VERİ ANALİZİNDE KULLANILABİLECEK EN İYİ KÜMELEME YÖNTEMİNİN BULUNMASI İÇİN BİR SİSTEM ÖNERİSİ

İhtiman Emre BİLGE
191402104

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı
Bilgisayar Mühendisliği Tezli Yüksek Lisans Programı
Danışman: Dr. Öğr. Üyesi Erdal GÜVENOĞLU

İstanbul
T.C. Maltepe Üniversitesi
Lisansüstü Eğitim Enstitüsü
Şubat, 2023

VERİ ANALİZİNDE KULLANILABİLECEK EN İYİ KÜMELEME YÖNTEMİNİN BULUNMASI İÇİN BİR SİSTEM ÖNERİSİ

İhtiman Emre BİLGE
191402104
Orcid No: 0000-0002-5729-5060

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı
Bilgisayar Mühendisliği Tezli Yüksek Lisans Programı
Danışman: Dr. Öğr. Üyesi Erdal GÜVENOĞLU

İstanbul
T.C. Maltepe Üniversitesi
Lisansüstü Eğitim Enstitüsü
Şubat, 2023

JÜRİ VE ENSTİTÜ ONAYI

Bu belge, Yükseköğretim Kurulu tarafından 19.01.2021 tarihli "Lisansüstü Tezlerin Elektronik Ortamda Toplanması, Düzenlenmesi ve Erişime Açılmasına İlişkin Yönerge" ile bildirilen 6698 Sayılı Kişisel Verilerin Korunması Kanunu kapsamında gizlenmiştir.

ETİK İLKE VE KURALLARA UYUM BEYANI

Bu belge, Yükseköğretim Kurulu tarafından 19.01.2021 tarihli "Lisansüstü Tezlerin Elektronik Ortamda Toplanması, Düzenlenmesi ve Erişime Açılmasına İlişkin Yönerge" ile bildirilen 6698 Sayılı Kişisel Verilerin Korunması Kanunu kapsamında gizlenmiştir.

TEŞEKKÜR

Tez çalışmamın her aşamasında çok değerli yardımlarını esirgemeyen, her konuda destek olan ve yol gösteren değerli danışman hocam Dr. Öğr. Üyesi Erdal GÜVENOĞLU'na, çalışmamın olgunlaşmasında ve tamamlanmasında değerli görüş ve katkılarından istifade ettiğim arkadaşlarıma sonsuz şükranlarımı sunuyorum iyi ki varsınız.

İhtiman Emre BİLGE

Şubat 2023

ÖZET

VERİ ANALİZİNDE KULLANILABİLECEK EN İYİ KÜMELEME YÖNTEMİNİN BULUNMASI İÇİN BİR SİSTEM ÖNERİSİ

İhtiman Emre Bilge
Yüksek Lisans Tezi
Bilgisayar Mühendisliği Anabilim Dalı
Bilgisayar Mühendisliği Tezli Yüksek Lisans Programı
Danışman: Dr. Öğr. Üyesi Erdal Güvenoğlu
Maltepe Üniversitesi Lisansüstü Eğitim Enstitüsü, 2023

Makine öğrenimi ile var olan veri ile 7 adet model üzerinden en yüksek skoru elde eden kümeleme yöntemi oluşturularak veriye göre en yüksek skor elde ettiğimiz algoritma mantığı geliştirilmiştir. Bu şekilde kısıtlı veri ile modeller ilişkilendirilip çıktıları analiz edilmiştir.

Birden fazla mantık içeren modeller seçilerek, sınıflandırma mantığı ve benzeri veriler baz alınarak test puanı ile tahmin puanı arasında en başarılı algoritmalar oluşturulmuştur. Veri analizi 7 model kullanılarak yapılır ve puanlar üzerinden fonksiyonlar yardımıyla verilerinize göre kullanım amacına göre en iyi puan sonuçlarını vermektedir.

Anahtar Sözcükler: Makine Öğrenimi, Veri Analizi, Kümeleme Yöntemleri, Sınıflandırma Yöntemleri, Makine Öğrenimi Algoritmaları

ABSTRACT

TO FIND THE BEST CLUSTERING METHOD THAT CAN BE USED IN DATA ANALYSIS SYSTEM RECOMMENDATION

İhtiman Emre Bilge
Master Thesis
Department of Computer Engineering
Computer Engineering Programme With Thesis
Thesis Advisor: Assist. Prof. Dr. Erdal Güvenoğlu
Maltepe University Graduate School, 2023

With the existing data with machine learning, the clustering method that achieves the highest score over 7 models was created, and the algorithm logic, in which we obtained the highest score according to the data, was developed. In this way, models were associated with limited data and their outputs were analyzed.

By choosing models containing more than one logic, the most successful algorithms were created between the test score and the prediction score on the basis of the classification logic and similar data. Data analysis is made using 7 models and it provides the best score results according to the purpose of use according to your data with the help of functions over the scores.

Keywords: Machine Learning, Data Analysis, Clustering Methods, Classification Methods, Machine Learning Algorithms

İÇİNDEKİLER

TABLOLAR LİSTESİ

ŞEKİLLER LİSTESİ

KISALTMALAR

AI	:	Artificial Intelligence	: Yapay Zeka
ML	:	Machine Learning	: Makine Öğrenimi
SVM	:	Support Vector Machine	: Destek Vektör Makinesi
RF	:	Random Forest	: Rastgele Orman
DC	:	Decisions Tree	: Karar Ağacı
KNN	:	K-Nearest Neighbors	: K-En Yakın Komşular
ADABOOST	:	Adaptive Boosting	: Uyarlanabilir Güçlendirme
LR	:	Logistic Regression	: Lojistik regresyon

1. GİRİŞ

1.1 Problem

Günümüzde karmaşık verilerde en büyük problem verilerin sınıflandırılmasıdır. Bu çalışmada, karmaşık verilerde hangi sınıflama yönteminin kullanılacağına yönelik yaşanan karmaşıklığın önlenmesi için bir yöntem önerilmektedir.

1.2 Amaç

Verinin en iyi sonuç vereceği sınıflama yönteminin belirlenmesi ve derin öğrenme ile sonuçlarının paylaşılması sağlanacaktır. Karmaşık verilerin anlamlı hale getirilerek en iyi sonucu verecek kümeleme yönteminin belirlenmesi sağlanacaktır ve o kümeleme yöntemleri derin öğrenme ile eğitilerek en iyi sonucu verecek yöntem belirlenecektir.

1.3 Önem

Çok sayıda kümeleme yönteminin olmasından dolayı makine öğretmesi adımında genellikle veriye uygun kümeleme yöntemi kullanmak yerine bilinen kümeleme yöntemleri kullanılmaktadır. Sonuçlarda da bu bilinen yöntemi iyileştirmek üzerine yoğunlaşılmaktadır. Bu çalışmada makine öğrenmesi ile verinin en iyi sonuç vereceği en iyi sınıflama yöntemleri belirlenecektir. Eldeki veriden en iyi eğitim datası elde edilecektir, dolayısıyla daha doğru sonuçlara ulaşılacaktır.

1.4 Varsayımlar

Yapay zekâ tarafından belirlenen en iyi sınıflandırma yönteminin gerçekten en verimli sonucu vereceği kabul edilmiştir. Kullanılan veri sayısının sınıflandırma yöntemi belirlemede yeterli sayıda olduğu varsayılmıştır.

1.5 Sınırlıklar

Kümeleme yöntemindeki çeşitliliğin fazla olduğundan ve çok büyük datalar kullanılacağından, oluşacak performans sorununun önüne geçilebilmesi için ve yaygın kullanım sağlamak amacıyla sınırlı sayıda kümeleme yöntemi ve data kullanılacaktır.

1.6 Tanımlar

Veri madenciliği, veri tabanı teknolojisi, istatistik, yapay zekâ, makine öğrenimi ve veri görselleştirme gibi birçok teknik alanı birbirine bağlayan bir mekanizmadır. Veri madenciliği astronomi, biyoloji, kimya, pazarlama, sigorta ve tıp gibi birçok alanda kullanılmaktadır.

Kümeleme, en önemli denetimsiz öğrenme sorunu olarak kabul edilmektedir, yani bu türden diğer tüm problemlerde olduğu gibi, etiketlenmemiş verilerden oluşan bir koleksiyonda bir yapı bulmakla ilgilenmektedir. Bu nedenle bir küme, aralarında "benzer" olan ve diğer kümelere ait nesnelere "benzer olmayan" nesnelerin bir koleksiyonudur (Madhulatha, 2012). Kümeleme modelinde, verilerin herhangi bir sınıfı bulunmamaktadır.

Veri kümeleme, denetlenmeyen verileri analiz etmek için kullanılan bir tekniktir ve verilerin şekil, boyut ve yoğunluk gibi çeşitli özelliklerine ve diğer özelliklere dayalı olarak nispeten kompakt gruplar halinde gruplanması olarak tanımlanabilmektedir (Jain, 2010). Veri kümeleme Aynı kategori içinde homojenliği maksimize etmeyi ve farklı kategoriler arasındaki heterojenliği maksimize etmeyi amaçlamaktadır(Hong ve ZHENG 2009). Veri kümeleme yöntemleri günümüzde yaygın olarak kullanılmakla beraber, biyoloji, sağlık, psikoloji, matematik, ekonomi, istatistik vb gibi birçok alanda örneklerine rastlanmaktadır. Bu tez kapsamında literatürde mevcut olan telekomünikasyon alanında veri kümeleme çalışmaları araştırılmış, mobil şirketlere ait kamusal datalar kullanılarak veri kümeleme yöntemleri örneklendirilmiştir. Veri kümeleme yöntemlerinin amacı telekomünikasyon şirketlerinin karlarını maksimize etmek için diğer müşteri özellikleri ile ürün ve hizmetler arasındaki doğal ilişkileri keşfetmektir.

2. LİTERATÜR ARAŞTIRMASI

Literatürdeki çalışmaların çoğunda iki kümeleme yöntemleri karşılaştırılmış, bu çalışmada ise altı farklı veri kümeleme yöntemi çalışılmış, veriye ait en iyi sonucu verecek yöntem belirlenmiştir. Aşağıda literatürdeki mevcut çalışmalar özetlenmiştir.

Hong ve arkadaşlarının 2009 yılındaki çalışmalarında, genetik temelli bir kümeleme yöntemi ile telekomünikasyon müşteri alt bölümü için algoritma (GA) sunulmuştur. Bu çalışmada öncelikle, telekomünikasyon müşterilerinin özellikleri (arama davranışı ve tüketim davranışı gibi) ayıklanmış, sonrasında telekomünikasyon müşterilerinin çok boyutlu öznitelik vektörleri arasındaki benzerlikler, iki boyutlu bir düzlemde örnekler arasındaki mesafe olarak hesaplanmış ve haritalanmıştır. Son olarak, mesafeler kademeli olarak benzerliklere yaklaşmak için ayarlanmıştır(Hong ve ZHENG 2009).

