş vektörü

Geri Yayılım (Backpropagation): Hata terimlerinin hesaplanması ve ağırlıkların güncellenmesi sürecidir. Hata terimi δ-delta ve ağırlık güncelleme aşağıdaki gibidir:

$$\delta^{(L)} = \nabla_a C \odot \sigma'(z^{(L)}) \quad \textbf{(2.22)}$$

$$d^{(l)} = ((W^{(l+1)})^T d^{(l+1)}) es'(z^{(l)}) \quad \textbf{(2.23)}$$

Ağırlık güncellemesi,

$$W^{(l)} = W^{(l)} - \eta \Delta W^{(l)} \quad (2.24)$$

$\delta^{(L)}$\: L-inci katmandaki hata terimi

$\nabla_a C$: Maliyet fonksiyonunun aktivasyonlara göre gradyanı

$\sigma'(z)$: Aktivasyon fonksiyonunun türevi

$\eta$: Öğrenme oranı

Yapay Sinir Ağları, katmanlar halinde düzenlenmiş yapay nöronlardan oluşan ve ileri yayılım ve geri yayılım süreçleriyle çalışan güçlü algoritmalardır. Bu algoritma, el yazısı rakamların tanınması gibi çeşitli makine öğrenmesi problemlerinde başarılı bir şekilde uygulanabilir.

### 2.7.7. Lojistik Regresyon Algoritması

Lojistik regresyon, bağımlı değişkenin kategorik olduğu durumlarda kullanılan bir regresyon türüdür. Özellikle ikili (binary) sınıflandırma problemlerinde yaygın olarak kullanılır. Lojistik regresyon, doğrusal regresyonun aksine, bağımlı değişkenin olasılıklarını tahmin etmek için sigmoid fonksiyonunu kullanır.

Lojistik regresyon, ikili sınıflandırma görevleri için kullanılan bir makine öğrenimi algoritmasıdır. Verilen bir girdinin iki olası sınıftan birine ait olma olasılığını modelleyen istatistiksel bir yöntemdir. Adına rağmen, lojistik regresyon bir regresyon algoritması değil, bir sınıflandırma algoritmasıdır. Lojistik regresyon, belirli bir girdinin pozitif sınıfa (örneğin, sınıf 1) ait olma olasılığını tahmin etmek için genellikle sigmoid fonksiyon olarak adlandırılan lojistik fonksiyonu kullanır.

Sigmoid fonksiyonu, herhangi bir gerçek değerli sayıyı 0 ile 1 arasında bir değere eşleyen S şeklinde bir eğridir. Optimum parametreleri bulmaya yönelik tipik yaklaşım, gradyan inişi gibi bir optimizasyon algoritmasıdır. Amaç, tahmin edilen olasılıklar ile gerçek sınıf etiketleri arasındaki uyumsuzluğu ölçen bir maliyet fonksiyonunu en aza indirmektir. Lojistik regresyonda yaygın olarak kullanılan

maliyet fonksiyonu çapraz entropi veya log kaybıdır. Lojistik regresyon, spam e-posta tespiti, hastalık tahmini, kredi riski değerlendirmesi ve daha fazlası dahil olmak üzere çeşitli uygulamalarda yaygın olarak kullanılmaktadır. Giriş özellikleri ile ikili bir sonucun olasılığı arasındaki ilişki hakkında bilgi sağlayabilen basit ve yorumlanabilir bir modeldir. Lojistik regresyonun uzantıları arasında çok sınıflı sınıflandırma için multinomial lojistik regresyon ve sıralı kategoriler için ordinal lojistik regresyon yer alır. Lojistik regresyon, ismine rağmen regresyondan ziyade sınıflandırma için doğrusal bir modeldir. Lojistik regresyon, literatürde logit regresyon, maksimum entropi sınıflandırması (MaxEnt) veya log-lineer sınıflandırıcı olarak da bilinir. Bu modelde, tek bir deneyin olası sonuçlarını tanımlayan olasılıklar lojistik bir fonksiyon kullanılarak modellenir (Sayad,2023).

Lojistik regresyon modelinin temel formülü ve adımları aşağıda verilmiştir:

$$P(y=1|x)=\frac{1}{1+e^{-(\beta_0+\beta_1x_1+\beta_2x_2+\cdots+\beta_px_p)}} \quad \textbf{(2.25)}$$

(2.25) bağımlı değişkenin belirli bir sınıfa ait olma olasılığını hesaplar.

$P(y=1|x)$ : Bağımlı değişkenin 1 olma olasılığı

$\beta_0$ : Y-ekseni kesişimi (intercept)

$\beta_i$: Model parametreleri (i = 1, 2, ..., p)

$x_i$ : Bağımsız değişkenler

Sigmoid Fonksiyonu: Lojistik regresyon, bağımlı değişkenin olasılığını tahmin etmek için sigmoid fonksiyonunu kullanır.

$$\sigma(z)=\frac{1}{1+e^{-z}}$$

Doğrusal Kombinasyon: Modelin tahminini yapmak için giriş değişkenleri ($x_i$) ve ağırlıklar ($w_i$) kullanılarak doğrusal bir kombinasyon hesaplanır:

$$z=w_0+w_1x_1+w_2x_2+\cdots+w_nx_n \quad \textbf{(2.26)}$$

Olasılık Tahmini: Doğrusal kombinasyon sonucunu sigmoid fonksiyonuna uygulayarak olasılık tahmini yapılır:

$$P(y=1|x)=\sigma(z)=\frac{1}{1+e^{-(w_0+w_1x_1+w_2x_2+\cdots+w_nx_n)}} \quad \textbf{(2.27)}$$

Kayıp Fonksiyonu: Lojistik regresyon modeli, log-loss (lojistik kayıp) fonksiyonunu minimize ederek en iyi ağırlıkları bulur. Log-loss fonksiyonu şu şekilde tanımlanır:

$$L(y,\hat{y})=-\left[y\log(\hat{y})+(1-y)\log(1-\hat{y})\right] \quad \textbf{(2.28)}$$

Modelin Eğitimi: Modelin eğitimi sırasında ağırlıklar, gradyan iniş yöntemi kullanılarak güncellenir. Her bir ağırlık için güncelleme kuralı şu şekildedir:

$$w_i := w_i-\eta\frac{\partial L}{\partial w_i} \quad \textbf{(2.29)}$$

### 2.7.8. AdaBoost Algoritması

AdaBoost (Adaptive Boosting), topluluk (ensemble) yöntemlerinden biridir ve zayıf öğrenicileri güçlü bir öğreniciye dönüştürmeyi amaçlar. Bu algoritma, her bir öğrenicinin hatalarını dikkate alarak ağırlıkları ayarlayarak çalışır. AdaBoost, özellikle sınıflandırma problemlerinde kullanılır ve yaygın olarak karar ağaçları gibi zayıf öğrenicilerle birlikte kullanılır.

Algoritmanın adımları aşağıda verilmiştir.

1. Başlangıçta tüm örneklere eşit ağırlık verilerek başlatılır.
2. Her bir adımda, bir zayıf öğrenici eğitilir.
3. Eğitilen öğrenicinin hatalarına göre ağırlıklar güncellenir.
4. Bu adımlar belirli bir iterasyon sayısı boyunca tekrarlanır.
5. Nihai sınıflandırma, tüm zayıf öğrenicilerin ağırlıklı oylarına dayanarak yapılır.

Adım 1: Ağırlıkların Başlatılması: Başlangıçta tüm örneklere eşit ağırlık verilir:

$$w_i^{(1)}=\frac{1}{N} \quad \textbf{(2.30)}$$

$w_i^{(1)}$ : İlk iterasyondaki i-inci örneğin ağırlığı

$N$ : Toplam örnek sayısı

Adım 2: Zayıf Öğrenicinin Eğitilmesi: Her bir iterasyonda bir zayıf öğrenici eğitilir ve hata oranı hesaplanır:

$$\grave{o}_m = \sum_{i=1}^{N} w_i^{(m)} \cdot I(y_i \neq h_m(x_i)) \quad \textbf{(2.31)}$$

$\grave{o}_m$ : m-inci zayıf öğrenicinin hata oranı

$I(\cdot)$: Gösterge fonksiyonu, doğruysa 1, yanlışsa 0 döner

$y_i$ : Gerçek etiket

$h_m(x_i)$ : m-inci zayıf öğrenicinin i-inci örnek için tahmini

Adım 3: Öğrenicinin Ağırlığının Hesaplanması: Zayıf öğrenicinin ağırlığı, hata oranına bağlı olarak hesaplanır:

$$\alpha_m = \frac{1}{2} \ln\left(\frac{1-\grave{o}_m}{\grave{o}_m}\right) \quad \textbf{(2.32)}$$

$\alpha_m$ : m-inci zayıf öğrenicinin ağırlığı

Adım 4: Ağırlıkların Güncellenmesi: Örnek ağırlıkları, zayıf öğrenicinin performansına göre güncellenir:

$$w_i^{(m+1)} = w_i^{(m)} \cdot \exp\left(-\alpha_m \cdot y_i \cdot h_m(x_i)\right) \quad \textbf{(2.33)}$$

Ağırlıkların normalizasyonu:

$$w_i^{(m+1)} = \frac{w_i^{(m+1)}}{\sum_{j=1}^{N} w_j^{(m+1)}} \quad \textbf{(2.34)}$$

Nihai Sınıflandırma: Tüm zayıf öğrenicilerin ağırlıklı oylarına dayanarak nihai sınıflandırma yapılır:

$$H(x) = \text{sign}\left(\sum_{m=1}^{M} \alpha_m \cdot h_m(x)\right) \quad \textbf{(2.35)}$$

$H(x)$ : Nihai sınıflandırma sonucu

AdaBoost algoritması, zayıf öğrenicilerin bir araya getirilerek güçlü bir öğrenici oluşturulmasını sağlar. Bu yöntem, özellikle sınıflandırma problemlerinde oldukça etkilidir ve yaygın olarak kullanılmaktadır.

## 2.8. Kümeleme Algoritmaları

### 2.8.1. K Ortalamalar

K-ortalamalar, makine öğrenimi ve denetimsiz öğrenmede popüler bir kümeleme algoritmasıdır. Bir dizi veri noktasını benzerliklerine göre kümeler halinde gruplamak için kullanılır. K-ortalamalar kümelemesinin amacı, verileri K kümelerine ayırmaktır; burada K, kullanıcı tanımlı bir parametredir. K-ortalamalar algoritması şu şekilde çalışır:

1. *Başlatma:* K adet ilk küme merkezini seçin. Bu merkezler rastgele seçilmiş veri noktaları veya özellik uzayında önceden tanımlanmış konumlar olabilir.
2. *Atama:* Her veri noktasını en yakın küme merkezine atayın. Bu genellikle her veri noktası ile merkezler arasındaki Öklid mesafesi (veya diğer mesafe ölçütleri) hesaplanarak ve en yakın merkeze sahip küme seçilerek yapılır.
3. *Güncelleme:* Her kümeye atanan tüm veri noktalarının ortalamasını alarak küme merkezlerini yeniden hesaplayın. Bu, merkez noktaları kendi kümelerinin merkezine taşır.
4. *Tekrarla:* 2. ve 3. adımlar bir durdurma kriteri karşılanana kadar iteratif olarak tekrarlanır. Yaygın durdurma kriterleri arasında maksimum iterasyon sayısı veya merkezlerin artık önemli ölçüde değişmemesi yer alır.
5. *Çıktı:* Nihai küme merkezleri K kümelerinin merkezlerini temsil eder ve her veri noktası bu kümelerden birine atanır.

K-ortalamalar küme içi varyansı en aza indirmeyi amaçlar, yani bir küme içindeki veri noktalarını olabildiğince benzer hale getirmeye çalışırken farklı kümeleri olabildiğince farklı tutmaya çalışır. Yinelemeli bir optimizasyon algoritmasıdır ve nihai küme atamaları başlangıç merkezlerine ve verilerin özel dağılımına bağlıdır.