Velmurugan 2014 yılındaki çalışmasında, en çok temsil edilen, bölüm tabanlı kümeleme algoritmalarından ikisi, yani k-Ortalamalar ve Bulanık C-Ortalamalar analiz yöntemlerini kullanmıştır. Telekomünikasyon verileri bu analiz için kaynak veri olarak kullanılmıştır. Bağlantıya yönelik geniş bant verileri, seçilen algoritmaların performansını bulmakta kullanılmış, veriler işlendikten sonra sunucu konumları ve bağlantıları arasındaki mesafe (Öklid mesafesi) yeniden düzenlenmiştir. Her algoritmanın hesaplama karmaşıklığı (yürütme süresi) analiz edilerek sonuçlar karşılaştırılmış, elde edilen sonuçların daha doğru, anlaşılması kolay olduğunu ve hepsinden önemlisi, Fuzzy C-Ortalamalar algoritmasında verilerin işlenmesi için harcanan zamanın k-Ortalamalara göre önemli ölçüde yüksek olduğunu bulunmuştur (Velmurugan 2014).

Bahzad 2021 yılındaki çalışmasında karar ağaçlarına ayrıntılı bir yaklaşım sağlar. Kullanılan algoritmalar/yaklaşımlar, veri kümeleri ve elde edilen sonuçlar gibi kâğıt özellikleri kapsamlı bir şekilde değerlendirilir ve özetlenir. Ayrıca, analiz edilen tüm yaklaşımlar, yazarların temalarını göstermek ve en doğru sınıflandırıcıları belirlemek için tartışılmıştır. Sonuç olarak, farklı veri setlerinin kullanımları tartışılmış ve bulguları analiz edilmiştir (Charbuty ve Abdulazeez 2021).

Liyang bu çalışmasında, kümelenmiş mikrokalsifikasyonların (MC'ler) otomatik olarak sınıflandırılması için birkaç son teknoloji makine öğrenme yöntemini araştırmıştır. Sınıflandırıcı, radyologların mamogramlarda daha doğru meme kanseri teşhisi koymalarına yardımcı olmayı amaçlayan bilgisayar destekli bir teşhis (CADx) şemasının bir parçasıdır. Destek vektör makinesi (SVM), çekirdek Fisher diskriminant (KFD), uygunluk vektör makinesi (RVM) ve çoğu son zamanlarda istatistiksel öğrenme teorisinde geliştirilmiş olan komite makineleri yöntemlerinden faydalanmıştır. Denetimli bir öğrenme problemi olarak kötü ve iyi MC'lerin ayrımını formüle etmiştir ve sınıflandırma algoritmasını geliştirmek için bu öğrenme yöntemlerini uygulamıştır. Girdi olarak, bu yöntemler kümelenmiş MC'lerden otomatik olarak çıkarılan görüntü özelliklerini kullanmıştır (Wei ve Yang 2005).

Yanli Liu ve arkadaşları Rastgele Orman Makine Öğrenimini ve yeni bir kombinasyon algoritmasını çalışmalarında kullanmışlardır. Rastgele Orman, bir dizi ağaç yapısı sınıflandırıcısının birleşimidir. Random Forest'ın birçok iyi karakteri vardır. Rastgele Orman, sınıflandırma ve tahminde çılgınca kullanılmış ve regresyonda da kullanılmıştır. Geleneksel algoritmalarla karşılaştırıldığında Random Forest'ın birçok iyi avantajı vardır. Bu nedenle Rastgele Orman'ın uygulama kapsamı çok geniştir. Ağaç sınıflandırıcının bir kombinasyonu olarak RF, etkili bir sınıflandırma tahmin aracıdır. Aşağıdaki avantajlara sahiptir:

1) Rastgele ormanların doğruluğu Adaboost'tan daha az değildir, daha hızlı çalışır ve fazla uyum sağlamamaktadır.

2) OOB verileri, RF genelleme hatasını, korelasyonunu ve gücünü tahmin etmek için kullanılabilir, ayrıca bireysel değişkenlerin önemini de tahmin edebilir.

3) Torbalama ve bölünecek özelliklerin rastgele seçimi kombinasyonu, RF'nin gürültüyü daha iyi tolere etmesini sağlamaktadır.

4) RF, sürekli değişkenleri ve kategorik değişkenleri işleyebilir (Liu ve Wang 2012).

Dietterich 2000 yılında yaptığı sınıflandırıcılar kümesi ve daha sonra tahminlerinin (ağırlıklı) bir oyu alarak yeni veri noktalarını sınıflandırmışlardır. Orijinal topluluk yöntemi Bayes ortalamasıdır, ancak daha yeni algoritmalar hata düzeltici çıktı kodlamasını, Torbalama ve artırmayı içermektedir. Bu makale, bu yöntemleri gözden geçirmekte ve toplulukların neden genellikle herhangi bir sınıflandırıcıdan daha sonuç verebileceğini açıklamaktadır. Topluluk yöntemlerini karşılaştıran önceki bazı çalışmalar gözden geçirilmiştir ve Adaboost'un hızlı bir şekilde fazla uyum sağlamamasının nedenlerini ortaya çıkarmak için bazı yeni deneyler sunulmuştur. Topluluklar, daha az doğru olanları birleştirerek yüksek doğrulukta sınıflandırıcılar elde etmek için bir yöntem olarak iyi yapılandırılmıştır. Bu makale, topluluklar oluşturmak için yöntemler hakkında kısa bir inceleme sunmuş ve topluluk yöntemlerinin topluluk içindeki herhangi bir tek sınıflandırıcıdan daha iyi performans göstermesinin üç temel nedenini gözden geçirmiştir. Makale ayrıca AdaBoost'un bu kadar iyi performans göstermesinin nedenlerinden birini aydınlatmak için bazı deneysel sonuçlar da sağlamıştır. Bu yazıda tartışılmayan bir açık soru, AdaBoost ile temel öğrenme algoritmasının özellikleri arasındaki etkileşimle ilgilidir. AdaBoost ile birleştirilen öğrenme algoritmalarının çoğu, küresel bir karaktere sahip algoritmalardır (yani, nispeten düşük boyutlu bir karar sınırını öğrenen algoritmalar). Ayrıca yeni öğrenme algoritmaları elde etmek için AdaBoost aracılığıyla karlı bir şekilde birleştirilmiştir (Dietterich 2000).

Kotsiantis 2006 da yaptığı çalışmada denetimli sınıflandırmanın, Akıllı Sistemler tarafından en sık gerçekleştirilen görevlerden biri olduğunu söylemiştir. Yapay Zeka (Mantık tabanlı teknikler, Algılayıcı tabanlı teknikler) ve İstatistik (Bayes Ağları, Örnek tabanlı teknikler) temelli çok sayıda teknik geliştirilmiştir. Denetimli öğrenme, tahmin edici özellikler yönünden sınıf etiketlerinin dağılımının kısa bir prototipini oluşturmaktır. Ortaya çıkan sınıflandırıcı, tahmin edici özelliklerin değerlerinin bilindiği ancak sınıf etiketinin değerinin bilinmediği test örneklerine sınıf etiketleri atamak için kullanılmıştır. Bu makale, çeşitli sınıflandırma algoritmalarını ve sınıflandırma doğruluğunu iyileştirmeye yönelik son girişimleri- sınıflandırıcı topluluklarını açıklamaktadır. Bu makale, en iyi bilinen denetimli teknikleri göreceli ayrıntılı olarak açıklamaktadır. ML sınıflandırmasıyla uğraşırken anahtar soru, bir öğrenme algoritmasının diğerlerinden üstün olup olmadığı değil, belirli bir yöntemin belirli bir uygulama probleminde hangi koşullar altında diğerlerinden önemli ölçüde daha iyi performans gösterebileceğidir.

Meta-öğrenme, veri kümelerini algoritma performansına eşleyen işlevleri bulmaya çalışarak bu yönde ilerliyor (Kalousis ve Gama 2004).

Yalnızca mümkün olan en iyi sınıflandırma metodu ile ilgileniyorsak, iyi bir sınıflandırıcı topluluğu kadar iyi performans gösteren tek bir sınıflandırıcı bulmak zor veya imkânsız olabilir. Bariz avantajlara rağmen, topluluk yöntemlerinin en az üç zayıf yönü vardır. İlk zayıflık, eğitimden sonra tek bir sınıflandırıcı yerine tüm bileşen sınıflandırıcılarının depolanması gerekliliğinin doğrudan bir sonucu olarak artan depolamadır. Toplam depolama, her bileşen sınıflandırıcısının kendisinin boyutuna ve topluluğun boyutuna (topluluktaki sınıflandırıcıların sayısı) bağlıdır. İkinci zayıflık, artan hesaplamadır, çünkü bir girdi sorgusunu sınıflandırmak için (tek bir sınıflandırıcı yerine) tüm bileşen sınıflandırıcıları işlenmelidir. Son zayıflık, anlaşıla bilirliğin azalmasıdır. Karar vermede birden fazla sınıflandırıcının katılımıyla, uzman olmayan kullanıcıların bir karara yol açan temel akıl yürütme sürecini algılaması daha zordur. Topluluklardan anlamlı kurallar çıkarmaya yönelik ilk girişim (Wall ve ark., 2003)'de sunulmuştur. Tüm bu nedenlerden dolayı, mümkün olan en iyi sınıflandırma doğruluğuyla ilgileniyorsak, topluluk yöntemlerinin uygulanması önerilir. Anlaşıla bilirliği azaltmadan sınıflandırma doğruluğunu artırmaya çalışan diğer bir zaman alıcı girişim de sarmalayıcı öznitelik seçim prosedürüdür (Guyon ve Elissee, 2003). Teorik olarak, daha fazla özelliğe sahip olmak, daha fazla ayırt edici güçle sonuçlanmalıdır. Ancak, makine öğrenimi algoritmalarıyla ilgili pratik deneyim, durumun her zaman böyle olmadığını göstermiştir. Sarma yöntemleri, tahmin için çapraz doğrulama kullanarak özellik seçimini kullanılacak tümevarım algoritmasının etrafına sarmaktadır.

Kullanılan özellik alt kümesinden bir özellik eklemenin veya çıkarmanın faydaları. Veri tabanı topluluğu, gigabayt veri tabanlarıyla ilgilenir. Tabii ki, bir veri ambarındaki tüm verilerin aynı anda çıkarılması olası değildir. Mevcut öğrenme algoritmalarının çoğu, hesaplama açısından pahalıdır ve tüm verilerin ana bellekte bulunmasını gerektirir; bu, birçok gerçekçi problem ve veri tabanı için açıkça savunulamaz. Dağıtılmış makine öğrenimi, veri kümesini alt kümelere ayırmayı, bu alt kümelerden eşzamanlı olarak öğrenmeyi ve sonuçları birleştirmeyi içerir (Basak ve Kothari 2004). Dağıtılmış aracı sistemleri, makine öğrenme süreçlerinin bu paralel yürütülmesi için kullanılabilir (Kotsiantis ve Zaharakis 2006).

Sınıflandırma, veri örnekleri için grup üyeliğini tahmin etmek için kullanılan bir veri madenciliği (makine öğrenimi) tekniğidir. Sınıflandırma amacıyla kullanılabilecek birkaç sınıflandırma tekniği vardır. Bu yazıda, temel sınıflandırma tekniklerini sunuyoruz. Daha sonra Bayes ağları, karar ağacı tümevarımı, k-en yakın komşu sınıflandırıcısı ve Destek Vektör Makineleri (SVM) dahil olmak üzere bazı ana sınıflandırma yöntemlerini güçlü yönleri, zayıf yönleri, potansiyel uygulamaları ve mevcut çözümleriyle ilgili sorunları tartışacağız. Bu çalışmanın amacı, makine öğrenmesinde farklı sınıflandırma tekniklerinin kapsamlı bir incelemesini sağlamaktır. Bu çalışma, sınıflandırma yöntemlerinin temelini daha da güçlendirmek için hem akademi hem de makine öğrenimi alanında yeni gelenler için yardımcı olacaktır.

Aized Âmin Soofi ve Arshad Awan 2017 de yaptıkları bu çalışmada, makine öğreniminin çeşitli popüler sınıflandırma teknikleri, temel çalışma mekanizmaları, güçlü ve zayıf yönleri ile tartışılmıştır. Mevcut çözümlerle ilgili potansiyel uygulamalar ve sorunlar da vurgulanmıştır. Sınıflandırma yöntemleri, etkileşimleri modellemede tipik olarak güçlüdür. Tartışılan sınıflandırma teknikleri, sağlık, finans vb. gibi farklı veri setleri üzerinde uygulanabilmektedir Her tekniğin kendine has avantajları, dezavantajları ve uygulama sorunları olduğu için hangi tekniğin diğerinden üstün olduğunu bulmak zordur. Sınıflandırma tekniğinin seçimi, kullanıcı problem alanına bağlıdır. Bununla birlikte, sınıflandırma alanında çok fazla çalışma yapılmıştır, ancak Büyük Verinin sınıflandırılmasındaki sorunlar gibi yeni sınıflandırma sorunlarıyla uğraşmaktan kaynaklanan sınıflandırma sorunlarının üstesinden gelmek için hala araştırma topluluğunun resmi dikkatini gerektirmektedir (Soofi ve Awan 2017).

Osisanwo ve arkadaşları 2017 yılında çalışmada çeşitli Denetimli Makine Öğrenimi (ML) sınıflandırma tekniklerini açıklamış, çeşitli denetimli öğrenme algoritmalarını karşılaştırır ve ayrıca veri seti, örnek sayısı ve değişkenler (özellikler) temelinde en verimli sınıflandırma algoritmasını belirlemektedir. Yedi farklı makine öğrenme algoritmaları ele alınmıştır. Sonuçlar, en kesin ve doğrulukla algoritmanın SVM olduğunu göstermiştir. Naïve Bayes ve Random Forest sınıflandırma algoritmalarının buna göre SVM'den sonraki doğru olduğu bulunmuştur (Osisanwo ve Akinsola 2017).

2019 yılında yapılan bu çalışmada veri madenciliği, yeni bilgi üretmek için önceden var olan büyük bir veri tabanını inceleme uygulaması olarak tanımlanmıştır. Sağlam bir

teknolojinin, kuruluşların depolanan veri ambarlarındaki en önemli bilgilere odaklanmasına yardımcı olma konusunda büyük bir potansiyeli vardır. Veri madenciliği araçları ve teknikleri, işletmeyi daha proaktif ve daha iyi bilgiye dayalı kararlar alarak gelecekteki eğilimleri tahmin edecektir. Veri madenciliği teknikleri, geleneksel olarak çözülmesi çok zaman alan işle ilgili soruları yanıtlayabilir. Bu yazının amacı veri madenciliği tekniklerini tanıtmaktır. Odak, Kümeleme, Karar, ağaç Tahmini ve Sinir Ağlarını içeren teknikler üzerinde olacaktır. Bu makale, veri madenciliği tekniklerinin ayrıntılı bir temsilini göstermektedir. Büyük veri, büyük hacimli karmaşık veri kümeleriyle ilgili bir terimdir. Veri madenciliği, tarihsel verilerden yararlı kurallar veya ilginç modeller çıkarmayı içermektedir. Yüksek performanslı bilgi işlem paradigması, bu makalede listelenen bazı teknikleri kullanarak büyük veri sorununu çözmek için gereklidir. Örneğin Kümeleme, Karar, ağaç Tahmini ve Sinir Ağları. Bahsi geçen veri madenciliği teknikleri işletmeyi kapsamlı ve başarılı bir şekilde yönlendirmek için yol gösterici olmaktadır (Lee ve Siau 2001).

Patel 2016 yılında yaptığı bu çalışmada veri madenciliğinde en iyi kümeleme yöntemleri ile ilgili çalışmıştır. Veri madenciliğinde, Kümeleme en popüler, güçlü ve yaygın olarak kullanılan denetimsiz öğrenme tekniğidir. Bazı benzerliklere dayalı olarak benzer veri nesnelerini kümelere yerleştirmenin bir yoludur. Kümeleme algoritmaları, Hiyerarşik kümeleme algoritması, Yoğunluk tabanlı kümeleme algoritması, Bölümleme kümeleme algoritması, Grafik tabanlı algoritma, Izgara tabanlı algoritma, Model tabanlı kümeleme algoritması ve Kombinasyonel kümeleme algoritması olmak üzere yedi gruba ayrılabilmektedir. Bu kümeleme algoritmaları koşullara göre farklı sonuçlar vermektedir. Bazı kümeleme teknikleri büyük veri kümeleri için daha iyidir ve bazıları rastgele şekillere sahip kümeleri bulmak için iyi sonuçlar vermektedir. Bu makale, çeşitli veri madenciliği kümeleme algoritmalarını öğrenmek ve ilişkilendirmek için planlanmıştır. Araştırılan algoritmalar şunlardır: K-Means algoritması, K-Medoids, Distributed K-Means kümeleme algoritması, Hiyerarşik kümeleme algoritması, Grid tabanlı Algoritma ve Yoğunluk tabanlı kümeleme algoritması. Bu makale, tüm bu kümeleme algoritmalarını birçok faktöre göre karşılaştırmıştır. Bu kümeleme algoritmalarını karşılaştırdıktan sonra, en iyi sonucu elde etmek için farklı koşullarda hangi kümeleme algoritmalarının kullanılması gerektiğini açıklamıştır. Farklı kümeleme tekniklerinin sonuçlarını araştırdıktan sonra, bazı kümeleme tekniklerinin büyük miktarda veri seti için

kullanıldığını, ancak bazılarının yoğunlukta yüksek varyanslı veriler oluştuğunda iyi sonuç vermediği bulunmuştur. Yani kümeleme algoritmalarını bilmeden iyi sonuçlar elde edilememektedir. Çünkü her kümeleme algoritması her koşul için en iyi sonucu vermemektedir (Patel ve Thakral 2016).

Hailei Zou 2019 yılında yaptığı çalışmada kümeleme algoritmalarını ve kümeleme madenciliğine uygulanmasını ele almıştır. Kümeleme analizi, veri madenciliğindeki ana araştırma yönlerinden biridir. Şu anda, tüm alanların derinliklerine inmiş ve iyi bir ilerleme kaydetmiştir. Veri madenciliğinde kümeleme analizinin rolünü hedefleyen bir kümeleme analizi algoritması ve veri madenciliğinde uygulaması önermiştir. Literatür karşılaştırmalı analiz yöntemiyle, küme analizinin temel kavramları ayrıntılı olarak açıklanmakta ve küme analizinde klasik algoritmalar tartışılmıştır. Kümeleme K-means algoritmasının temel gerçekleştirme süreci incelenmiş ve örnek bir simülasyon gerçekleştirmiştir. Araştırma, bu algoritmanın güçlü bir evrenselliğe sahip olduğunu ve çoğu veri analiz sitesine uygulanabileceğini ve büyük miktarda verinin zamanında tespiti ve analizi için teorik bir temel sağladığını göstermiştir. Veri madenciliği teknolojisi kullanılarak, kümeleme analizi algoritmasının bilgi madenciliğindeki rolü ayrıntılı olarak incelenmiş ve kümeleme K-means algoritmasının işleyişini analiz etmek için bir örnek verilmiştir. Sonuçlar, K-means kümeleme algoritmasında, önce bölünecek bir başlangıç değeri belirlememiz ve ardından ilk bölümü etkili bir şekilde optimize etmek için algoritmayı kullanmamız gerektiğini göstermektedir. Veri madenciliğinde K-ortalamalar algoritmasını kümelemenin anahtar ve zor noktasının, kümeleme sonucunu büyük ölçüde etkileyecek olan ilk kümeleme merkezinin seçimi olduğu deneysel olarak bulunmuştur. Deneysel sonuçlar, kümeleme K-means algoritmasının yüksek doğruluk, güçlü anti-parazit ve evrenselliğe sahip olduğunu ve büyük gelişme beklentilerine sahip olduğunu göstermiştir (Sharifzadeh 2019).

Himani ve Sunil 2013 yılında karar ağacı algoritmaları üzerine yaptıkları araştırmada, Bilgisayar teknolojisi ve bilgisayar ağ teknolojisi geliştikçe, bilgi endüstrisindeki veri miktarı giderek artmaktadır. Bu büyük miktardaki veriyi analiz etmek ve ondan faydalı bilgiler çıkarmak gerekiyor. Büyük miktarda eksik, gürültülü, bulanık ve rastgele verilerden yararlı bilgileri çıkarma işlemine veri madenciliği denir. Karar ağacı sınıflandırma tekniği, en bilinir veri madenciliği modellerinden biridir. Karar ağacında

temel öğrenme stratejisi olarak böl ve fethet tekniği kullanılmaktadır. Karar ağacı, kök düğüm, dallar ve yaprak düğümleri içeren bir yapıdır. Her bir düğüm, bir öznitelik üzerindeki bir testi belirtir, her dal, bir testin sonucunu belirtir ve her yaprak düğüm, bir sınıf etiketine sahiptir. Ağaçtaki en üstteki düğüm, kök düğümdür. Bu makale, Karar ağacının çeşitli algoritmalarına (ID3, C4.5, CART), bunların özelliklerine, zorluklarına, avantajlarına ve dezavantajlarına odaklanmıştır. Bu makalede çeşitli karar ağacı algoritmalarını incelenmiştir. Bu yazıda verildiği gibi her algoritmanın kendi artıları ve eksileri vardır. Çeşitli karar ağacı algoritmalarının verimliliği, ağacı türetmek için harcanan zaman ve doğruluklarına göre analiz edilebilmektedir. Bu makale, öğrencilere ve araştırmacıya karar ağacı algoritmaları, araçları ve uygulamaları hakkında bazı temel temel bilgiler sağlamaktadır (Sharma ve Kumar 2016).

3. MATERYAL VE YÖNTEM

Makine öğrenmesi ile veriler analiz edilerek k-fold ya da cross validation gibi öğrenme yöntemleri kullanılarak verilerin en iyi sonuç vereceği kümeleme yöntemleri belirlenecektir.

3.1 Makine Öğrenimi

Makine öğrenimi terimi 1959 yılında Arthur Samuel tarafından icat edilmiştir. Kendisi yapay zekâ ve bilgisayar oyunlarında öncülük yapan isimlerden biridir. Makine öğrenmesi, veriler üzerinden tahmin yapılan aynı zamanda yapısal işlev olarak da öngörülebilen algoritmaların çalışmalarını ve inşalarını araştıran bir sistemdir.