K-ortalamalar, görüntü segmentasyonu, müşteri segmentasyonu, anomali tespiti ve daha fazlası gibi çeşitli uygulamalarda yaygın olarak kullanılmaktadır. Bununla birlikte, ilk merkez yerleşimine duyarlılık ve küme sayısını (k) önceden belirleme

ihtiyacı gibi bazı sınırlamaları vardır. Bu sorunların bazılarını ele alan K-means++ gibi K-means varyasyonları da vardır.

$$J = \sum_{i=1}^{k} \sum_{j=1}^{n} || x_j^{(i)} - \mu_i ||^2 \quad \textbf{(2.36)}$$

*J*: Amaç fonksiyonu

k: Kümelerin sayısı

x: Veri noktası

$\mu_i$ :i-inci kümenin merkezi

### 2.8.2. Hiyerarşik Kümeleme

Hiyerarşik kümeleme, kümeleri art arda birleştirerek veya ayırarak iç içe kümeler oluşturan genel bir kümeleme algoritmaları ailesidir. Bu hiyerarşik küme bir ağaç (veya dendrogram) olarak temsil edilir. Ağacın kökü, tüm örnekleri içeren benzersiz kümedir; yapraklar ise her biri yalnızca bir örnek içeren kümelerdir. Hiyerarşik kümeleme, bir küme hiyerarşisi oluşturmak için denetimsiz öğrenmede kullanılan bir makine öğrenimi tekniğidir. Veri noktalarının benzerliklerine göre art arda daha küçük veya daha büyük gruplar halinde birleştirildiği, dendrogram olarak da bilinen ağaç benzeri bir küme yapısı oluşturur. Hiyerarşik kümeleme iki tür olarak sınıflandırılabilir: Aglomeratif (aşağıdan yukarıya) ve Divisive (yukarıdan aşağıya). Aglomeratif Hiyerarşik Kümeleme, hiyerarşik kümelemenin en yaygın kullanılan türüdür. Her veri noktasının kendi kümesi olmasıyla başlar ve tüm veri noktalarını kapsayan tek bir küme kalana kadar en yakın kümeleri tekrar tekrar birleştirir. Aglomeratif hiyerarşik kümeleme için adımlar aşağıdaki gibidir:

a- Her veri noktasını tek bir küme olarak başlatın.

b- Tüm küme çiftleri arasındaki mesafeyi veya farklılığı hesaplayın (örneğin, Öklid mesafesi, Ward's bağlantısı, tek bağlantı, tam bağlantı vb. kullanarak).

c- En yakın iki kümeyi yeni bir kümede birleştirin.

d- Yalnızca bir küme kalana kadar b ve c adımlarını tekrarlayın.

Divisive hiyerarşik kümeleme tek bir kümedeki tüm veri noktalarıyla başlar ve her veri noktası kendi kümesinde olana kadar bunları özyinelemeli olarak daha küçük kümelere böler. Bu yöntem pratikte daha az kullanılmaktadır. Hiyerarşik kümelemenin

avantajlarından biri, dendrogram aracılığıyla verilerin doğal gruplandırmasının görsel bir temsilini sağlamasıdır. Bu, çeşitli seviyelerdeki kümelerin hiyerarşisini anlamada yardımcı olabilir.

Hiyerarşik kümeleme, verileriniz için en uygun küme sayısını belirlemek üzere dendrogramda belirli bir seviye seçmenize olanak tanıdığından keşifsel veri analizi için kullanışlı bir tekniktir. Hiyerarşik kümeleme, K-ortalamalar gibi diğer kümeleme algoritmalarının ortak bir sınırlaması olan küme sayısının önceden belirtilmesini gerektirmez. Bununla birlikte, özellikle büyük veri kümeleri için hesaplama açısından pahalı olabilir ve bağlantı yöntemi ve mesafe metriği seçimi sonuçları önemli ölçüde etkileyebilir. Yığılmalı küme, eşit olmayan küme boyutlarıyla sonuçlanan bir "zengin daha zengin olur" davranışı sergiler. Bu bağlamda, tek bağlantı en kötü stratejidir ve Ward en tutarlı sonuçları sağlar. Ancak yakınlık (veya kümelemede kullanılan mesafe) Ward ile değiştirilemez. Öklid dışı metrikler için ortalama bağlantı iyi bir alternatiftir. Tek bir bağlantı da küresel olmayan verilerde iyi performans gösterebilir (Sayad, 2023).

$$N_{\grave{o}}(x) = \{y \in D \mid d(x,y) \leq \grave{o}\} \qquad \textbf{(2.37)}$$

$$d(C_i, C_j) = \min\{d(x,y) : x \in C_i, y \in C_j\}$$

$C_i, C_j$ : Kümeler

$d(x,y)$ : İki veri noktası arasındaki mesafe

### 2.8.3. Yoğunluk Tabanlı Kümeleme

Gürültülü Uygulamaların Yoğunluk Tabanlı Mekânsal Kümelenmesi (Density-Based Spatial Clustering of Applications with Noise-DBSCAN), makine öğrenimi ve veri madenciliğinde kullanılan popüler bir kümeleme algoritmasıdır. Özellikle farklı şekil ve yoğunluklara sahip verilerdeki kümeleri bulmak için etkilidir. Ester et al, tarafından 1996 yılında geliştirilen DBSCAN, veri noktalarının yoğun bölgelerini kümeler olarak tanımlarken aynı zamanda aykırı değerleri gürültü noktaları olarak ayırt ederek çalışır. DBSCAN önceden belirlenmiş bir küme sayısı gerektirmez. Bunun yerine, iki parametreye dayalı olarak çalışır: Epsilon (ε) parametresi, yakındaki veri noktalarının aranacağı yarıçapı tanımlar. Komşuluk mesafesi olarak da bilinir. MinPoints (MinPts) parametresi, bir noktanın çekirdek nokta olarak kabul edilmesi için ε-komşuluğu içinde olması gereken minimum veri noktası sayısını belirtir.

Çekirdek nokta, ε-komşuluğunda en az MinPts veri noktası bulunan bir veri noktasıdır. Başka bir deyişle, yoğun bir bölge oluşturmak için etrafında yeterince veri noktası vardır. Bir sınır noktasının kendisi bir çekirdek nokta değildir, ancak bir çekirdek noktanın ε-komşuluğu içinde yer alır. Bir kümenin sınırında yer alır. Ne çekirdek noktası ne de sınır noktası olan veri noktaları gürültü noktaları olarak sınıflandırılır. Bunlar genellikle aykırı değerler olarak kabul edilir.

Algoritma aşağıdaki gibi ilerler:

- Rastgele, ziyaret edilmemiş bir veri noktasıyla başlayın.
- Nokta bir çekirdek noktasıysa, yeni bir küme oluşturun ve erişilebilir tüm noktaları (hem çekirdek hem de sınır noktaları) kümeye ekleyin.
- Kümeye daha fazla çekirdek noktası eklenemeyene kadar devam edin.
- Tüm veri noktaları kümelere atanana ya da gürültü olarak işaretlenene kadar ziyaret edilmemiş bir veri noktasıyla işlemi tekrarlayın.

Özetle, DBSCAN, özellikle farklı şekil ve yoğunluklarda kümeler içeren veri kümeleriyle çalışırken birçok kümeleme görevi için oldukça yararlı olabilecek güçlü bir yoğunluk tabanlı kümeleme algoritmasıdır. Başarılı bir uygulama için doğru parametre seçimi ve verilerinizin anlaşılması çok önemlidir. Yüksek yoğunluklu bölgelerin düşük yoğunluklu alanlardan ayrıldığı algoritma kümeleridir. DBSCAN tarafından bulunan kümeler, kümelerin dışbükey şekilli olduğunu varsayan k means'in aksine herhangi bir şekilde olabilir. DBSCAN'in merkezi bileşeni, yüksek yoğunluklu alanlarda bulunan örnekler olan çekirdek örnekler kavramıdır. Algoritmanın iki parametresi vardır: min_samples ve eps. (Eps parametresi, bir noktanın komşularını seçerken dikkate alacağı maksimum mesafeyi temsil eder). Daha yüksek min_samples değerleri veya daha düşük eps değerleri, bir küme oluşturmak için gereken daha yüksek bir yoğunluğu gösterir.

$$\text{Core Point} \quad \Rightarrow \quad \left|N_{\text{ö}}(p)\right| \geq \text{MinPts} \tag{2.38}$$

ϵ: Komşuluk yarıçapı

MİnPts: Bir küme oluşturmak için gereken minimum nokta sayısı

### 2.8.4. Gaussian Karışım Modelleri

Gaussian Karışım Modelleri (Gaussian Mixture Models-GMM), verilerin birden fazla Gaussian dağılımının karışımı olarak modellenmesini sağlar. Bu yöntem, verilerin alt gruplara ayrılmasını ve her bir alt grubun kendi Gaussian dağılımı ile temsil edilmesini mümkün kılar. GMM, genellikle kümeleme problemlerinde kullanılır ve Expectation-Maximization (EM) algoritması ile optimize edilir (Patacchiola, 2020).

GMM'nin Matematiksel Temeli: Bir GMM, k adet Gaussian dağılımın karışımından oluşur. Her bir dağılımın ağırlığı, ortalaması ve kovaryans matrisi bulunur. GMM'nin olasılık yoğunluk fonksiyonu (2.39)'da verilmiştir:

$$p(\mathbf{x} \mid \lambda) = \sum_{k=1}^{K} \pi_k \mathrm{N}\ (\mathbf{x} \mid \mu_k, \Sigma_k) \tag{2.39}$$

$\pi_k$ :k-inci bileşenin karışım ağırlığı

$\mathrm{N}\ (\mathbf{x} \mid \mu_k, \Sigma_k$ : k-inci bileşenin Gaussian olasılık yoğunluk fonksiyonu

$\mu_k$ : k-inci bileşenin ortalama vektörü

$\Sigma_k$ :k-inci bileşenin kovaryans matrisi

x: Gözlemler

$\lambda$ : Model parametreleri ( $\pi_k, \mu_k, \Sigma_k$ )

GMM parametrelerini tahmin etmek için EM algoritması kullanılır. EM algoritması iki adımda çalışır: Expectation (E) adımı ve Maximization (M) adımı.

E Adımı: E adımında, her bir veri noktasının her bir bileşene ait olma olasılığı (sorumluluk) hesaplanır:

$$\gamma(z_{ik}) = \frac{\pi_k \mathrm{N}\ (\mathbf{x}_i \mid \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathrm{N}\ (\mathbf{x}_i \mid \mu_j, \Sigma_j)} \tag{2.40}$$

$\gamma(z_{ik})$: i-inci veri noktasının k-ıncı bileşene ait olma olasılığı

M Adımı: M adımında, sorumluluklar kullanılarak model parametreleri güncellenir:

$$\pi_k = \frac{1}{N}\sum_{i=1}^{N}\gamma(z_{ik}) \tag{2.41}$$

$$\mu_k = \frac{\sum_{i=1}^{N}\gamma(z_{ik})\mathbf{x}_i}{\sum_{i=1}^{N}\gamma(z_{ik})} \tag{2.42}$$

$$\Sigma_k = \frac{\sum_{i=1}^{N}\gamma(z_{ik})(\mathbf{x}_i - \mu_k)(\mathbf{x}_i - \mu_k)^T}{\sum_{i=1}^{N}\gamma(z_{ik})} \tag{2.43}$$

GMM, veri kümesindeki gizli yapıları ortaya çıkarmak için güçlü bir yöntemdir ve EM algoritması ile optimize edilerek parametre tahminleri yapılır (Towards Data Science, 2020).

### 2.8.5. Spektral Kümeleme

Spektral Kümeleme (Spectral Clustering), makine öğrenmesi ve grafik teorisinde kullanılan, veri noktalarını örtüşmeyen kümelere ayırmak için kullanılan bir tekniktir. Bu algoritma, veri noktaları arasındaki bağlantıları kullanarak kümeleri oluşturur ve genellikle veri noktalarının benzerlik matrisinin spektrumunu (özdeğerlerini) kullanır.

Algoritmanın adımlarını sırayla yazalım.

*Benzerlik Matrisi Oluşturma:* Veriler arasındaki benzerlikleri ölçmek için bir benzerlik matrisi (S) oluşturulur. Bu matris, genellikle Gaussian benzerlik fonksiyonu gibi yöntemlerle hesaplanır.

$$S_{ij} = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) \tag{2.44}$$

*Laplacian Matrisi Hesaplama*: Benzerlik matrisinden dereceler matrisi (D) hesaplanır ve ardından Laplacian matrisi (L) elde edilir:

$$D_{ii} = \sum_j S_{ij} \tag{2.45}$$

$$L = D - S \tag{2.46}$$

*Özvektörlerin Hesaplanması:* Laplacian matrisinin en küçük k özdeğerine karşılık gelen k özvektör (u1,u2,..., uk) hesaplanır. Bu özvektörler, veri noktalarını yeni bir uzaya yansıtır.