Makine öğrenmesi, büyüyüp gelişen veri birimi alanının önemli bileşenlerinden biridir. Algoritmalar, tahminler veya sınıflandırmalar yapmak üzere istatistiksel veriler kullanılarak eğitilmektedir. Böylece temel iç görüler veri madenciliği projelerinde ortaya çıkartılmış olmaktadır. Makine öğrenmesi, odağına performansı geliştirilecek sistemleri oluşturma fikrini koymaktadır. Bunun için de matematiksel ve istatistiksel tüm yöntemleri kullanmaktadır.

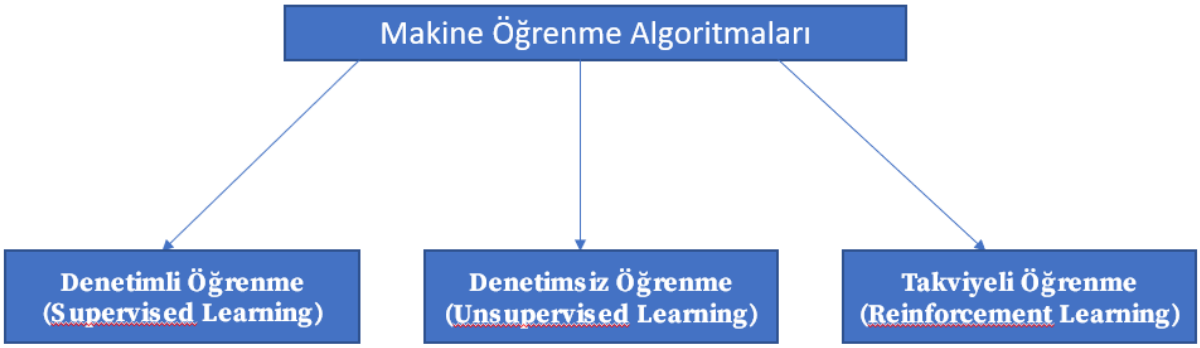
Günümüzde makine öğrenmesini birçok alanda kullanmaktayız. Bu alanlar, internette alışveriş yapma, spam filtreleme, sosyal medya kullanımı, ağ güvenliği tehdidini algılama, sahtekarlık tespiti, yüz tanıma, bankalarla iletişime geçme ve belge sınıflandırmadan oluşmaktadır (Janiesch ve Zschech 2021)

3.2 Makine Öğrenmesinin Çeşitli Tanım ve Anlamları

Bilgisayarların veri tabanları ya da algılayıcı verisi gibi veri türlerine dayalı ve öğrenime olanak sağlayan algoritmaların tasarım ve geliştirme süreçlerini konu alan bir bilim dalıdır. Bu araştırmalar bilgisayarlara çeşitli beceriler kazandırma üzerine odaklanmıştır. Bu beceriler, karmaşık örüntüleri algılama ve veriye dayalı akılcı kararlar verebilme yetisinden oluşmaktadır.

Regresyon problemleri ya da bir sınıflandırmanın makine tarafından çözüm metotlarını inceleyen bir bilgisayar bilimi dalı olarak ortaya çıkmıştır. Teori geliştirmeye başladıktan sonra biyoloji, psikoloji, matematik, istatistik gibi birçok dalın uzmanlarının bir araya gelerek bütüncül bir çalışma gerektiren çok disiplinli bir çalışma konusu haline gelmiştir.

Mevcut verilerden çıkarım yapabilmek için matematiksel ve istatistiksel verilerin kullanıldığı ve bu çıkarımlarla bilinmeyene dair çeşitli tahminlerde bulunan yöntem paradigmasıdır. Model tanıma ve hesaplamalı öğrenme teorisi çalışmaları yapay zekâ içinde yer almaktadır ve tüm bu çalışmalardan çıkan bilgisayar alt bilimidir.



Şekil 1. Makine Öğrenmesi Algoritmaları

Şekil 1'de görüldüğü üzere makine öğrenmesindeki algoritmaların genel olarak üç alt kolu vardır. Bunlar denetimli (denetlenen/gözetimli) makine öğrenmesi, Denetimsiz (Denetimsiz/Gözetimsiz) makine öğrenmesi ve Takviyeli (yarı denetimli) makine öğrenmesidir.

3.3 Denetimli Makine Öğrenimi

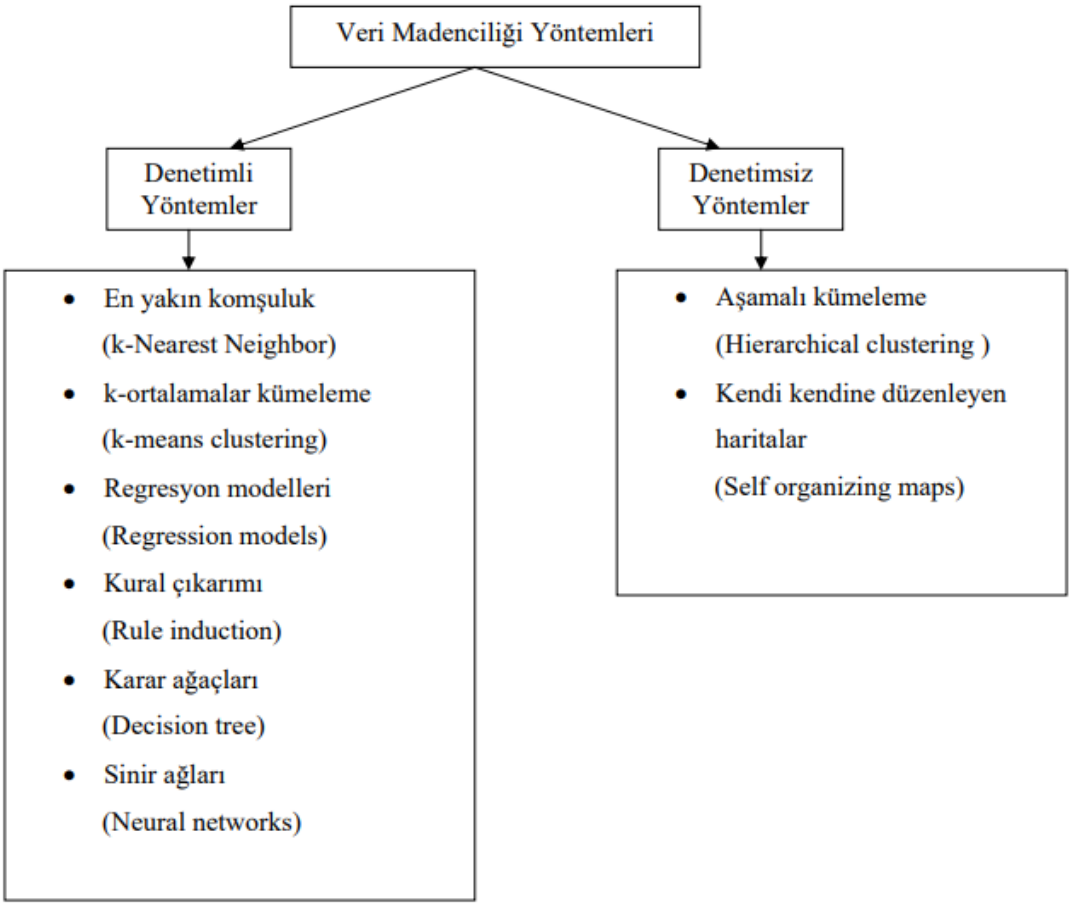
Denetimli öğrenme girdi değişkenlerine ve bir çıktı değişkenine sahip olunan ve girdiden çıktıya eşleme işlemini öğrenmek için kullanılan bir yerdir. Amaç; eşleme işlemini o kadar iyi tahmin etmektir ki, yeni girdi verileri olduğunda, bu veriler için çıktı değişkenini tahmin edebilmeliyiz. Denetimli öğrenme problemleri regresyon ve sınıflandırma problemleri olarak gruplandırılmaktadır.

Regresyon: Bir regresyon problemi, çıktı değişkeninin “tl” ya da “ağırlık” gibi sayısal, gerçek bir değer olduğu zamandır.

Sınıflandırma: Bir sınıflandırma problemi, çıktı değişkeninin “mavi” veya “sarı” veya “hastalık var” veya “hastalık yok” gibi kategori olması durumudur.

3.4 Denetimsiz Makine Öğrenimi

Denetimsiz öğrenme, yalnızca girdi verilerimizin olduğu ve karşılık gelen çıktı değişkenlerimizin olmadığı yerdir. Denetimsiz öğrenmenin amacı, veriler hakkında daha fazla bilgi edinmek için verilerdeki temel yapıyı veya dağılımı modellemektir. Denetimli öğrenmenin aksine doğru cevaplar yoktur ve öğretmen yoktur. Algoritmalar, verilerdeki ilginç yapıyı keşfetmek ve sunmak için kendi icatlarına bırakılmıştır. Denetimsiz öğrenme problemleri, kümeleme ve ilişkilendirme problemleri olarak gruplandırılmaktadır. Kümeleme, kümeleme sorunu, müşterilerin satın alma davranışına göre gruplandırılması gibi verilerdeki doğal gruplamaları keşfetmek istediğimiz yerdir. X ürününü satın alan kişilerin aynı zamanda Y ürününü satın alma eğiliminde olması gibi, verilerimizin büyük bölümlerini tanımlayan kuralları keşfetmek istediğimiz yer ise ilişkilendirmedir. Çok kullanılan veri madenciliği yöntemleri denetimli ve denetimsiz yöntemler olmak üzere Şekil 2’ deki gibi kategorize edilmektedir (Vatansever 2009) .



Şekil 2. Veri Madenciliği Yöntemleri (Vatansever 2009)

3.5 Takviyeli (Yarı Denetimli) Makine Öğrenimi

Büyük miktarda girdi verisine sahip olunan ve yalnızca bazı verilerin etkilendiği problemlere yarı denetimli öğrenme problemleri denilmektedir. Bu problemler hem denetimli hem de denetimsiz öğrenme arasında yer almaktadır.

Birçok makine öğrenimi bu alana girmektedir. Bunun nedeni, etki alanı uzmanlarına erişim gerektirebileceğinden, verileri etiketlemenin pahalı veya zaman alıcı olabilmesidir. Oysa etiketlenmemiş veriler ucuzdur ve toplaması, saklaması kolaydır. Verilerimizin büyük bölümlerini tanımlayan kuralları keşfetmek istediğimiz yer ise ilişkilendirmedir.

3.6 Veri Madenciliği Süreci

Veri madenciliği, veri analizi için yapay zekâ, istatistik, veri tabanı teknolojisi ve veri ambarından kapsamlı bir şekilde yararlanmaktadır. Çünkü veri madenciliği sadece kullanıma hazır verileri analiz etmekle ilgili değildir. Veri madenciliği sadece veri analizini değil, araştırılan problemle ilgili veri tabanlarının hazırlanmasını, ilgili veri tabanlarından veri sorgulanmasını, verilerin analize hazırlanmasını ve analiz sonucunda elde edilen bilgilerin bilgiye dönüştürülmesini içeren uzun bir süreçtir. Veri madenciliği süreci dört kısımda düşünülebilir: karar problemi belirleme, veri ön işleme, veri analizi ve sonuçların yorumlanmasıdır.

3.7 Karar Probleminin Belirlenmesi

Veri madenciliği çalışmasının gerçekleştirilebilmesi için öncelikle çalışmanın amacının netleştirilmesi gerekmektedir. Soruşturmanın amacı problem odaklı olmalı ve açık bir dille ifade edilmelidir. Elde edilen sonuçların başarısını nasıl ölçülmek istenildiğini belirtmek gerekmektedir. Hedeflere karar verdikten sonra veri hazırlama sürecine geçilmelidir. Doğru veriler seçilmeli ve analiz için uygun hale getirilmelidir. Yüzlerce, binlerce değişkenle araştırma yapmak yerine, ilgili değişkenleri kullanmak zaman kazandırmaktadır.

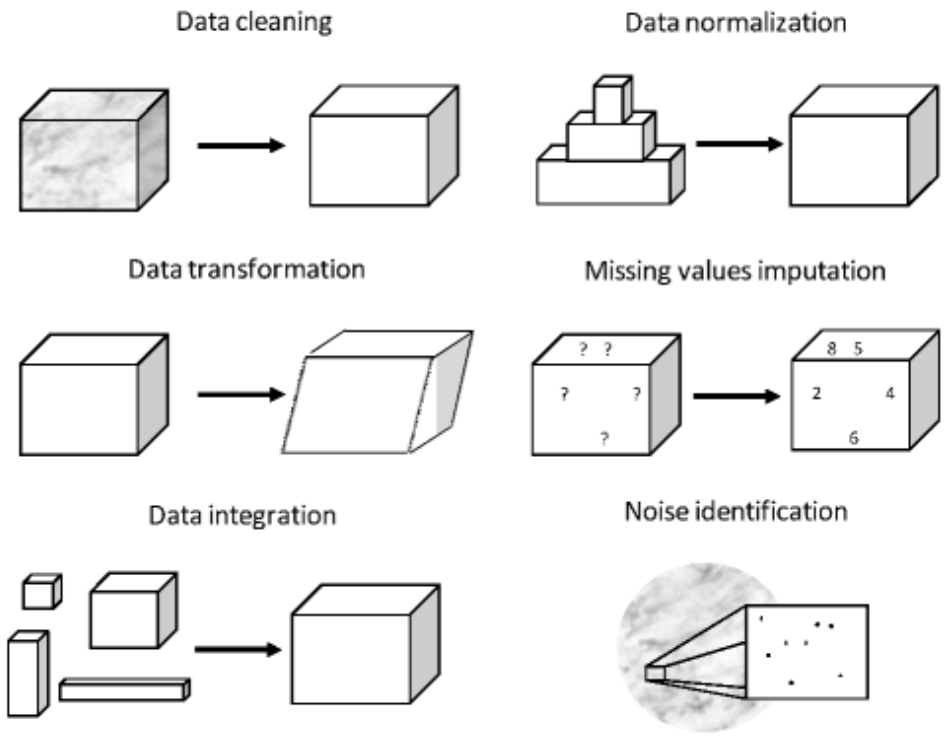
3.8 Veri Ön İşleme

Veri kalitesi, veri madenciliğinin anahtarıdır. Veri madenciliğinin güvenilirliğini artırmak için bir veri ön işleme aşaması gereklidir. Aksi takdirde, yanlış giriş verileri yanlış sonuçlara yol açabilir. Veri ön işleme, aşağıdaki nedenlerle verilere uygulanır (Şirin 2017):

1- Verilere herhangi bir analiz yapılmasına engel olan veri problemlerini çözmek,

2- Verinin doğasını anlamak ve anlamlı veri analizi gerçekleştirmek,

3- Belirli bir veri kümesinden daha anlamlı bilgiler çıkarmak.



Şekil 3. Veri Ön İşleme örnekleri (Şirin 2017)

3.9 Veri Analizi

Modele hangi değişkenlerin dahil edileceğine karar verildikten sonra uygun veri hazırlama süreci, verileri analize uygun hale getirir. Veri madenciliği modellemesi daha sonra veri madenciliği algoritmaları arasından karar problemini çözmek için bir yöntem seçilerek gerçekleştirilir. Veri madenciliği teknikleri birbirinden bağımsız veya birlikte kullanılabilir. Yani bir veri madenciliği tekniğinin çıktısı başka bir veri madenciliği tekniğinin girdisi olabilmektedir. Örneğin küme analizi veya regresyon analizi gerçekleştirmek için temel bileşen analizi sonucunda elde edilen skor değerleri girilebilmektedir.

Kümeleme analizi ile üretilen homojen fakat heterojen gruplar diğer veri madenciliği tekniklerinde de kullanılabilir. Genel olarak veri madenciliği tekniklerinin birlikte kullanılması daha etkili sonuçlar verecektir. Mümkün olduğu kadar çok model oluşturup

test ederek karar verme probleminize en uygun model bulunabilmektedir. Bu nedenle, veri hazırlama ve model oluşturma aşamaları, siz optimal kabul edilen bir modele ulaşana kadar tekrarlanan bir süreçtir. İteratif süreçte başarısız olan örnekler, başarılarının nasıl artırılabileceğini görmek için incelenir. Örneğin programa verilen bilgiler standart forma yeni alanlar eklenerek genişletilebilir. Veya bilgi başka bir şekilde kodlanmış olabilir. Ya da amacınızı farklı tanımlayabilirsiniz (Vatansever, 2008).

3.10 Sonuçların Yorumlanması

Verilere bir veya daha fazla veri madenciliği algoritması uygulandıktan sonra, sonuçlar sıralanır ve ilgili konumlarda görüntülenir. Bu sonuçlar olabildiğince görselleştirilmiştir ve son kullanıcılar için uygundur.

3.11 Sınıflandırma

Fikirleri ve nesneleri önceden tanımlanmış kategoriler veya alt popülasyonlar halinde tanıma, anlama ve gruplandırma sürecidir. Önceden sınıflandırılmış bir eğitim veri kümesi kullanan bir makine öğrenimi programı, gelecekteki veri kümelerini sınıflandırmak için farklı bir algoritma kullanılmaktadır.

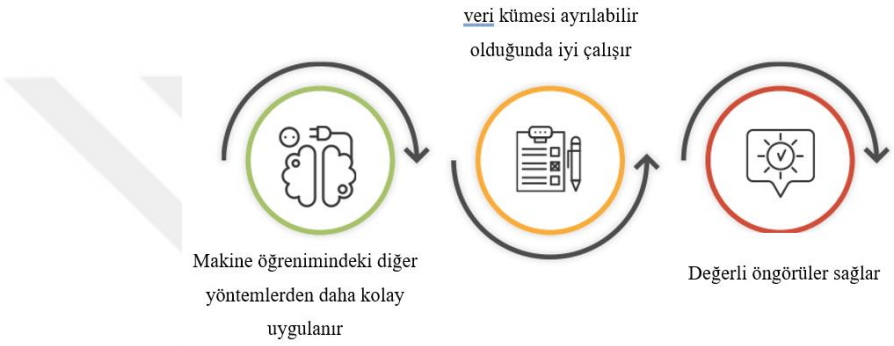
Makine öğrenimi sınıflandırma algoritmaları, sonraki verilerin belirtilen kategorilerden birine girme olasılığını tahmin etmek için girdi eğitim verilerini kullanmaktadır.

Sınıflandırma, gelecekteki veri setlerinde aynı kalıpları (benzer kelimeler veya anlamlar, sayı dizileri, vb.) bulmak için eğitim verilerine bir sınıflandırma algoritması uygulayan bir kalıp tanıma türüdür(Wolff 2020).

3.12 Logistic Regresyon

Lojistik regresyon, doğrusal bir model oluşturan istatistiksel bir tekniktir. Genellikle ikili bir durumu tahmin etmek için kullanılmaktadır. Bu analitik yaklaşımında, bağımlı değişken sonlu veya kategoriktir. A veya B (ikili regresyon) ya da bir dizi sonlu A, B, C, D seçeneği (çok terimli regresyon) istatistiksel yazılımda, bir lojistik regresyon denklemi kullanarak olasılıkları tahmin ederek bağımlı değişken ile bir veya daha fazla bağımsız değişken arasındaki ilişkiyi anlamak için kullanılmaktadır.

Bu tür analiz, bir olayın olma olasılığını veya yapılan bir seçimin olasılığını tahmin etmemize yardımcı olabilir. Örneğin, bir ziyaretçinin web sitenizde yapılan bir teklifi seçip seçmediğini (bağımlı değişken) bilmek isteyebiliriz. Analiz, ziyaretçilerin geldikleri siteler, sitemize tekrar ziyaret, sitemizdeki davranışlar (bağımsız değişkenler) gibi bilinen özelliklere bakabilir. Lojistik regresyon modelleri, hangi ziyaretçilerin teklifi kabul edip etmeyeceğine dair bir olasılık belirlenmesine yardımcı olmaktadır. Sonuç olarak, teklifimizi tanıtma konusunda daha iyi kararlar verebilir ve teklifin kendisi hakkında kararlar verilebilmektedir. Aşağıdaki şekilde LR'nun temel avantajları gösterilmiştir(Science 2018).



Şekil 4. Lojistik Regresyonun Temel Avantajları (Science 2018)

3.13 Decisions Tree

Karar ağacı, birçok kayıt içeren bir veri setini bazı kurallar uygulayarak daha küçük setlere bölmek için kullanılan bir yapıdır. Karar ağaçları, kazanan yöntemi izleyerek verileri kökten yaprağa yinelemeli olarak bölümleyerek oluşturulmaktadır. Başlangıçta, tüm veriler ağacın kökünde toplanır. Değişkenlerin seçimi, bilgi kazancının değeri tarafından belirlenir. Yinelemeli algoritmanın döngüden çıkabilmesi için o düğümdeki tüm elemanların aynı sınıfa ait olması gerekmektedir. Kalan değerler sadece bir sınıfa aitse veya sınıflandırılabilecek değer yoksa döngü algoritması sona erer ve bir karar ağacı oluşturulur. Ortaya çıkan sınıftaki her öğe, aynı sınıfın diğer öğelerine benzer özellikler gösterir. Ağaç yapıları, heterojen veri kümelerini daha küçük homojen yapılara dönüştürmek için kurallar tanımlamaktadır.

İstatistiksel bir yöntem olarak karar ağacı yönteminin ilk adımı bir ağaç yapısı oluşturmak ve veri setindeki verileri bu ağaca işlemektir. Bir karar ağacının yapısı bir kök düğüm,

düğümler ve dallardan oluşmaktadır. Kural oluştururken sorular sorulur ve verilen cevaplara göre işlem yapılmaktadır. Cevaplar birleştirilerek yeni kurallar oluşturulur. Karar verildikten sonra ağaç yapısının ilk aşamadaki kök düğümü oluşturulur. Kök düğüm, bağımlı değişkeni temsil etmektedir. Kök düğümden başlayarak ağaç yapısında aşağı doğru yürüdüğünüzde, veri kümesi parçalara ve dallara bölünür ve yeni düğümler oluşturulur. Her düğüm bir soruyu temsil eder ve o soruya verilen cevaplar kadar dala ayrılmaktadır. Her dalda "eğer-o zaman" kuralları oluşturulur ve seçilen karara göre bir sonraki düğüme geçilir. Yeni bir problem ortaya çıkana kadar ağaç yapısı bu şekilde ele alınır. Son olarak sınıfı temsil eden son düğüme ulaşılmaktadır (Çelik, 2009).