*Kümeleme:* Özvektörler matrisi (U), her bir satırı bir veri noktasına karşılık gelen yeni bir temsilci matrisidir. Bu matriste, K-means gibi bir kümeleme algoritması uygulanır.

*Sonuçların Yorumlanması*: K-means sonuçları, orijinal veri noktalarına uygulanarak her bir noktanın hangi kümeye ait olduğu belirlenir.

$$S_{ij} = \exp\left( -\frac{\|x_i - x_j\|^2}{2\sigma^2} \right) \quad \textbf{(2.47)}$$

# 3. ARAŞTIRMA VE BULGULAR

## 3.1. Evren ve Örneklem

Regresyon problemi için kullanılan veri seti Automobile price data _Raw_ .csv, Azure ML'de bulunan veri setlerinden alınmıştır.

Veri setinde 205 kayıt bulunmaktadır. Bu kayıtlar, farklı araçların çeşitli özelliklerini içermektedir. 25 adet bağımsız değişken (özellik) ve 1 adet bağımlı (hedef) değişken vardır.

Veriye ait değişkenler;

*Symboling:* Sigorta risk derecelendirmesi; -3 (düşük risk) ile +3 (yüksek risk) arasında değişir.

*Normalized-losses:* Belirli bir araç modelinin sigorta tazminat ödemelerine göre normalleştirilmiş kayıp oranı.

*Make:* Aracın markası veya üreticisi (Örneğin; Honda, Toyota).

*Fuel-type:* Aracın kullandığı yakıt türü (Örneğin; benzin, dizel).

*Aspiration:* Motorun hava emiş türü (Örneğin; turbo, doğal emiş).

*Num-of-doors:* Aracın kapı sayısı (Örneğin; iki kapılı, dört kapılı).

*Body-style:* Aracın gövde tipi (Örneğin; sedan, hatchback).

*Drive-wheels:* Çekiş sistemi (Örneğin; önden çekiş, arkadan çekiş, dört çeker).

*Engine-location:* Motorun konumu (Örneğin; ön, arka).

*Wheel-base:* Aracın aks mesafesi (iki dingil arasındaki mesafe).

*Length:* Aracın uzunluğu.

*Width:* Aracın genişliği.

*Height:* Aracın yüksekliği.

*Curb-weight:* Aracın boş ağırlığı (yüklü değilken).

*Engine-type:* Motor tipi (Örneğin; SOHC, DOHC).

*Num-of-cylinders:* Motorun silindir sayısı.

*Engine-size:* Motor hacmi (genellikle litre veya cc cinsinden).

*Fuel-system:* Yakıt sistemi türü (Örneğin; karbüratör, enjeksiyon).

*Bore:* Silindir çapı.

*Stroke:* Pistonun silindir içinde hareket ettiği mesafe.

*Compression-ratio:* Motorun sıkıştırma oranı.

*Horsepower:* Motorun beygir gücü.

*Peak-rpm:* Motorun maksimum devir hızı (devir/dakika cinsinden).

*City-mpg:* Şehir içi yakıt tüketimi (mil/galon cinsinden).

*Highway-mpg*: Şehir dışı yakıt tüketimi (mil/galon cinsinden).

*Price:* Aracın satış fiyatı. Hedef değişken.

Aracın fiyat tahmini için Linear Regresyon, Lasso Regresyon ve Ridge Regresyon algoritmaları kullanılmıştır.

İkinci uygulama olarak sınıflandırma problemi için kullanılan "WA_Fn-UseC_-Telco-Customer-Churn.csv" veri seti Kaggle ve IBM'in örnek veri seti koleksiyonundan alınmıştır. Veriye ait web adresi: https://www.kaggle.com/datasets/blastchar/telco-customer-churn.

Verinin değişkenleri ve işlevleri aşağıda belirtilmiştir;

*Churn:* Müşterinin vazgeçip vazgeçmediği (Evet veya Hayır)

*PhoneService:* Müşterinin telefon hizmetinin olup olmadığı (Evet veya Hayır)

*MultipleLines:* Müşterinin birden fazla hattının olup olmadığı (Evet, Hayır, Telefon hizmeti yok)

*InternetService:* Müşterinin sahip olduğu bir tür internet hizmeti (DSL, Fiber Optik, Hayır)

*OnlineSecurity:* Müşterinin çevrimiçi güvenliğinin olup olmadığı (Evet, Hayır, İnternet Hizmeti Yok)

*OnlineBackup:* Müşterinin çevrimiçi yedeğinin olup olmadığı (Evet, Hayır, İnternet Hizmeti Yok)

*DeviceProtection:* Müşterinin cihaz korumasına sahip olup olmadığı (Evet, Hayır, İnternet Hizmeti Yok)

*TechSupport:* Müşterinin teknik desteğe sahip olup olmadığı (Evet, Hayır, İnternet Hizmeti Yok)

*StreamingTV:* Müşterinin yayın yapan bir TV'si olup olmadığı (Evet, Hayır, İnternet Hizmeti Yok)

*StreamingMovies:* Müşterinin film akışı olup olmadığı (Evet, Hayır, İnternet Hizmeti Yok)

*Tenure:* Müşterinin şirkette ne kadar süre kaldığı

*Contrat :* Müşterinin sahip olduğu sözleşme türü (Aydan Aya, Bir Yıllık, İki Yıllık)

*PaperlessBilling :* Müşterinin kağıtsız faturalandırması olup olmadığı (Evet, Hayır)

*PaymentMethod :* müşteri tarafından kullanılan ödeme yöntemi (Elektronik çek, Posta çeki, Banka havalesi (otomatik), Kredi kartı (otomatik)

*MonthlyCharges :* Müşteriden aylık olarak tahsil edilen tutar

*TotalCharges :* Müşteriden tahsil edilen toplam tutar

*Müşteri Kimliği :* Her müşteri için benzersiz değer

*Cinsiyet :* Her müşterinin cinsiyet türü (Kadın, Erkek)

*SeniorCitizen :* Müşterinin yaşlı olup olmadığı (Evet, Hayır)

*Partner :* Müşterinin ortağı olup olmadığı (Evet, Hayır)

*Dependents :* Müşterinin bağlı olduğu aile bireyleri var mı yok mu sorusuna cevap veren değişken (Evet, Hayır)

Müşterilerin terk edip terk etmeme durumlarını modellemek amacıyla GBM, Lojistik Regresyon, Karar Ağaçları, Rastgele Orman algoritmaları kullanılmıştır.

Diğer uygulama için Cilt Kanseri Görüntü Sınıflandırmada kullanılan veri setleri Isic arşiv web sitesinden indirilmiştir (https://www.isic-archive.com /#!/topWithHeader/onlyHeaderTop/gallery?filter=%5B%5D). Seçilen üç farklı cilt kanseri türüyle ilgili toplam 285 görüntü vardır. Farklı cilt kanseri türlerine ilişkin bilgiler Tablo 3.1.'de verilmiştir:

Tablo 3.1. Cilt kanseri görüntü verisi sınıfları

| | | |
|---|---|---|
| nv | Nevus, melanositic nevies | 125 images |
| vasc | Vasculer | 72 images |
| df | Dermatofibroma | 88 images |

### 3.2. Uygulama 1: Araba Fiyat Tahmini

Araçların verilen özelliklerinden yararlanarak fiyat tahminlerini yapabilmek amaçlanmaktadır.

#### 3.2.1. Bağımlı (Hedef) Değişken

Bağımlı (hedef) değişken olarak adlandırılan "price" değişkeni, bu veri setinde diğer özelliklerin değerlerinden yola çıkarak tahmin edilen bir değeri temsil eden fonksiyondur. Price değişkeni, aracın motor hacmi, silindir çapı, pistonun silindir içinde hareket ettiği mesafe, motorun sıkıştırma oranı, motorun beygir gücü, motorun maksimum devir hızı, şehir içi yakıt tüketimi, şehir dışı yakıt tüketimi, belirli bir araç modelinin sigorta tazminat ödemelerine göre normalleştirilmiş kayıp oranı, sigorta risk derecelendirmesi gibi özelliklerin kombinasyonu ile tahmin edilir.

#### 3.2.2. Bağımsız Değişkenler (Özellikler)

Bağımsız değişkenler, bir veri setinde araba fiyatının sonucunu belirleyen veya etkileyen değişkenlerdir. Örneğin, bir aracın fiyatını tahmin etmek için kullanılabilirler. Veri tabanındaki bağımsız değişkenleri şu şekilde tanımlayabiliriz:

*Demografik Değişkenler:* Demografik değişkenler, araçların genel özelliklerini ve sınıflandırmalarını temsil eder. Bu değişkenler, araçların çeşitli fiziksel ve performans özelliklerini içerir.

Length: Aracın uzunluğu, genellikle inç veya santimetre cinsinden ölçülür.

Width: Aracın genişliği, inç veya santimetre cinsinden ölçülür.

Height: Aracın yüksekliği, inç veya santimetre cinsinden ölçülür.

Wheel-base: Aracın aks mesafesi, yani iki dingil arasındaki mesafe.

Curb-weight: Aracın boş ağırlığı, araç yüklenmemişkenki ağırlığıdır.

*Kategorik Değişkenler:* Kategorik değişkenler, belirli kategorilere veya gruplara ayrılmış verilerdir. Bu değişkenler, araçların farklı sınıflarını veya türlerini tanımlamak için kullanılır.

Make: Aracın markası veya üreticisi (Örneğin; Honda, Toyota).

Fuel-type: Aracın kullandığı yakıt türü (Örneğin; benzin, dizel).

Aspiration: Motorun hava emiş türü (Örneğin; turbo, doğal emiş).

Num-of-doors: Aracın kapı sayısı (Örneğin; iki kapılı, dört kapılı).

Body-style: Aracın gövde tipi (Örneğin; sedan, hatchback).

Drive-wheels: Çekiş sistemi (Örneğin; önden çekiş, arkadan çekiş, dört çeker).

Engine-location: Motorun konumu (Örneğin; ön, arka).

Engine-type: Motor tipi (Örneğin; SOHC, DOHC).

Num-of-cylinders: Motorun silindir sayısı.

Fuel-system: Yakıt sistemi türü (Örneğin; karbüratör, enjeksiyon).

*Sayısal Değişkenler:* Bu değişkenler, araçların performans ve mekanik özelliklerini ölçen sayısal değerlerdir.

Engine-size: Motor hacmi (genellikle litre veya cc cinsinden).

Bore: Silindir çapı.

Stroke: Pistonun silindir içinde hareket ettiği mesafe.

Compression-ratio: Motorun sıkıştırma oranı.

Horsepower: Motorun beygir gücü.

Peak-rpm: Motorun maksimum devir hızı (devir/dakika cinsinden).

City-mpg: Şehir içi yakıt tüketimi (mil/galon cinsinden).

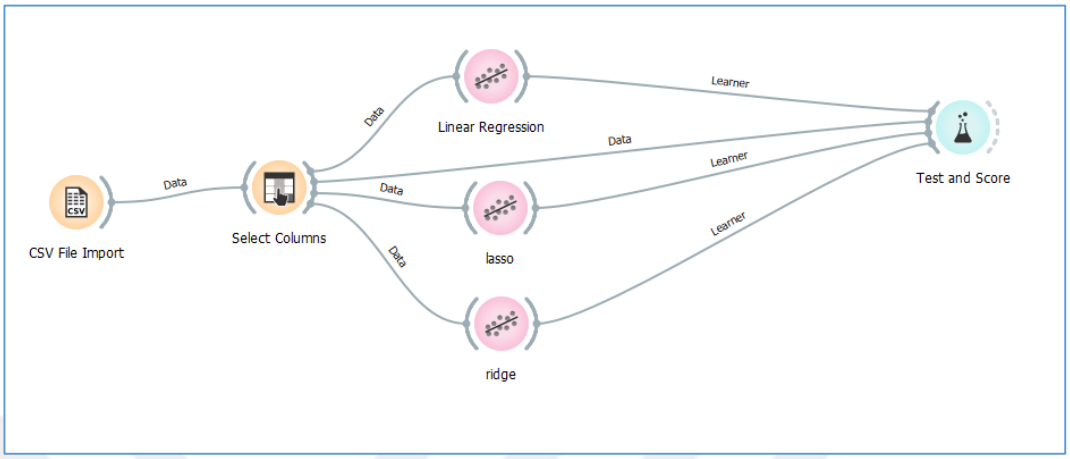
Highway-mpg: Şehir dışı yakıt tüketimi (mil/galon cinsinden).

Normalized-losses: Belirli bir araç modelinin sigorta tazminat ödemelerine göre normalleştirilmiş kayıp oranı.

Symboling: Sigorta risk derecelendirmesi; -3 (düşük risk) ile +3 (yüksek risk) arasında değişir.

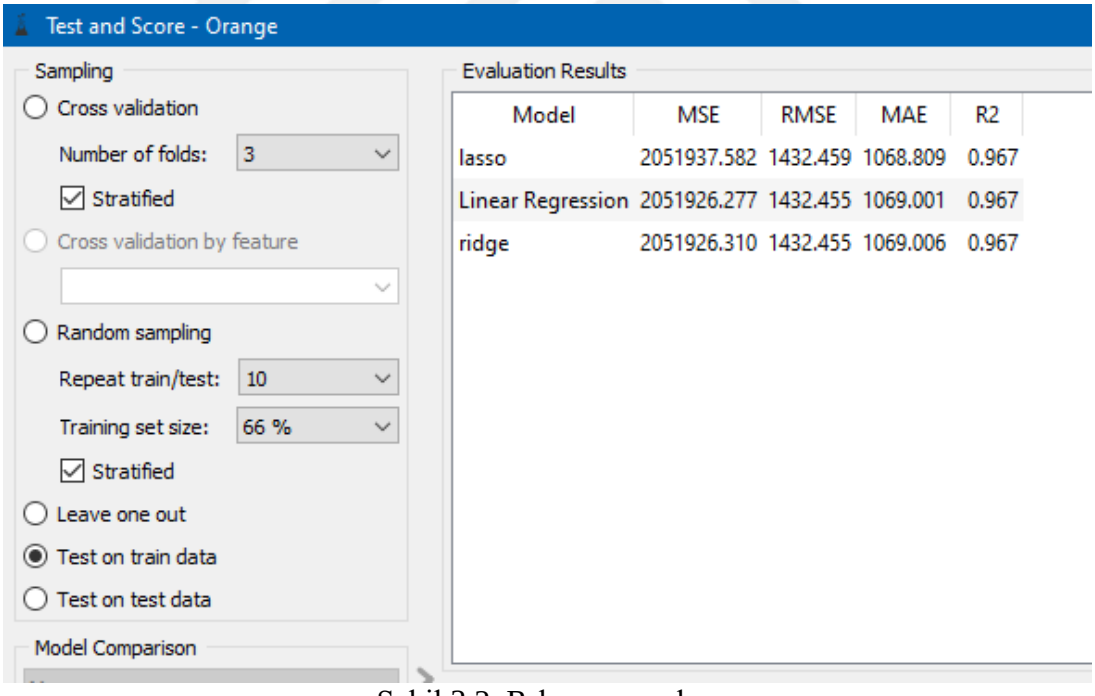
### 3.2.3. Verilerin analizi

Linear Regresyon, Ridge Regresyon ve Lasso Regresyon algoritmalarıyla veri Orange programında analiz edilmiştir.




Şekil 3.1. Orange Programı Regresyon Modelleri Uygulama

Veriler analiz edildikten sonra çıkan üç algoritma için de R-kare sonucu 0,967 çıkmıştır.




Şekil 3.2. R-kare sonuçları

Aynı veri Azure ML platformunda analiz edildiğinde Linear Regresyona ait R-kare sonucu 0,96 olarak çıkmıştır.

Tablo 3.2. Yeni Girilen Kayıtlar İçin Algoritmalara Ait Tahmin Değerleri

| price | Linear | lasso | ridge | symboling | wheel-base | horsepower | peak-rpm | city-mpg |
|---|---|---|---|---|---|---|---|---|
| 30760 | 31872.8 | 31872.2 | 31872.7 | 0 | 103.5 | 182 | 5400 | 16 |
| 9549 | 11261.9 | 11257.6 | 11261.7 | 0 | 97.2 | 97 | 5200 | 27 |
| 11850 | 12548 | 12546.3 | 12548.2 | 3 | 99.1 | 110 | 5250 | 21 |
| 7799 | 6794.13 | 6796.14 | 6794.13 | 1 | 94.5 | 69 | 5200 | 31 |
| 7788 | 7862.25 | 7860.55 | 7862.38 | 0 | 95.7 | 56 | 4500 | 38 |
| 9258 | 7724.88 | 7724.26 | 7724.68 | 0 | 95.7 | 70 | 4800 | 28 |
| 10198 | 9255.8 | 9256.27 | 9255.89 | 0 | 97 | 94 | 5200 | 25 |
| 7775 | 8441.25 | 8440.79 | 8441.31 | 2 | 97.3 | 52 | 4800 | 37 |
| 13295 | 12644.2 | 12650.2 | 12644.9 | 0 | 100.4 | 110 | 5500 | 19 |
| 8238 | 6758.37 | 6753.11 | 6758.16 | 1 | 94.5 | 70 | 4800 | 29 |
| 18280 | 15941.4 | 15936.5 | 15941.4 | 0 | 104.9 | 120 | 5000 | 19 |
| 9988 | 9396.81 | 9396.72 | 9396.64 | -1 | 102.4 | 92 | 4200 | 27 |
| 40960 | 42160.7 | 42162.3 | 42160.5 | 0 | 120.9 | 184 | 4500 | 14 |
| ? | 24257 | 24257.8 | 24256.6 | 0 | 99.5 | 160 | 5500 | 16 |
| 28248 | 28031.8 | 28033.9 | 28031.8 | -1 | 110 | 123 | 4350 | 22 |

### 3.3. Ugulama 2: Büyük Veri Analitiği Metodu İle Müşteri Kayıp Analizi

Müşterilerimizin bizi terk edip terk etmeme durumlarını modellemek amaçlanmaktadır.

#### 3.3.1. Bağımlı (Hedef) Değişken

Bağımlı (hedef) değişken olarak adlandırılan Churn değişkeni, bir veri setinde diğer değişkenlerin değerlerinden etkilenen ve genellikle incelenen olayın sonucunu belirleyen değişkendir. Genellikle bir sonucu tahmin etmek veya anlamak için diğer özellikleri (bağımsız değişkenleri) kullanır.

Örneğin, bir müşteri churn değişkeni üzerinden incelenirse, bu değişken müşterinin belirli bir süre içinde hizmeti terk edip etmediğini ifade edebilir. Bu durumda, Churn değişkeni "yes" değerini alıyorsa, müşteri hizmeti terk etmiş demektir; "no" değerini alıyorsa, müşteri hizmeti terk etmemiştir demektir.

Veritabanındaki örnek verileri kullanarak açıklayacak olursak, 7043 kayıttan 1869 tanesinin Churn değeri "Yes" ve 5174 kayıtın Churn değeri "No" ise:

"Yes" değerini alan 1869 kayıt, müşterilerin hizmeti terk ettiği durumları temsil eder.

"No" değerini alan 5174 kayıt ise, müşterilerin hizmeti terk etmedikleri durumları temsil eder.

Bu tanımlama ile Churn değişkeni, müşterilerin terk etme eğilimini veya terk etmeme eğilimini ifade eden bir bağımlı değişkendir.

### 3.3.2. Bağımsız Değişkenler (Özellikler)

Bağımsız değişkenler, bir veri setinde incelenen olayın sonucunu belirleyen veya etkileyen değişkenlerdir. Örneğin, bir müşterinin hizmeti terk etme olasılığını belirlemede kullanılabilirler. Veri tabanındaki bağımsız değişkenleri şu şekilde tanımlayabiliriz:

Müşterilerin kayıt esnasında verdiği bilgileri içeren kategorik değişkenler;

*PhoneService :* Müşterinin telefon servisi olup olmadığını belirten değişkendir. Eğer varsa “Yes”, yoksa “No” değeri alır. Müşterinin telefonunu kaydetmesi gerekir.

*MultipleLines :* Müşteri birden fazla hattı olduğunu belirtmişse “Yes”, birden fazla hat kaydı girmemişse “No” değerini alır.

*InternetService :* Müşterinin hangi tip internet servisine sahip olduğunu((DSL, Fiber Optic, No) belirten değişkendir.

*OnlineSecurity :* Müşteriye ait bir çevrimiçi güvenlik siteminin olup olmadığının bilgisini veren değişkendir. (Yes, No, No Internet Service)

*OnlineBackup :* Müşteriye ait bir çevrimiçi yedekleme sisteminin olup olmadığının bilgisini veren değişkendir. (Yes, No, No Internet Service)

*DeviceProtection :* Müşterinin veri korumasını bir cihazla yapıp yapmadığını belirten bir değişkendir. (Yes, No, No Internet Service)

*TechSupport :* Müşteri bir teknik desteğe sahip mi? Ya da bir teknik destek hizmeti alıyor mu sorusunu cevabını veren değişkendir. (Yes, No, No Internet Service)

*StreamingTV :* Müşteriye ait bir televizyon kanalı olup olmadığını gösteren bir değişkendir. (Yes, No, No Internet Service)

*StreamingMovies :* Müşteriye ait bir film olup olmadığını gösteren bir değişkendir. (Yes, No, No Internet Service)

Müşteri hesap bilgilerini içeren değişkenler:

*Tenure :* Müşterinin şirketle kaç aydır devam ettiğini gösteren sayısal bir değişkendir.

*Contract :* Müşterinin mevcut sözleşme türünü aydan aya, 1 yıllık, 2 yıllık şeklinde belirten değişkendir.

*PaperlessBilling :* Müşterinin kağıtsız faturalamayı seçip seçmediğini belirtir. (Yes, No)

*PaymentMethod :* Müşterinin faturasını nasıl ödediğini belirtir: Bankadan Para Çekme, Kredi Kartı, Posta Çeki.

*MonthlyCharges :* Müşterinin şirketten aldığı tüm hizmetler için geçerli toplam aylık ücretini gösterir.

*TotalCharges :* Müşterinin şirkette toplam kaldığı süre göz önünde bulundurularak hesaplanan toplam ücretini gösterir.

Demografik bilgileri içeren değişkenler;

*CustomerID :* Her müşteriye ayrı ayrı verilen kimlik.

*Gender :* Müşterinin cinsiyetini belirten değişken. Female (kadın), Male (erkek).

*SeniorCitizen :* Müşterinin 65 yaşının üstünde olup olmadığını belirten değişken. (Yes, No)

*Partner :* Müşterini ortağının olup olmadığını ifade eden değişkendir. Varsa"Yes",yoksa "No" değerini almıştır.

*Dependents :* Müşterinin bakmakla yükümlü olduğu kişilerle birlikte yaşayıp yaşamadığını belirtir: Evet, Hayır. Bağımlı kişiler çocuklar, ebeveynler, büyükanne ve büyükbabalar vb. olabilir. (Yes, No)

Bu bağımsız değişkenler, müşterilerin terk etme olasılıklarını değerlendirmek için kullanılabilir ve terk etme modelleri oluştururken dikkate alınabilirler.

### 3.3.3. Verilerin Analizi

Gradyan Arttırma Makinesi (Gradient Boosting Machine-GBM) ile Müşteri Terk Modellemesi, Lojistik Regresyon, Karar Ağacı Sınıflandırma Algoritması ve Rastgele Orman Sınıflandırma algoritması kullanılmıştır.