Karar ağaçlarını sınıflandırma için kullanmanın avantajları;

• Kolayca karar ağaçları oluşturulur

• Küçük ağaçların anlaşılması kolaydır

• Kolay anlaşılır kurallar oluşturulabilir

• Hem sürekli hem de ayrı öznitelik değerleri için kullanılabilir.

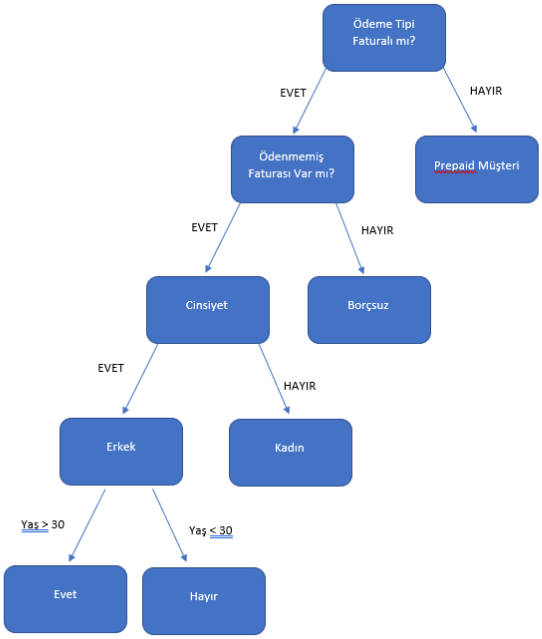
Sınıflandırma için karar ağaçlarını kullanmanın dezavantajları:

• Sürekli öznitelik değerlerini tahmin etmede pek iyi değildir.

• Sınıf sayısı çok ve öğrenme kümesi örneklerinin sayısı az olduğunda model oluşturma çok başarılı değildir.

• Zaman ve yerin karmaşıklığı, öğrenme seti örneklerinin sayısına, özelliklerin sayısına ve ortaya çıkan ağacın yapısına bağlıdır.

• Hem ağaç oluşturma karmaşıklığı hem de ağaç budama karmaşıklığı yüksektir.



Şekil 5. Decisions Tree Kullanım Örneği

3.14 Adaboost

AdaBoost, başlangıçta ikili sınıflandırıcıların verimliliğini artırmak için oluşturulmuş bir topluluk öğrenme yöntemidir ("meta-öğrenme" olarak da bilinir). AdaBoost, zayıf sınıflandırıcıların hatalarından ders çıkarmak ve bunları güçlü sınıflandırıcılara dönüştürmek için yinelemeli bir yaklaşım kullanmaktadır.

Ada-boost veya Adaptive Boosting, 1996 yılında Yoav Freund ve Robert Schapire tarafından önerilen topluluk güçlendirme sınıflandırıcılarından biridir. Sınıflandırıcı doğruluğunu artırmak için birden çok sınıflandırıcı birleştirilmektedir. AdaBoost yinelemeli bir topluluk yöntemidir. AdaBoost sınıflandırıcı, birden çok düşük performanslı sınıflandırıcıyı birleştirerek güçlü bir sınıflandırıcı oluşturur ve size güçlü, yüksek doğrulukla bir sınıflandırıcı verebilir. Adaboost'un arkasındaki temel kavram, anormal gözlemlerin doğru tahminlerini sağlamak için sınıflandırıcı ağırlıklarını ayarlamak ve her yinelemede veri örneklerini eğitmektir. Herhangi bir makine öğrenimi

algoritması, eğitim seti ağırlıklarını kabul ederse temel sınıflandırıcı olarak kullanılabilir. Adaboost iki koşulu karşılamalıdır:

Sınıflandırıcı, farklı ağırlıklı eğitim örnekleri ile etkileşimli olarak eğitilir. Her yineleme, eğitim hatasını en aza indirerek bu örneklere mükemmel şekilde uymaya çalışır.

Adaboost Algoritmasının Çalışma Prensibi;

1. Adaboost başlangıçta rastgele bir eğitim alt kümesi seçer.

2. Son eğitimin doğru tahminine göre eğitim seti seçilir ve AdaBoost makine öğrenimi modelini yinelemeli olarak eğitilmektedir.

3. Yanlış sınıflandırılmış gözlemlere daha yüksek ağırlıklar atar, böylece bir sonraki iterasyonda daha yüksek sınıflandırılma olasılıkları elde ederler.

4. Ayrıca, sınıflandırıcının doğruluğuna dayalı olarak her yinelemede eğitimli sınıflandırıcılara ağırlıklar atar. Daha doğru sınıflandırıcılar daha yüksek ağırlıklar alacaktır.

5. Tüm eğitim verileri eşleşene veya belirtilen maksimum tahmin edici sayısına ulaşılana kadar bu işlemi tekrarlanmaktadır.

6. Sınıflandırmak için oluşturduğunuz tüm öğrenme algoritmalarına "oy verilmektedir.".

3.15 Random Forest

Rastgele bir orman sınıflandırıcısı, bir topluluk olarak çalışan birden çok karar ağacından oluşur. En yüksek oyu alan model tahminde bulunur. Uyum üzerinde daha iyi doğruluk ve kontrol için bu karar ağaçlarının ortalamasını alır.

Rastgele Orman (RF) algoritması, Breiman tarafından 2001 yılında karar ağaçlarının bir kombinasyonu olarak önerildi. RF, en iyi şekilde "*her ağacın bağımsız olarak örneklendiği ve ormandaki tüm ağaçlar için aynı dağılıma sahip rastgele bir vektörün değerine bağlı olduğu ağaç belirleyicilerinin bir kombinasyonu*"(Breiman, 2001). Olarak tanımlanan bir topluluk makine öğrenimi algoritmasıdır.

RO, regresyon ve sınıflandırma problemlerinde varyansı ve hatayı azaltarak hedef veya bağımlı değişkeni daha iyi tahmin etmemizi sağlamaktadır. Bunu bir önyükleme toplama veya torbalama yöntemi kullanarak yapmaktadır. Bu yöntem, veri kümesini alt örnek kümelerine veya alt örnek ağaç kümelerine bölmek için değiştirmeli basit rasgele örneklemeyi kullanır. Bütün bunlar, yerleşik modellerdeki hataları azaltmayı amaçlamaktadır.

Rastgele Orman Algoritması, sınıflandırma ve regresyon problemlerini çözmek için kullanılan makine öğreniminin denetimli öğrenme bölümünde yer alan yüksek tahmin oranlı bir algoritmadır.

RO'da torbalama yöntemini tercih etmemizin iki önemli nedeni vardır. İlk olarak, torbalama işlemi, doğruluğu artıran rastgele özellikler kullanır. İkincisi, genelleştirilmiş hatanın (of-of-bag (OOB)) hesaplanmasıdır (Breiman, 2001). Rastgele öznitelik seçiminde, önce gerçek veri kümesi kaydırılarak yeni bir eğitim veri kümesi oluşturulur. Daha sonra rastgele özellik seçimi kullanılarak yeni eğitim setinden bir ağaç oluşturulur. Olgun ağaçlar budanmaz ((Archer ve Kimes 2008) Breiman, 2001)). Pal (2005), budama yöntemi seçiminin ve özellik seçim kriterlerinin olmamasının ağaç tabanlı sınıflandırıcıların performansını etkilediğini belirtmektedir. Budama olmaması, RO'ya diğer karar ağacı tekniklerine göre bir avantaj sağlamaktadır.

3.16 Xgboost

XGBoost, verimli, esnek ve taşınabilir olacak şekilde tasarlanmış, optimize edilmiş bir dağıtılmış gradyan artırma kitaplığıdır. Gradient Boosting çerçevesi altında makine öğrenimi algoritmalarını uygular. XGBoost, birçok veri bilimi sorununu hızlı ve doğru bir şekilde çözebilen bir paralel ağaç güçlendirme (GBDT, GBM olarak da bilinir) sağlamaktadır. Aynı kod, büyük dağıtılmış ortamlarda (Kubernetes, Hadoop, SGE, MPI, Dask) çalışır ve milyarlarca örnek üzerinde sorunları çözebilir.

XGBoost'ta, birden çok çekirdek kullanılarak bireysel ağaçlar oluşturulur ve veriler, arama sürelerini en aza indirir. Bu, modellerin eğitim süresini azaltır ve bu da performansı artırmaktadır.

XGBoost ayrıca özünde gradyan artırma özelliğine sahiptir. Ancak, basit gradyan artırma arasındaki fark algoritması ve XGBoost algoritması, gradyan artırmadan farklı olarak, zayıf ekleme işlemidir. Öğrenenler birbiri ardına gerçekleşmez; doğru kullanımının olduğu çok iş parçacıklı bir yaklaşım benimsemektedir. Makinenin CPU çekirdeği kullanılır, bu da daha yüksek hız ve performans sağlar. Bunun dışında, eksiklerin otomatik olarak işlenmesini de içeren seyrek farkında bir uygulama vardır. Veri değerleri, daha sonra ağaç yapımının paralelleşmesini desteklemek için yapıyı bloke etmektedir.

Chen ve Guestrin tarafından geliştirilen XGBoost sınıflandırıcıları, sınıflandırma ve regresyon problemleri için uçtan uca ağaç sistemlerinde ölçeklenebilir makine öğrenimi teknikleridir. XGBoost bir CART topluluğu kullanır, her düğümde "$KKEE\ ii\ \backslash ii \in 1\ \ldots\ .KK$" düğümü bulunur. Son tahmin, her ağacın tahmin puanlarının toplamıdır. Bu yöntemin denklemi Denklem 3.1'de verilmiştir.

$$\sum_{k=0}^{n} \binom{n}{k} x^k a^{n-k} \tag{3.1}$$

XGBoost modelinin hiperparametreleri, ızgara arama optimizasyonu algoritması tarafından belirlenen maksimum derinlik, bölüm sayısı ve öğrenme oranıdır (Wolff 2020)

3.17 Gaussian Naive Bayes

Naive Bayes sınıflandırıcısı, beyaz kutu yapısına sahip istatistiksel bir sınıflandırıcıdır; bu, ML sürecinin şeffaf olduğu ve nasıl davrandıklarının net bir şekilde anlaşıldığı anlamına gelmektedir (Bergin, 2006). Adını Thomas Bayes'ten alan Bayes sınıflandırma yöntemi, Bayes Teorisine dayanmaktadır. Bu yaklaşımda, her örneğin sonsal olasılığı, kendi özel sınıfları dikkate alınarak hesaplanmaktadır. Bu sınıflandırıcı, öznitelik vektörü tarafından tanımlanan belirli bir örneğe en olası sınıfı atar. Bu algoritmada, özelliklerin ilgili sınıf için bağımsız olduğunu varsaymak, öğrenmeyi büyük ölçüde basitleştirmektedir. Bu nedenle algoritma Naive olarak adlandırılmıştır. Bazı makine öğrenimi sınıflandırıcı karşılaştırma çalışmaları, Naive Bayes Sınıflandırıcısı olarak bilinen basit bir Bayes öğrenme algoritmasının, sinir ağı sınıflandırma algoritmaları ile rekabetçi bir performansa sahip olduğunu bulmuştur. Ayrıca, büyük veri tabanlarına

uygulandığında Bayes sınıflandırıcıları iyi bir doğruluk ve yürütme süresi verimliliği göstermektedir (Wolff 2020).