### 3.3.4. Müşteri Kayıp Analizi

Aşağıdaki ekran görüntüsü spark session oluşturan ve MACustomerChurn.csv dosyasını spark ortamına taşıyan kodları içermektedir.

```
import pyspark
from pyspark.sql import SparkSession
from pyspark.conf import SparkConf
from pyspark import SparkContext

spark = SparkSession.builder \
    .master("local") \
    .appName("churn_modellemesi") \
    .config("spark.executer.memory", "16gb") \
    .getOrCreate()



sc = spark.sparkContext
sc
```

**SparkContext**

Spark UI

**Version**
`v3.2.1`

**Master**
`local`

**AppName**
`churn_modellemesi`

```
spark_df = spark.read.csv("churn.csv",
                          header = True,
                          inferSchema = True,
                          sep = ",")
spark_df.cache()
```

```
DataFrame[_c0: int, Names: string, Age: double, Total_Purchase: double, Account_Manager: int, Years: double, Num_Sites: double,
Churn: int]
```

Şekil 3.3. Spark session

GBM ile Müşteri Terk Modellemesi uygulandığında aşağıdaki şekilde görüldüğü gibi algoritmanın doğruluk oranının yüzde 79 düzeyinde olduğu sonucu ortaya çıkmaktadır. Yani algoritmanın yüzde 79 oranında doğru sonuç ürettiği sonucu ortaya çıkmaktadır.

**GBM ile Müşteri Terk Modellemesi**

```
from pyspark.ml.classification import GBTClassifier

gbm = GBTClassifier(maxIter = 10, featuresCol = "features", labelCol = "label")

gbm_model = gbm.fit(train_df)

y_pred = gbm_model.transform(test_df)

y_pred
```

```
DataFrame[features: vector, label: int, rawPrediction: vector, probability: vector, prediction: double]
```

```
ac = y_pred.select("label","prediction")

ac.filter(ac.label == ac.prediction).count() / ac.count()
```

```
0.78954802259887
```

Şekil 3.4. GBM

Lojistik regresyon uygulandığında aşağıdaki şekilde görüldüğü gibi GBM algoritmasına yakın bir değer olan yüzde 78 çıkmaktadır.

```
from pyspark.ml.classification import LogisticRegression
loj = LogisticRegression(featuresCol = "features", labelCol = 'label', maxIter=10)
loj_model = loj.fit(train_df)
y_pred = loj_model.transform(test_df)
ac = y_pred.select("label", "prediction")
ac.filter(ac.label == ac.prediction).count() / ac.count()
```

```
0.7838983050847458
```

Şekil 3.5. Lojistik regresyon

Karar Ağacı Sınıflandırma Algoritması da aşağıdaki şekilde yer aldığı gibi yüzde 78 ile yakın bir sonuç değeri üretmiştir.

```
from pyspark.ml.classification import DecisionTreeClassifier
dt = DecisionTreeClassifier(featuresCol = 'features', labelCol = 'label', maxDepth = 3)
dt_model = dt.fit(train_df)
y_pred = dt_model.transform(test_df)
ac = y_pred.select("label", "prediction")
ac.filter(ac.label == ac.prediction).count() / ac.count()
```

```
0.782015065913371
```

Şekil 3.6. Karar ağacı

Rastgele Orman Sınıflandırma Algoritmasının yüzde 78'lik bir değer ürettiği aşağıda görülmektedir.

```
from pyspark.ml.classification import RandomForestClassifier
rf = RandomForestClassifier(featuresCol = 'features', labelCol = 'label')
rf_model = rf.fit(train_df)
y_pred = rf_model.transform(test_df)
ac = y_pred.select("label", "prediction")
ac.filter(ac.label == ac.prediction).count() / ac.count()
```

```
0.7848399246704332
```

Şekil 3.7. Rastgele orman

Algoritmaların, yeni müşteriler girdiğimizde müşteri kayıp (churn) verisinin hangi değerleri aldığı aşağıda gösterilmiştir.

Tablo 3.3. Algoritmaların ortak sonucu

| seniorcitizen | tenure | monthlycharges | prediction(churn) |
|---|---|---|---|
| 1 | 10 | 20 | 1 |
| 1 | 15 | 38 | 1 |
| 0 | 85 | 85 | 0 |
| 0 | 40 | 64 | 0 |
| 1 | 50 | 78 | 0 |

Bu verilere göre, müşterilerin kayıp (churn) durumlarına etki eden bazı faktörleri görebiliriz:

*SeniorCitizen-MonthlyCharges :* (Yaş ve Aylık Satın Alma): Daha genç müşteriler genellikle daha fazla satın alma yapar. Bu nedenle, yaşın ve aylık satın alma miktarının churn üzerindeki etkisi belirgin olabilir.

*Tenure(Şirkette Geçirilen Ay Sayısı):* Müşterilerin şirkette geçirdikleri ay sayısı da churn üzerinde etkilidir. Daha uzun süredir müşteri olanlar genellikle daha sadık olabilir ve churn oranı daha düşüktür.

*Contract(Abonelik Durumu):* Müşterilerin aylık abonelik seçimleri churn oranını artırabilir.

*Tahmin (Churn):* Verilerde tahmini churn sonuçları da var. Bu tahminler, veri analizi ve öngörü modelleri kullanılarak elde edilmiş olabilir ve müşteri kayıp riskini belirlemede önemli bir rol oynar.

Bu verileri kullanarak, müşteri kayıp oranlarını analiz edip, bu faktörleri göz önünde bulundurarak müşteri ilişkileri stratejileri geliştirilebilir. Örneğin, genç müşterilerle daha sık iletişim kurarak sadakatlerini artırabilir veya müşterilerin

abonelik süresini daha da uzatıp müşteri memnuniyetini artırarak churn oranları düşürülebilir.

### 3.4. Uygulama 3: Cilt Kanseri Görüntü Sınıflandırma ve Kümeleme

Orange programı kullanılarak bu görüntülere makine öğrenimi algoritmaları uygulanmıştır. Görüntüler üzerinde çeşitli sınıflandırma ve kümeleme algoritmaları uygulanmıştır. Görüntü verilerini inception-v3 tekniğini kullanarak analiz edip sayısallaştırdığımızda sınıflandırma algoritmaları başarılı sonuçlar vermiştir (Kara, 2023; Terzi, 2023; Cengiz, 2023).

#### 3.4.1. Cilt Kanseri Görüntü Sınıflandırma

Tablo 3.4. Sınıflandırma algoritmaları sonuçları

| Model | AUC | CA | F1 | Precision | Recall |
|---|---|---|---|---|---|
| KNN | 0.962 | 0,884 | 0,883 | 0,887 | 0,884 |
| DVM | 0,99 | 0,905 | 0,904 | 0,904 | 0,905 |
| Neural Network | 0,988 | 0,919 | 0,918 | 0,918 | 0,919 |

Tablo 3.4.'teki kısaltmaların anlamları şu şekildedir.

AUC (Area Under the Curve), ROC eğrisinin altındaki alanı temsil eder ve modelin sınıfları ayırt etme yeteneğini ölçer. Daha yüksek bir AUC, daha iyi model performansı gösterir. CA (Classification Accuracy), doğru tahmin edilen örneklerin toplam örnek sayısına oranıdır. Modelin genel doğruluğunu ölçer. F1 Skoru, kesinlik ve duyarlılığın harmonik ortalamasıdır. Kesinlik ve duyarlılık arasında bir denge sağlar ve sınıf dağılımının dengesiz olduğu durumlarda özellikle kullanışlıdır. Kesinlik (Precision), pozitif olarak tahmin edilen örnekler arasındaki gerçek pozitif örneklerin oranını gösterir. Pozitif tahminlerin doğruluğunu ölçer. Duyarlılık (Recall), aynı zamanda Duyarlılık ya da Gerçek Pozitif Oranı olarak da bilinir ve tüm gerçek pozitif örnekler arasındaki gerçek pozitif örneklerin oranını ölçer. Modelin tüm pozitif örnekleri yakalama yeteneğini yansıtır.

KNN modeli genel olarak iyi bir performans sergilemekte, ancak diğer modellerle karşılaştırıldığında tüm metriklerde en düşük değerlere sahiptir. Bu model, diğer modellere göre daha düşük doğruluk ve sınıflandırma performansı göstermektedir.

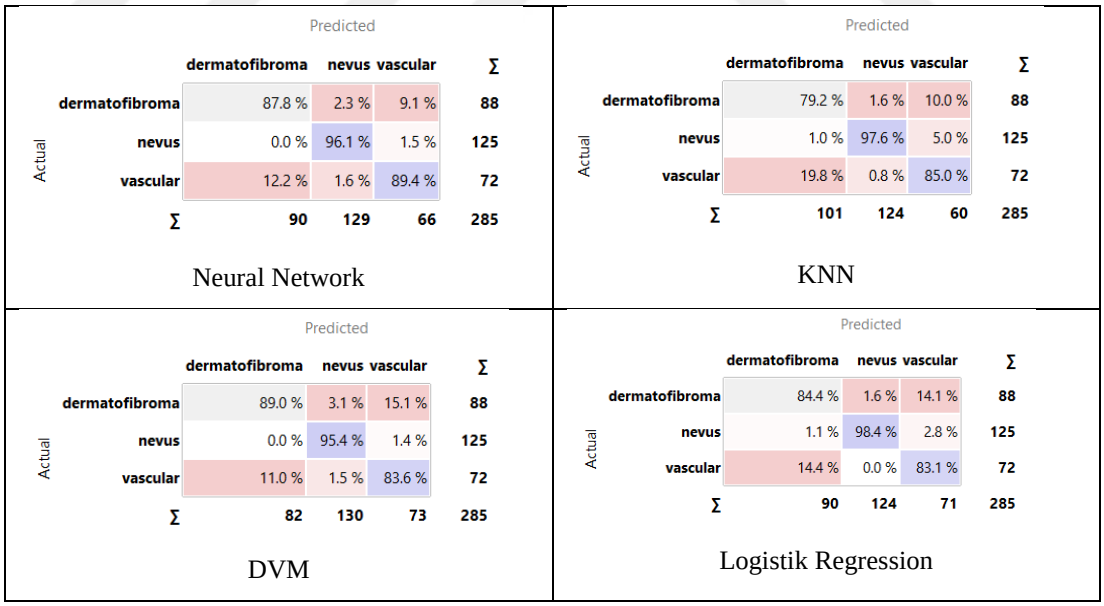
DVM modeli, yüksek bir AUC değeriyle güçlü bir sınıflandırma yeteneği gösterir. Doğruluk (CA) ve F1 skoru da yüksek, bu da modelin dengeli ve tutarlı bir performans sergilediğini gösterir. Precision ve Recall değerleri de yüksektir, bu da doğru pozitif tahminlerde iyi olduğunu gösterir.

Neural Network modeli, AUC, CA, F1, Precision ve Recall metriklerinde en yüksek değerlere sahiptir. Bu, modelin genel olarak en iyi performansı sergilediğini gösterir. Hem doğruluk hem de sınıflandırma yeteneği açısından en iyi sonuçları veren modeldir.

Bu değerlendirmeye göre:

Neural Network, tüm metriklerde en iyi performansı göstererek, veri setindeki sınıflandırma görevini en iyi şekilde yerine getiren modeldir. DVM, yüksek AUC ve dengeli performansıyla oldukça başarılı bir modeldir ancak Neural Network'ün biraz gerisindedir. KNN, diğer iki modele kıyasla daha düşük performans göstermektedir, ancak yine de kabul edilebilir sonuçlar üretmektedir.

Bu sonuçlar, Neural Network modelinin bu veri seti ve problem için en uygun model olduğunu göstermektedir.