3.18 KNN

KNN ifadesi İngilizcede "k-nearest neighbors" “k” en yakın komşu ifadesinin kısaltılmış halidir. Bu ifade KNN algoritmasının çalışma mantığını ifade eder. Knn algoritması tanımlı gözlemlere ait özellikler uzayında, henüz tanımlı olmayan bir gözlemin geri kalan tanımlı özelliklere uzaklığına göre bu yeni gözlemin tanımlanabilmesini ifade eder. Buradaki k bir yakınlık ölçüsünü ifade etmektedir(Lantz 2013).

KNN algoritması tanımlı gözlemlerin sınıflandırılmasına ve regresyon yapılabilmesine olanak sağlamaktadır. KNN algoritması basitliği ve kolay uygulanabilirliği nedeniyle yaygın kullanılan bir algoritmadır. Bu özelliğinden ötürü sıkça kullanılmaktadır. Birçok algoritma verilerin istatistiksel dağılımı ile ilgili olarak bazı varsayımları gerektirmesine rağmen KNN kullanılacağı durumlarda böyle bir varsayıma gereksinim olmamaktadır. Knn algoritmasının tahmin yapabilmesi için eğitimi hızlı olmaktadır (Lantz 2013).

KNN algoritması bir model üretilmesine olanak vermemektedir. Bu nedenle algoritma ilgili gözlem verileri içerisinde özelliklerin birbiri ile ilişkisine dair gizlenmiş yapı ve ilişkileri tespit etme yeteneğine sahip değildir. Ayrıca KNN algoritması kullanımında boş verilerin temizlenmesi ve normalize olmayan verilerin normalize edilmesi gereksinimi ek bir ön işlem sürecini gerektirmektedir (Lantz 2013). Knn algoritması bir sınıflandırma algoritması olduğundan belirli özellik bilgileri ve etiketleri verilen bir veri setinde özellik bilgilerinden yararlanarak gözlemleri sınıflandıracak yapıları modeller. Daha sonra etiketsiz olarak verilen herhangi bir verinin hangi sınıfa dâhil olduğunu önemli bir doğrulukta tahmin edebilmektedir.

KNN uygulaması kolay ve esnek bir sınıflandırma yöntemidir. KNN sınıflandırması, verilerin genellikle sayısal olduğu durumu öğrenmek için kullanılır. KNN, verileri merkezi olarak depolayıp işlediği için parametrik olmayan bellek tabanlı bir m.ö. yaklaşımıdır.

Tablo 1. Veri Seti

MSISDN	AUTOMATIC PAYMENT FLAG	PHONE NO CHANGE FLAGM1	SUM ARPU M1	SUM ARPU TL M1	GENDER	AGE	MOST LIVED CELL CITY M3	SUM GPRS USAGE KB M1	SUM GPRS USAGE KB M6	SUM GPRS USAGE KB M12
5779254817	N	N	4	71	1	49	ISTANBUL	5,496,629	58,224,357	121,592,264
5756730361	N	N	13	242	1	41	ESKISEHIR	6,932,558	63,794,056	83,186,084
5705409513	N	N	2	34	-1		MARDIN	204,011	11,779,807	34,569,125
5727870972	Y	N	4	74	1	57	ADANA	3,233,123	3,385,258	3,385,258
5788847102	N	N	0	0	1	30	UNKNOWN	0	0	0
5721662810	N	N	3	61	-1		TRABZON	15,864,427	52,393,087	52,393,087
5718176402	N	N	3	56	1	30	ADIYAMAN	52,818,256	178,350,560	457,056,949
5757604072	N	N	0	0	2	42	BURSA	0	44,813,312	44,813,312
5794217952	N	N	0	1	-1		SIIRT	3,487	24,959	52,537
5734663996	N	N	10	188	1	21	MERSIN	52,068,277	170,550,504	170,550,504
5726384889	N	N	4	65	1	45	ISTANBUL	3,755,307	28,403,545	45,427,611
5716262419	N	N	4	68	2	25	ISTANBUL	16,150,096	83,943,775	166,825,732
5733269701	Y	N	3	53	2	15	IZMIR	5,923,789	40,926,084	77,666,115
5787369571	N	N	2	43	-1		GAZIANTEP	9,614,116	69,496,542	136,314,352
5715587207	N	N	2	35	2	28	RIZE	4,514,991	36,921,596	67,051,660
5736242562	Y	N	20	367	1	72	ANTALYA	2,544,127	84,453,130	163,514,034
5720348537	Y	N	0	0	-1		UNKNOWN	0	0	0
5767436311	N	N	0	0	-1		UNKNOWN	0	0	0
5770546009	N	N	0	0	2	72	BURSA	0	184,934	184,934
5822880999	Y	N	7	126	2	29	ISTANBUL	2,692,271	20,972,183	44,501,354
5818921518	N	N	2	40	2	51	ISTANBUL	1,766,535	7,374,831	17,346,953
5900809229	N	N	3	60	1	54	ISTANBUL	0	0	0

5887932222	Y	N	3	62	2	12	BALIKESIR	558,286	9,354,410	16,623,327
5609470408	N	N	6	103	1	21	BALIKESIR	11,057,441	134,335,759	326,244,030
5726536803	N	N	3	51	1	23	MANISA	30,039,766	145,342,270	359,671,825
5824954687	N	N	3	60	2	58	GIRESUN	3,066,735	15,709,757	32,949,974
5984526160	N	N	6	106	1	50	RIZE	8,853,358	54,398,734	92,422,671
5815352990	N	N	2	39	1	42	IZMIR	820,650	2,578,144	3,846,343
5969340891	N	N	5	93	2	31	ISTANBUL	25,516,348	45,016,301	45,016,301
5778485064	N	N	3	46	1	33	BINGOL	0	0	539
5877801350	N	N	4	80	1	15	SANLIURFA	14,914,220	61,525,509	105,800,706
5957672848	N	N	0	5	2	16	ANKARA	69,002	337,363	469,744
5859847692	N	N	0	1	-1		OSMANIYE	2	1,265	1,265
5787319882	N	N	0	1	2	39	IZMIR	0	3,171,270	4,023,983
5873578170	N	N	3	50	1	69	RIZE	193,542	625,994	1,775,374
5827756300	N	N	5	97	2	9	SANLIURFA	49,210,175	254,075,558	524,365,287
5835578989	Y	N	3	52	2	41	KIRKLARELI	1,171,431	11,721,989	17,582,607
5849366165	N	N	0	0	2	24	GAZIANTEP	11,683,548	11,712,252	11,712,252
5877052845	N	N	2	38	2	56	MANISA	128,811	717,191	4,668,889
5874543399	Y	N	3	62	1	63	SANLIURFA	7,317,493	49,856,529	78,514,352
5868772858	N	N	3	59	1	38	ADANA	10,892,884	50,258,229	72,766,186
5849835949	Y	N	3	54	1	28	IZMIR	18,647,916	118,889,021	220,762,816
5905542935	N	N	15	269	-1		KOCAELI	165,901,79	386,814,714	487,052,327
5993168620	N	N	4	74	1	56	KIRIKKALE	1,392,024	10,458,248	24,022,452
5927227741	Y	N	3	58	1	71	BURSA	226,927	1,996,582	4,887,299
5942190339	N	N	0	0	1	43	UNKNOWN	0	14,339,433	14,339,433
5914508122	N	N	11	197	1	60	GAZIANTEP	53,497,628	311,600,108	651,509,702
5701021769	N	N	5	86	1	64	GIRESUN	11,322,836	66,062,611	66,065,363
5735810701	N	N	3	61	1	30	TOKAT	0	0	0
5752947973	N	N	1	10	-1		ISTANBUL	3,763	19,566	44,778

5757688481	N	N	14	253	1	58	BURSA	9,786,088	30,080,996	41,859,206
5898453544	N	N	3	53	1	48	KONYA	15,637,016	89,577,033	198,976,579
5836686431	Y	N	4	68	1	54	ANTALYA	6,476,364	41,027,317	84,460,014
5872102566	N	N	2	29	1	30	YOZGAT	45,742	273,278,188	804,544,905
5854456997	N	N	5	92	1	41	MUGLA	29,318,658	116,952,047	216,759,976
5826944229	N	N	3	54	2	25	DENIZLI	1,157,525	16,384,030	28,879,588
5844946771	N	N	4	69	1	48	TRABZON	0	13	14
5947414267	N	N	3	51	1	30	ISTANBUL	162,244	736,495	1,527,591
5992039095	N	N	7	129	1	41	MERSIN	29,113,821	86,918,637	119,281,844
5930396945	N	N	4	74	1	55	ANKARA	0	0	517,967
5959120258	N	N	5	82	1	6	KIRIKKALE	17,173,362	54,974,414	84,218,257
5984409770	N	N	0	0	1	55	KONYA	96,959	96,959	885,740
5017430567	N	N	5	85	1	39	ZONGULDAK	0	42	42
5009308903	N	N	5	98	1	43	ISTANBUL	34,596,391	124,463,782	302,034,961
5031988144	N	N	2	41	1	34	ANTALYA	1,459,940	13,141,522	36,667,077
5055510960	N	N	0	0	1	53	MERSIN	0	129,807,850	129,807,850
5095008362	N	N	4	64	1	79	ANTALYA	0	0	2,272,422
5014619458	N	N	7	119	1	52	KONYA	25,704,435	99,986,527	198,916,237
5837354271	N	N	6	112	1	22	KONYA	14,882,536	39,889,731	39,889,731
5897928660	N	N	2	34	1	37	ANKARA	4,035,615	19,821,670	41,612,232
5817843554	N	N	4	80	-1		SAKARYA	17,100,178	115,107,522	172,627,595
5863136002	N	N	0	1	-1		ISTANBUL	669	3,476	7,581
5899489377	Y	N	5	87	1	56	AYDIN	8,304,036	49,114,547	76,352,845
5865168320	Y	N	1	18	-1		HATAY	0	0	0
5852500468	N	N	4	67	1	42	KAYSERI	2,891,718	15,069,108	22,904,683
5997824058	N	N	5	90	1	45	SIVAS	26,467,643	122,008,850	265,652,023
5797309285	N	N	13	236	-1		ISTANBUL	32,564,118	174,650,519	297,841,695
5907811351	N	N	3	53	1	79	ADANA	8,843,940	40,139,750	98,073,302