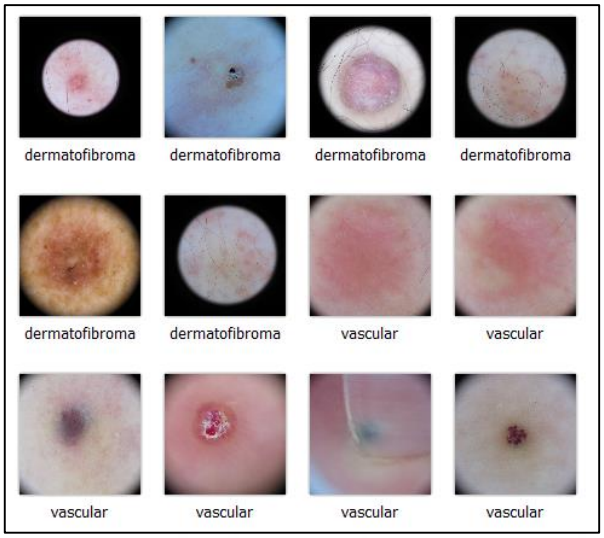

Şekil 3.8. Inception-v3 yöntemine göre karışıklık matrisi değerleri

Şekil 3.8., çeşitli makine öğrenimi modellerinin dermatoloji alanındaki tahmin performansını göstermektedir. Özellikle, Destek Vektör Makinesi'nin (DVM) dermatofibroma değerlerini tahmin ederken yüzde 89'luk bir skorla en yüksek doğruluk seviyesine ulaştığını göstermektedir. Ayrıca, sinir ağı yüzde 89,4 gibi

etkileyici bir doğruluk oranına ulaşarak vasküler değerler için olağanüstü bir tahmin doğruluğu sergilemiştir. Lojistik regresyon, nevus değerlerini tahmin ederken yüzde 98,4'lük olağanüstü bir doğruluk oranı elde ederek diğer modellerden daha iyi performans göstermiştir. Bu tahminler Inception-v3 modeli ile birlikte veri gömme teknikleri kullanılarak oluşturulmuştur. Daha ayrıntılı bir açıklama yapabilmek için, bu sonuçların dermatoloji alanındaki önemini anlamak önemlidir. Dermatofibroma, vasküler ve nevus üç farklı cilt durumu veya cilt lezyonu türüdür. Burada tartışılan modeller, görüntüler veya teşhis bilgileri gibi çeşitli girdi verilerine dayalı olarak bu koşullar hakkında tahminlerde bulunmak için kullanılmaktadır. Destek Vektör Makinesi (DVM), veri noktalarını farklı sınıflara ayırmadaki etkinliği ile bilinen bir makine öğrenimi algoritmasıdır. Bu durumda, belirli bir cilt rahatsızlığı olan dermatofibroma vakalarını tanımlama ve tahmin etmede yüksek oranda (%89) doğru sonuç vermiştir.

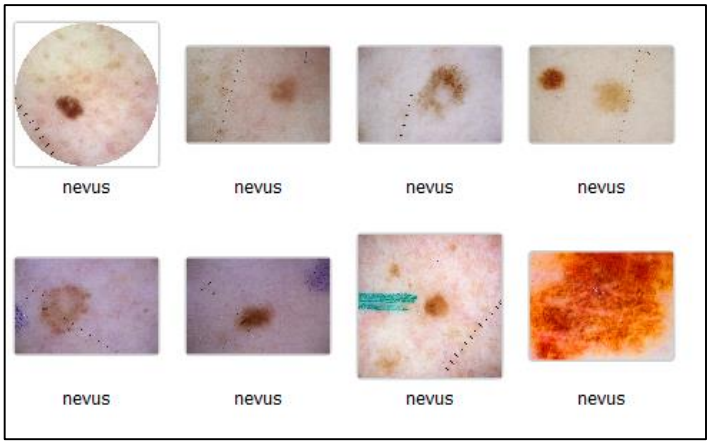
Derin öğrenmenin bir alt kümesi olan sinir ağları, verilerdeki örüntüleri tanımada mükemmeldir. Burada, sinir ağı vasküler durum vakalarını tahmin etmede %89,4 gibi etkileyici bir doğruluk oranına ulaşmıştır. Bu, bu özel cilt durumunu diğerlerinden ayırt etmede oldukça yetkin olduğunu göstermektedir. Lojistik regresyon, ikili sınıflandırma görevleri için kullanılan istatistiksel bir modeldir. Bu bağlamda, bir başka cilt rahatsızlığı olan nevus vakalarını tahmin etmede %98,4'lük olağanüstü bir doğruluk oranıyla diğer modelleri geride bırakmıştır. Bu yüksek doğruluk düzeyi, modelin nevusleri diğer cilt hastalıklarından ayırt etmede çok güvenilir olduğunu göstermektedir. Veri gömme için Inception-v3 modelinin kullanılması, cilt lezyonlarının görüntüleri gibi girdi verilerinin bu makine öğrenimi modelleriyle uyumlu olması için belirli bir işleme ve dönüşüme tabi tutulduğu anlamına gelir. Inception-v3, görüntü sınıflandırma görevleri için sıklıkla kullanılan popüler bir evrişimsel sinir ağı mimarisidir. Bu sonuçlar, Inception-v3 kullanılarak veri gömme ile donatıldığında, söz konusu makine öğrenimi modellerinin çeşitli cilt koşulları için etkileyici bir tahmin doğruluğu sergilediğini göstermektedir. DVM dermatofibromada mükemmel performans gösterirken, sinir ağı vasküler vakalar için oldukça iyi performans göstermiş ve lojistik regresyon nevus tahminlerinde olağanüstü doğruluk göstermiştir. Bu bilgiler, dermatoloji alanında cilt rahatsızlığı teşhislerinin doğruluğunu artırmak için değerlidir.

### 3.4.2. Cilt Kanseri Görüntü Kümeleme



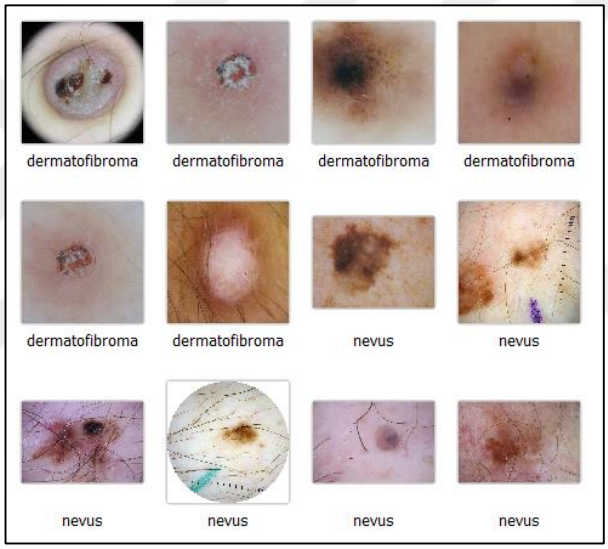

Şekil 3.9. K-Means C1 kümesi

Şekil 3.9.'da incelenen 73 görüntüden oluşan belirli bir veri seti sunulmuştur. Bu 73 görüntüden 15'inin vasküler lezyonlarla ilişkili olduğu, 58 görüntüden oluşan çoğunluğun ise dermatofibroma lezyonlarıyla bağlantılı olduğu ortaya çıkmaktadır. Bu görüntü koleksiyonu "C1 kümesi" olarak adlandırılmaktadır. Daha kapsamlı bir anlayış sağlamak için, bu bilgiler incelenen iki farklı cilt lezyonu kategorisi olduğunu göstermektedir: vasküler lezyonlar ve dermatofibroma lezyonları. Bu kategoriler arasında vasküler lezyonları temsil eden 15 görüntü bulunurken, 58 görüntüden oluşan daha büyük grup dermatofibroma lezyonlarına karşılık gelmektedir. "C1" olarak adlandırılan bu kümeleme, bu görüntüleri muhtemelen bir kümeleme veya sınıflandırma işlemi aracılığıyla tanımlanan ortak özelliklerine veya niteliklerine göre kategorize etmektedir.




Şekil 3.10. K-Means C2 kümesi

Şekil 3.10.'da, toplam 101 görüntü içeren belirli bir veri kümesini gözlemliyoruz. Bu veri kümesinde çarpıcı bir örüntü ortaya çıkmaktadır. Bu görüntülerin 97'si nevus lezyonlarıyla ilişkiliyken, kalan 4 görüntü dermatofibroma lezyonlarıyla bağlantılıdır. Bu özel görüntü koleksiyonu "C2 kümesi" olarak sınıflandırılmıştır. Daha ayrıntılı bir açıklama yapmak gerekirse, bu bilgi, bu veri setinde iki ana deri lezyonu kategorisini incelediğimizi göstermektedir: nevus ve dermatofibroma lezyonları. Görüntülerin çoğunluğunu oluşturan baskın kategori (101 görüntüden 97'si) nevus lezyonlarına karşılık gelmektedir. Buna karşılık, veri kümesinde dermatofibroma lezyon kategorisine ait yalnızca 4 görüntü bulunmaktadır. "C2 kümesi" terimi, muhtemelen bir kümeleme veya sınıflandırma işlemi yoluyla belirlenen ortak özellikler veya nitelikler temelinde bu görüntüleri bir arada gruplamak için kullanılmaktadır.



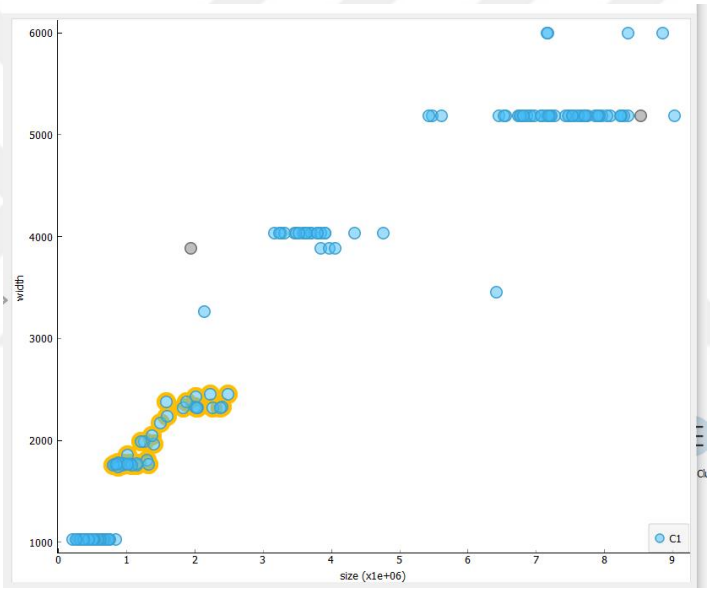

Şekil 3.11. K-Means C3 kümesi

Şekil 3.11., toplam 111 görüntüden oluşan bir veri kümesinin özetini sunmaktadır. Bu veri kümesinde, bu görüntülerin üç farklı lezyon görüntüsü arasında belirgin bir dağılımı vardır. Bunlar; dermatofibroma lezyon görüntülerini temsil eden 26 görüntü, nevus lezyon görüntülerini temsil eden görüntü ve vasküler lezyon görüntülerini temsil eden 56 görüntü olarak etiketlenmiştir. Bu görüntülerin kümelere ayrılması, görüntüleri ortak özelliklere veya modellere göre analiz eden ve gruplandıran bir algoritmanın sonuçlarına dayanmaktadır.

Bu kümeleme yaklaşımının başarısı, görüntüleri ilgili kümelere ne kadar doğru atadığının değerlendirilmesiyle ölçülebilir. Bu durumda, algoritma toplam 285 görüntüden 211'ini doğru bir şekilde kümelemiştir. Bu başarı oranı yüzde 74'e denk gelmektedir; bu da algoritmanın görüntüleri ortak özelliklerine göre gruplandırmada

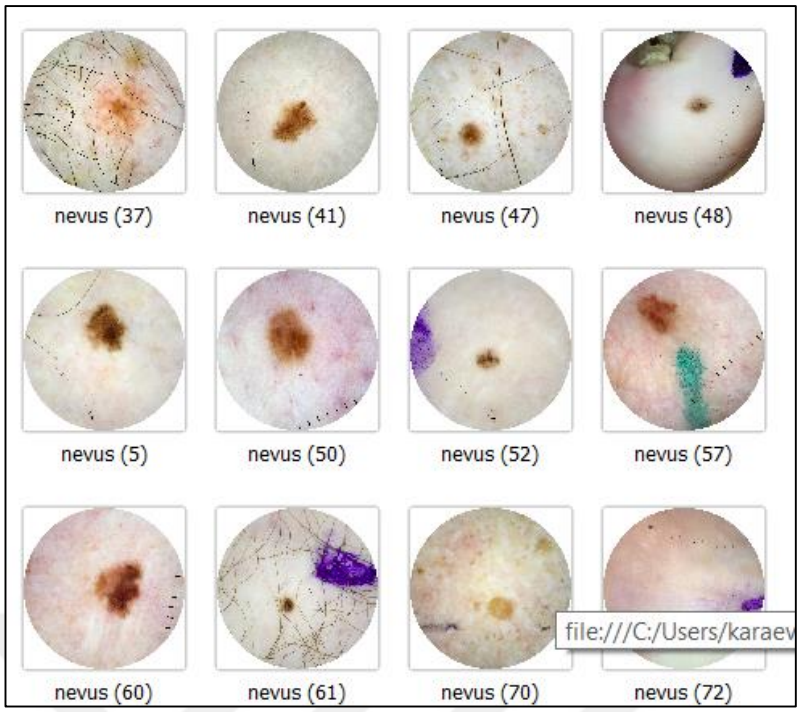
yüzde 74'lük bir doğruluk elde ettiğini göstermektedir. Özetle, Şekil 3.9. C1 kümesi, Şekil 3.10. C2 kümesi, Şekil 3.11. C3 kümesi hakkında bilgi vermektedir: dermatofibroma, nevus ve vasküler. Algoritma, görüntülerin çoğunu uygun kümelere doğru şekilde atayarak yüzde 74'lük bir başarı oranı elde etmiş ve böylece kümeleme işleminin etkinliğini göstermiştir.

Şekil 3.12., DBSCAN algoritmasının uygulanmasından kaynaklanan bir dağılım grafiğini göstermektedir. Grafiğin sol alt kısmında yer alan belirli bir kümeye odaklandığınızda, görüntülerin bir alt kümesini tanımlayabilirsiniz. Bu bağlamda, bu kümenin seçilmesi Şekil 3.13.'te gösterildiği gibi 43 nevus görüntüsünün tanımlanmasına yol açmaktadır. Daha basit bir ifadeyle, DBSCAN algoritması Şekil 3.12.'deki dağılım grafiğini oluşturmak için kullanılır.




Şekil 3.12. DBSCAN algoritması dağılım grafiği

Çizimin sol alt kısmında yer alan küme yakınlaştırıldığında, bir grup görüntü tam olarak belirlenebilir. Bu özel küme, nevus olarak kategorize edilen 43 adet görüntü içermektedir ve bunlar Şekil 3.13.'te daha ayrıntılı olarak görülür.




Şekil 3.13. DBSCAN Nevus Görüntüleri

# 4. TARTIŞMA VE SONUÇ

## 4.1. Uygulama 1: Araba Fiyat Tahmini

Bu çalışmada, araç fiyatlarının tahmin edilmesi amacıyla kullanılan çeşitli regresyon modelleri analiz edilmiştir. Orange programı ve Azure ML platformu kullanılarak gerçekleştirilen analizlerde, Linear Regresyon, Ridge Regresyon ve Lasso Regresyon algoritmalarının performansları karşılaştırılmıştır. Verilerin analizi sonucunda elde edilen bulgular aşağıda tartışılmıştır:

Model Performansları: Orange programı kullanılarak yapılan analizlerde, Linear Regresyon, Ridge Regresyon ve Lasso Regresyon algoritmalarının tümünde R-kare değerinin 0.967 olduğu görülmüştür. Bu, modellerin bağımsız değişkenlerin bağımlı değişken üzerindeki varyansın %96.7'sini açıkladığını göstermektedir. Azure ML platformunda ise Linear Regresyon algoritmasının R-kare değeri 0.96 olarak bulunmuştur. Bu sonuçlar, her iki platformda da regresyon modellerinin yüksek bir açıklayıcılık düzeyine sahip olduğunu göstermektedir.

Tablo 3.2'de sunulan yeni kayıtlar için yapılan tahminler, farklı algoritmaların benzer sonuçlar verdiğini göstermektedir. Örneğin, 30760 fiyatına sahip bir araç için Linear Regresyon, Lasso Regresyon ve Ridge Regresyon algoritmalarının tahminleri sırasıyla 31872.8, 31872.2 ve 31872.7 olarak bulunmuştur. Bu tutarlılık, modellerin genelleme yeteneğinin yüksek olduğunu ve yeni veriler üzerinde de güvenilir tahminler yapabileceğini göstermektedir.

Ridge ve Lasso Regresyon algoritmaları, aşırı uyum (overfitting) sorununu azaltmak için düzenleme (regularization) kullanır. Bu, özellikle yüksek boyutlu veri setlerinde modelin daha iyi genelleme yapmasına yardımcı olabilir. Çalışmamızda, her iki düzenleme yönteminin de Linear Regresyon ile benzer performans gösterdiği, ancak potansiyel aşırı uyum sorunlarını ele alarak daha dengeli modeller sunduğu gözlemlenmiştir.

Araç fiyat tahminine yönelik bu çalışmada, Linear Regresyon, Ridge Regresyon ve Lasso Regresyon algoritmalarının yüksek performans sergilediği ve bağımsız değişkenlerin fiyat üzerindeki etkisini büyük ölçüde açıklayabildiği görülmüştür. Orange programı ve Azure ML platformu kullanılarak yapılan analizler, modellerin tutarlılığını ve genelleme yeteneklerini ortaya koymuştur. Sonuç olarak, bu

modellerin, otomobil fiyatlarının tahmin edilmesinde etkili araçlar olduğu ve yeni veriler üzerinde de güvenilir tahminler yapılabileceği sonucuna varılmıştır.

Gelecek çalışmalar, daha geniş veri setlerinin kullanılması ve farklı özellik seçimi yöntemlerinin uygulanması ile modellerin performansının daha da iyileştirilmesine odaklanabilir. Ayrıca, derin öğrenme algoritmalarının kullanılması ve zaman serisi analizlerinin eklenmesi, araç fiyat tahminlerinin daha da kesinleştirilmesine katkı sağlayabilir.

### 4.2. Uygulama2: Müşteri Kayıp Analizi

WA_Fn UseC-Telco-Customer-Churn veri seti büyük veri analitiği yapılan bir platform olan Spark'ta işlenmiştir. Spark ile python programa dili kullanılmıştır. Spark Mlib kütüphanelerindeki hazır makine öğrenmesi kodları uygulamalara uyarlanmıştır. Aşağıdaki sonuçlar yeni müşterilerin firmayı terk edip etmeme tahminini yapan algoritmaların sonuçlarıdır.

Tablo 4.1. Algoritmaların Sonuçları

| **Algoritma** | **Başarı oranı** |
|---|---|
| GBM | 0,789 |
| Rastgele Orman | 0,784 |
| Lojistik Regresyon | 0,783 |
| Karar Ağacı | 0,782 |

Bu çalışmada, Spark platformu ve Python programlama dili kullanılarak Churn veri seti üzerinde büyük veri analitiği yapılmıştır. Spark Mlib kütüphanelerindeki hazır makine öğrenmesi kodları, yeni müşterilerin firma terkini tahmin eden algoritmalar için uyarlanmıştır. Elde edilen sonuçlar, farklı makine öğrenimi algoritmalarının başarılarının birbirine çok yakın olduğunu göstermektedir.

İncelenen algoritmaların başarılarının birbirine yakın olması, müşteri terkini öngörme konusunda çeşitli makine öğrenimi yöntemlerinin etkili olduğunu ve terk tahmini için güvenilir sonuçlar ürettiğini gösterir. Bu durum, işletmelerin müşteri terkini azaltmak için farklı algoritmalar arasında seçim yaparken esneklik sağlar ve çeşitli tekniklerin benzer derecede etkili olduğunu gösterir.

Öte yandan, başarı ölçütleri (accuracy, precision, recall, vb.) üzerinden detaylı bir karşılaştırma yapmak, algoritmalar arasında belirgin farklılıkları ortaya çıkarabilir. Özellikle, hangi başarı ölçütünün öncelikli olduğunu belirleyerek, terk tahmininde en

etkili yöntemi seçmek önemli olabilir. Örneğin, hassasiyet (precision) önemliyse, false positive oranını minimize eden bir algoritma tercih edilebilir.

Bu çalışmadan elde edilen sonuçlar, büyük veri analitiği ve makine öğrenimi tekniklerinin müşteri terkini öngörme konusunda güçlü bir araç olduğunu göstermektedir. Ancak, algoritmaların başarılarının benzer olması, her bir algoritmanın özelliklerini ve veri setine uygunluğunu dikkate alarak doğru seçimi yapmanın önemini vurgular.

Sonuç olarak, bu çalışma müşteri terkini öngörme konusunda farklı algoritmaların başarılarını karşılaştırarak işletmelere değerli bir rehberlik sunmaktadır. Makine öğrenimi yöntemlerinin etkili bir şekilde kullanılması, müşteri kaybını azaltmaya ve müşteri sadakatini artırmaya yönelik stratejilerin geliştirilmesine katkı sağlayabilir.

### 4.3. Uygulama3: Cilt Kanseri Görüntü Sınıflandırma ve Kümeleme

Cilt kanseri yaygın bir sağlık sorunudur ve cilt lezyonlarının kesin teşhisi erken teşhis ve başarılı tedavi için gereklidir. Bu çalışmada, teşhis amacıyla cilt kanseri görüntülerinin sınıflandırılması ve kümelenmesi için görüntü işleme tekniklerinin ve makine öğrenimi algoritmalarının kullanımı araştırılmıştır. Araştırmada 285 cilt kanseri görüntüsünden oluşan bir veri kümesi kullanılmıştır. Bu görüntüler üç farklı türe ayrılmıştır: 125 nevus lezyon (NV), 75 vasküler lezyon (VASC) ve 88 dermatofibroma lezyonu (DF). Sınıflandırma ve kümeleme algoritmaları uygulanmadan önce görüntüler ön işlemden geçirilmiştir. Özellikle Orange programı, görüntülerden ilgili özellikleri çıkarmak için Google'ın Inception v3 modelini kullanmıştır. Bu adım, verilerin kalitesini artırmak ve algoritmaların bunlarla etkili bir şekilde çalışabilmesini sağlamak için gereklidir. Önceden işlenmiş görüntü verileri üzerinde dört farklı sınıflandırma algoritması kullanılmıştır: K-En Yakın Komşular (KNN), Destek Vektör Makinesi (DVM), Yapay Sinir Ağı (YSA) ve Lojistik Regresyon. Bu algoritmalar görüntü sınıflandırma görevlerindeki etkinlikleri nedeniyle seçilmiştir.

Sonuçlar, bu sınıflandırma algoritmalarının cilt lezyonlarının teşhisinde etkinliğini göstermiştir. Algoritmaların her biri, gerçek dünyadaki tıbbi uygulamalardaki potansiyellerini gösteren başarılı sonuçlar sağlamıştır. Bununla birlikte, bu algoritmalar tek tek lezyonları sınıflandırmada iyi performans gösterirken,

kümeleme yaklaşımlarının ek içgörüler sunduğunu ve belirli bağlamlarda değerli olabileceğini belirtmek çok önemlidir. Benzer lezyonları bir araya getirmek için görüntü verilerine K-Means, Hiyerarşik Kümeleme ve DBSCAN gibi kümeleme algoritmaları uygulanmıştır. K-Means algoritması, diğer kümeleme yöntemlerine kıyasla nevus lezyonlarında daha yüksek bir başarı oranı elde ederek özellikle umut vaat etmiştir. Lezyonlar üç grupta sınıflandırılmıştır: C1 dermatofibroma, C2 nevus ve C3 vasküler. Daha da önemlisi, 285 görüntüden 211'i doğru bir şekilde kümelenerek %74 gibi etkileyici bir başarı oranı elde edilmiştir. Özellikle K-Means ile nevus verilerinin kümelenmesindeki yüksek başarı oranı, kümeleme algoritmalarının daha ayrıntılı bir analiz için benzer lezyonları etkili bir şekilde gruplandırabileceğini göstermektedir. Bu sadece teşhise yardımcı olmakla kalmaz, aynı zamanda çeşitli cilt lezyonlarının altında yatan modellerin ve özelliklerin daha iyi anlaşılmasına da potansiyel olarak katkıda bulunur.

Ayrıca, kümeleme algoritmalarının nevus özelliklerini sınıflandırmadaki başarısı, nevus görüntülerinin kalitesini gösterir. Lezyon tanılarını net bir şekilde ortaya koyan görüntüler daha doğru bir şekilde kümelenme eğilimindedir. Bu içgörü, cilt kanseri teşhisinde kümeleme algoritmalarının daha başarılı bir şekilde uygulanması için yüksek kaliteli görüntülerin önemini vurgulayarak gelecekteki veri toplama çabalarına rehberlik edebilir. Kümeleme algoritmalarının görüntüleri analiz etmede başarılı olabilmesi için görüntülerin uygun şekilde işlenmesi gerekir. Kümeleme algoritmalarında nevus verilerinin doğru kümelenme oranının yüksek olması, nevus görüntülerinin kalitesinin daha iyi olduğunu da düşündürmektedir. Buradan hareketle, lezyonlar için seçilen resimler, teşhis edilen lezyonları en net şekilde gösteren resimler olduğunda kümeleme algoritmalarının da başarılı sonuçlar vereceği tahmin edilmektedir. Sonuç olarak, bu araştırma cilt kanseri teşhisi alanında sınıflandırma ve kümeleme algoritmalarının potansiyelini vurgulamaktadır. Hem sınıflandırma hem de kümeleme algoritmaları cilt lezyonlarını doğru bir şekilde kategorize etme ve gruplandırma konusunda umut vaat etmektedir. Ayrıca sonuçlar, bu algoritmaların başarısı için görüntü kalitesinin ve uygun ön işlemenin önemini vurgulamaktadır. Bu araştırma, dermatolojide makine öğrenimi ve görüntü işleme tekniklerinin kullanımında genişleyen bilgi tabanına katkıda bulunmaktadır. Cilt kanseri teşhisini geliştirmek için büyük bir potansiyele sahiptir.

# KAYNAKLAR


Aftab, U. and Siddiqui, G.F. (2018) Big Data Augmentation with Data Warehouse: A Survey. 2018 IEEE International Conference on Big Data (Big Data), Seattle, 10-13 December 2018, 2785-2794.

Al-Nuaimy, W., Huang, Y., Nakhkash, M., Fang, M.T.C., Nguyen, V.T., & Eriksen, A. (2000). Automatic detection of buried utilities and solid objects with GPR using neural networks and pattern recognition. Journal of Applied Geophysics, 43(2-4), 157-165.

Alpaydın, E. (2010). Introduction to Machine Learning. MIT Press.

Aytekin, H.T. (2021). Makine Öğreniminin Araştırmacıların Veri Analizi Bağlamında Potansiyel Önemi. Ufuk Üniversitesi Sosyal Bilimler Enstitüsü Dergisi, 10(19), 85-106.

Bartosch-Härlid, A., Andersson, B., Aho, U., Nilsson, J., & Andersson, R. (2008). Artificial neural networks in pancreatic disease.

Bayes, T. (1763). An essay towards solving a problem in the doctrine of chances. Philosophical Transactions of the Royal Society of London, 53, 370-418.

Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.

Brown, M., & White, S. (2021). Predicting customer churn in the insurance industry using big data analytics. *Journal of Big Data Analytics in Insurance*, 10(3), 101-118.

Carvalho, G. (2017). The Internet Of Thingsand Big Data: Future Trends. Coimbra: ISEC –Instituto Superior de Engenharia de Coimbra.

Chen, P. (2016). Bıg Data Analytıcs In Statıc And Streamıng Provenance. Indiana: Doktora Tezi, Indiana University.

Cox, D. R. (1958). The regression analysis of binary sequences. Journal of the Royal Statistical Society: Series B (Methodological), 20(2), 215-242.

Çelik, S. (2018). Büyük veri ve istatistikteki uygulamaları. Doktora Tezi, Uludağ Üniversitesi / Sosyal Bilimler Enstitüsü / Ekonometri Anabilim Dalı / İstatistik Bilim Dalı, 191, Bursa.

Duda, R. O., Hart, P. E., & Stork, D. G. (2000). *Pattern Classification*. Wiley.

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and Its Applications. *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, 226-231.

Fix, E., & Hodges, J. L. (1951). Discriminatory analysis. Nonparametric discrimination: Consistency properties (Report No. 21-49-004). USAF School of Aviation Medicine, Randolph Field.

Freund, Y., & Schapire, R. E. (1997). "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting". Journal of Computer and System Sciences, 55(1), 119-139.

Galton Francis (1886); “Regression Towards Mediocrity in HereditaryStature”, Journal of Anthropological Institute of Great Britain andIreland, Vol. 15, 246–263

Ghahramani, M., Zhou, M. and Hon, C.T. (2019). Mobile Phone Data Analysis: A Spatial Exploration Toward Hotspot Detection, IEEE Transactions on Automatıon Science and Engineering,Vol. 16, No. 1, 351-362. January 2019.

Gini, C. (1912). Variabilità e mutabilità [Variability and Mutability]. Bologna: C. Cuppini.

Google Cloud. (2023). BigQuery Documentation. Retrieved from https://cloud.google.com/bigquery/docs.

Hinton, G. E., Rumelhart, D. E., & Williams, R. J. (1986). "Learning representations by back-propagating errors". Nature, 323(6088), 533-536.

Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. Technometrics, 12(1), 55-67. https://doi.org/10.1080/00401706.1970.10488634

Hui Liu, Y.X., Wu, X., Lv, X., Zhang, D., Zhong, G. (2019). Big Data Forecasting Model Of Indoor Positions For Mobile Robot Navigation Based On Apache Spark Platform, IEEE 4th International Conference on Cloud Computing and Big Data Analytics, 378-382, 2019.

Jhon, A., Smith, B., & Doe, C. (2018). Predicting customer churn in the telecommunication industry using machine learning techniques. *Journal of Telecommunications*, 12(3), 45-56.

Kara, M., Terzi, Y., & Cengiz, M. (2023). Comparative analysis of skin cancer image with classification and clustering algorithms. International Journal of Computer, 49(1), 229-244.

Khalifa S. (2017). Achıevıng Consumable Bıg Data Analytıcs By Dıstrıbutıng Data Mınıng Algorıthms. Doktoral Dissertation, Queen's University Kingston, 129, Ontario.

Khan, F. (2017). Büyük veri uygulamaları için onlıne non lineer olmayan modelleme. Doktora Tezi, İhsan Doğramacı Bilkent Üniversitesi / Mühendislik ve Fen Bilimleri Enstitüsü / Elektrik-Elektronik Mühendisliği Anabilim Dalı, 144, Ankara.

Kumar, K. A., Neupane, B., Malla, S., & Pandey, D. P. (2023). COVID-19 disease prediction using generative adversarial networks with convolutional neural network (GANs-CNN) model. In Proceedings of the International Conference on Recent Trends in Image Processing and Pattern Recognition (pp. 139-149). Springer Nature Switzerland.

Kustrin, T., & Beresford, R. (2000). Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research. *Journal of Pharmaceutical Sciences*, 89(10), 1371-1388.

Lee, J., & Kim, H. (2020). Predicting customer churn in the e-commerce industry using data mining techniques. *Journal of E-commerce Research,* 18(4), 250-265.

Legendre, A. M. (1805). Nouvelles méthodes pour la détermination des orbites des comètes [New Methods for Determining the Orbits of Comets]. Paris: Didot.

Li, P. L. (2016). Heterogeneous Archıtecture For Bıg Data Analytıcs. Doktoral Dissertation, Computer Engıneerıng Unıversıty Of Massachusetts Lowell, 134, Massachusetts.

McCulloch, W. S., & Pitts, W. (1943). "A logical calculus of the ideas immanent in nervous activity". Bulletin of Mathematical Biophysics, 5(4), 115-133.

Mitchell, T. M. (1997). Machine Learning. McGraw Hill.

Mlib, S. (2022, 5 22). Spark. Mlib: https://spark.apache.org/mllib/ adresinden alındı

Özcan, F. (2017). Bayesian Nonparametric Models On Big Data. Doktora Tezi, Unıversıty Of Calıfornıa, 129,Irvıne.

Patacchiola, M. (2020, July 31). An overview of Gaussian Mixture Models. Retrieved from https://mpatacchiola.github.io/blog/2020/07/31/gaussian-mixture-models.html [p. 1]

Paulsen, E. S. (2018). Bıgdataanalytıcs:Methods And Applıcatıons. Doktoral Dissertation, Unıversıty Of Wısconsın, 120, Madıson.

Quinlan, J. R. (1986). Induction of decision trees. Machine learning, 1(1), 81-106.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review, 65(6), 386-408.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). "Learning representations by back-propagating errors". Nature, 323(6088), 533-536.

Samuel, A. L. (1959). "Some Studies in Machine Learning Using the Game of Checkers". IBM Journal of Research and Development, 3(3), 210-229.

Sayad, S. (Erişim: Eylül 15, 2023). Logistic Regression. http://www.saedsayad.com/logistic_regression.htm

Sayad, S. (Erişim: Şubat 17, 2023). K-Nearest Neighbors (KNN). http://www.saedsayad.com/k_nearest_neighbors.htm

Schapire, R. E. (1999). "Theoretical views of boosting and applications". In Proceedings of the Tenth International Conference on Algorithmic Learning Theory (pp. 13-25).

Sengupta, S. (2016). Statıstıcal Analysıs Of Networks Wıth Communıty Structure And Bootstrap Methods For Bıg Data. Doktoral Dissertation, College Of The University Of Illinois At Urbana, 159, Champaign.

Shah, S. A. (2019). Afete dayanıklı akıllı şehırler için özgun bir çerceve: Büyük veri analitiği kullanımı. Doktora Tezi, İstanbul Teknik Üniversitesi / Bilişim Enstitüsü / Coğrafi Bilgi Teknolojileri Anabilim Dalı, 172, İstanbul.

Shannon, C. E. (1948). A mathematical theory of communication. Bell System Technical Journal, 27(3), 379-423.

Smith, A., & Johnson, B. (2019). Customer churn prediction in the banking industry using big data analytics. *International Journal of Data Science*, 15(2), 134-150.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, *58*(1), 267–288.

Towards Data Science. (2020, December 12). Gaussian Mixture Model Clearly Explained. Retrieved from https://towardsdatascience.com/gaussian-mixture-model-clearly-explained-115010f7d4cf [p. 2]

Turing, A. M. (1950). "Computing Machinery and Intelligence". Mind, 59(236), 433-460.

Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. New York: Springer.

Yue, Z., Zhang, W., & Zhang, Y. (2019). Sentetik açıklıklı radar görüntü tanıma için yeni bir yarı denetimli evrişimli sinir ağı yöntemi. Cognitive Computation, 11(6), 1077-1088. doi:10.1007/s12559-019-09659-9

Zebari, A. Y. H. (2023). *Ülkelere göre sosyal medyada büyük veri kullanarak Covid-19 gündeminin R programı ile analizi* (Doktora tezi). Hacettepe Üniversitesi Sosyal Bilimler Enstitüsü.

Zeng, Y. (2017). Scalable Sparse Machıne Learnıng Methods For Bıg Data. Doktoral Dissertation, The University Of Iowa, 115, Iowa City.

Zhao, T. (2017). Statistical Inference For Big Data. Doktoral Dissertation, Princeton University, 382, Princeton, USA:.

# ÖZ GEÇMİŞ

Muhammed KARA Samsun İmam-Hatip Lisesini bitirdikten sonra Gazi Üniversitesi Endüstriyel Sanatlar Eğitim Fakültesi, Bilgisayar Eğitimi Bölümünden 21.08.1997 tarihinde mezun oldu. 2011 yılında Hoca Ahmet Yesevi Uluslararası Türk-Kazak Üniversitesi/Mühendislik Fakültesi/Bilgisayar Mühendisliği (Tezsiz)Yüksek Lisans programına girdi. 9.6.2014 tarihinde bitirdi, 2017 yılında OMÜ LEE İstatistik ABD doktora programına başladı. 1997-2000 yılları arasında öğretmen, 2000 yılından sonra Öğretim Görevlisi olarak görev yapan M.K., orta derecede İngilizce bilmektedir (YÖKDİL:65). Temel ilgi alanları, Veri Bilimi, Makine Öğrenmesi. (15.06.2024).

**İletişim Bilgileri**

ORCID ID : *:https://orcid.org/0000-0002-5902-9065*

**Yayınlar:**

1. Kara, M., Terzi, Y., Cengiz, M. A. "Büyük Veri Analitiği İle Müşteri Davranış Analizi Customer Behavıor Analysıs Wıth Bıg Data Analytıcs", MAS 19th International European Conference on Mathematics, Engineering, Natural & Medical Sciences, Proceedings Book", 164-169, April 17-18, 2024, Mingachevir, Azerbaijan.
2. Kara, M., Terzi, Y., Cengiz, M. A. "Comparative Analysis of Skin Cancer Image with Classification and Clustering Algorithms", International Journal of Computer (IJC), 49(1), 229-244, 2023. https://ijcjournal.org/ index.php/InternationalJournalOfComputer/index.