5802439602	N	N	3	55	2	14	ADIYAMAN	12	81	126
5852558638	N	N	3	60	1	47	MERSIN	7,643,480	38,522,731	46,919,667
5748133492	N	N	0	5	-1		UNKNOWN	0	303,242	3,511,074
5840670877	N	N	9	156	1	26	KOCAELI	111,970,35	713,012,813	1,215,439,149
5702941185	N	N	5	94	1	22	ORDU	30,518,463	117,790,843	271,320,626
5740314257	N	N	0	0	2	43	ISTANBUL	0	19,893,795	19,893,795
5758792026	N	N	0	0	1	22	UNKNOWN	0	0	0
5764970570	N	N	0	0	1	49	ISTANBUL	299	94,040	166,952
5705118104	Y	N	0	0	-1		SIIRT	1,883	9,333	20,018
5754781428	N	N	8	143	1	65	BURSA	22,743,194	65,798,268	109,422,994
5759743839	N	N	2	45	-1		KIRKLARELI	299,386	863,894	967,390
5713843224	N	N	1	19		123	UNKNOWN	0	0	127,325
5754604561	N	N	1	16	2	77	TRABZON	3,065,515	23,486,987	23,486,987
5730332448	N	N	6	110	1	61	ANTALYA	12,520,125	73,913,992	142,075,502
5745181230	N	N	3	51	1	46	ARTVIN	2,807,789	15,867,443	58,887,610
5808597211	N	N	4	74	2	30	IZMIR	12,619,974	56,119,616	79,424,174
5870511990	N	N	6	104	2	57	ANTALYA	19,567,162	93,782,331	193,779,518
5823832403	N	N	5	99	1	53	SAMSUN	11,883,683	56,220,287	158,849,508
5868256619	N	N	4	81	1	48	ISTANBUL	990,704	4,668,796	20,409,802
5814080623	N	N	7	132	2	37	IZMIR	7,042,132	95,729,223	156,317,364
5865140985	N	N	0	0	2	49	ISTANBUL	0	408,744	408,744
5816964753	N	N	2	40	1	45	ISTANBUL	6,215,140	30,303,649	48,702,828

Tablo 2. Makine Öğrenmesi Örnek Çıktılar- 1

	Train rate	Test rate	Result	Train rate	Test rate	Result
Logistic Regression	%100	%100	84.69	%100	%70	74.28
Decisions Tree	%100	%100	100	%100	%70	100
Adaboost	%100	%100	75.53	%100	%70	73.26
Random Forest	%100	%100	77.11	%100	%70	67.53
Xgboost	%100	%100	95.57	%100	%70	90.26
Gaussian Naive Bayes	%100	%100	83.28	%100	%70	78.92
Knn	%100	%100	99.8	%100	%70	96.7

Tablo 2 de görüldüğü gibi eğitim datasını sabit tutarak eğitim datasında değişiklik yapılmaktadır. Skorlarda görüldüğü gibi doğrusal oran farklılık göstermektedir.

Tablo 3. Makine Öğrenmesi Örnek Çıktılar- 2

	Train rate	Test rate	Result	Train rate	Test rate	Result
Logistic Regression	%50	%50	76.54	%70	%100	57.6
Decisions Tree	%50	%50	96.3	%70	%100	68.6
Adaboost	%50	%50	70.6	%70	%100	66.6
Random Forest	%50	%50	67.8	%70	%100	61.3
Xgboost	%50	%50	88.9	%70	%100	86.5
Gaussian Naive Bayes	%50	%50	73.6	%70	%100	68.4
Knn	%50	%50	96.7	%70	%100	84.3

Her model de aynı oranda eğitim datası ve test data doğruluk oranı aynı olmamaktadır. Birkaç denemenin ardından az data ile en yüksek skor için tüm data eğitilmektedir. Test edilirken bu oranlar değişiklik gösterebilmektedir. En iyi oranda gerçekçi bir sonuç için tüm eğitim datasının üzerinden tümünü kapsayacak bir skor için test oranını %100 oranında tercih edilmektedir. Test aşamasında 250 ve 900 satır kullanılmıştır. Bunun sonucunda 7 model içerisinde bu datayı en iyi skorlayan ve oran olarak en yükseği gösteren modeller sırası ile Knn, Decisions Tree, Xgboost olmaktadır.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 250 entries, 0 to 250
Data columns (total 10 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   payment    250 non-null    int64
 1   gender     250 non-null    int64
 2   cc_cell    250 non-null    str
 3   sum_arpu   250 non-null    int64
 4   auto_py    250 non-null    int64
 5   arpu_tl    250 non-null    int64
 6   gprs_kb    250 non-null    int64
 7   gprs_m12   250 non-null    float64
 8   flag       250 non-null    str
 9   gprs_m6    250 non-null    float64
 10  age        250 non-null    int
dtypes: float64(2), int64(6), str(2)
memory usage: 136.0 KB
```

Şekil 6. Örnek Data Frame- 1

Şekil 6 da görüldüğü gibi 250 satır ile deneme yapılmıştır. Boş satır bulanmamaktadır ve int63, str, float64 tipinde data bulunmaktadır. Bu da toplamda 136 KB alan kaplamaktadır.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 900 entries, 0 to 900
Data columns (total 10 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   age        900 non-null    int64
 1   gender     900 non-null    int64
 2   cc_cell    900 non-null    str
 3   sum_arpu   900 non-null    int64
 4   payment    900 non-null    int64
 5   arpu_tl    900 non-null    int64
 6   gprs_kb    900 non-null    int64
 7   gprs_m12   900 non-null    float64
 8   flag       900 non-null    str
 9   gprs_m6    900 non-null    float64
 10  auto-py    900 non-null    int64
dtypes: float64(2), int64(7), str(1)
memory usage: 136.0 KB
```

Şekil 7. Örnek Data Frame- 2

Şekil 7 de görüldüğü gibi 900 satır ile deneme yapılmıştır. Boş satır bulanmamaktadır ve int63, str, float64 tipinde data bulunmaktadır. Bu da toplamda 136 KB alan kaplamaktadır.

3.19 Veri Sonuçları

Öncelikle veriler şekil 8'deki gibi sisteme yüklenmektedir. Şekil 9 da 1000 satırlık veri ile en iyi oranda gerçekçi bir sonuç için eğitim datası oranı ve test datası oranı %100 tercih edilmektedir. Bunun sonuncunda 7 modele ait skorlar şekil 9'daki gibi olmaktadır. Şekil 10' da ise 7 modele ait elde edilen skorlardan en iyi skor üreten ilk 3 model karar ağacı, rastgele orman, lojistik regresyon olmaktadır.

Şekil 9. Veri Setini Yükleme

Şekil 8. Training Data Sonuçları

Şekil 10. En iyi sınıflama modelleri

4. SONUÇ VE ÖNERİLER

Veri seti bazı sınıflama modellerinde %80 başarı oranı yakalamışken bazı sınıflama modellerinde ise %60 başarı oranı yakalamıştır. Makine öğrenim yöntemleri ile veri %80 oranında eğitilerek, anlamlı bir hale getirilmiştir. Tüm modellerin kendisine ait artıları ve eksileri bulunmaktadır. Bazı modellerde yüksek başarı oranı yakalanmışken daha uzun zamanlar harcanmıştır. Bazı modellerde veri daha hızlı eğitilmiştir fakat başarı oranı düşük kalmıştır. Bu çalışmada başarı oranı hedef alınarak ilerlenmiştir.

Kullanılan yöntemler arasında en uyumlu modelleme kullanılarak veriden en iyi sonuç elde edilmesi sağlanmıştır. Bu sonuçları elde etmek için deneme yanılma yöntemi ve k-fold yöntemlerinden gerektiğinde faydalanılmıştır. Verilerin gizliliği açısından msisdn kısımları değiştirilerek kullanılmıştır.

Literatürde daha önceden oluşturulmuş dokümanlar incelendiğinde bir modelleme yönteminin diğer modelleme yöntemine göre avantajları, dezavantajları kıyaslanmıştır. Bu araştırmada ise veriyi çeşitli sınıflandırma modelleriyle eğiterek en iyi şekilde anlamlandıracak modelleme sınıfı belirlenmesi sağlanmıştır. Sonucunda, başarı oranı %80 olan bir model oluşturulmuştur. Başarı oranının artması için daha fazla veri ve modelleme sınıfı ile çalışılmasına ihtiyaç vardır.

KAYNAKLAR

Basak J. ve Kothari R. (2004). "A classification paradigm for distributed vertically partitioned data." 16(7): 1525-1544. DOI: 10.1162/089976604323057470

Bergin S. (2006). Statistical and machine learning models to predict programming performance, *National University of Ireland Maynooth.*

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

Çelik M. (2009). "Veri madenciliğinde kullanılan sınıflandırma yöntemleri ve bir uygulama." (Yayın no. 261854) [Yüksek Lisans / Doktora Tezi, İstanbul Üniversitesi]. https://tez.yok.gov.tr/UlusalTezMerkezi/tezSorguSonucYeni.jsp

Dietterich T. G. (2000). Multiple Classifier Systems. Ensemble methods in machine learning. International workshop on multiple classifier systems. Springer Berlin Heidelberg, 2013.

Jain A. (2010). Pattern Recognition Letters "Data clustering: 50 years beyond K-means." 31(8): 651-666. Erişim tarihi: 23.12.2022 https://www.sciencedirect.com/science/article/abs/pii/S0167865509002323 adresinden alınmıştır.

Patel A. ve Thakral P. (2016). The best clustering algorithms in data mining. 2016 International Conference on Communication and Signal Processing (ICCSP), *IEEE.*

Vatansever, M. (2008). "Görsel veri madenciliği tekniklerinin kümeleme analizlerinde kullanımı ve uygulanması." (Yayın no. 237170) [Yüksek Lisans / Doktora Tezi, İstanbul Üniversitesi] https://tez.yok.gov.tr/UlusalTezMerkezi/tezSorguSonucYeni.jsp

Archer K. ve Kimes R. (2008). "Empirical characterization of random forest variable importance measures." 52(4): 2249-2260. Erişim tarihi: 17.11.2022 https://www.sciencedirect.com/science/article/abs/pii/S0167947307003076 adresinden alınmıştır.

Charbuty B. ve Abdulazeez A. (2021). "Classification based on decision tree algorithm for machine learning." 2(01): 20-28. Erişim tarihi: 30.11.2022 https://jastt.org/index.php/jasttpath/login?source=%2Findex.php%2Fjasttpath%2Farticle%2Fview%2F65 adresinden alınmıştır.

Hong R. ve Y. Zheng (2009). "Clustering analysis of telecommunication customers." 16(2): 114-128. Erişim tarihi: 15.12.2022 https://www.sciencedirect.com/science/article/abs/pii/S1005888508602149 adresinden alınmıştır.

Janiesch, C., Zschech P., Heinrich K. (2021). "Machine learning and deep learning." 31(3): 685-695. Erişim tarihi: 08.09.2022 https://link.springer.com/article/10.1007/s12525-021-00475-2 adresinden alınmıştır.

Kalousis, A. ve J. H. Gama, Melanie (2004). "On data and algorithms: Understanding inductive performance." 54(3): 275-312. DOI:10.1023/B:MACH.0000015882.38031.85

Kotsiantis, S., Zaharakis I., Panayiotis E. (2006). "Machine learning: a review of classification and combining techniques." 26(3): 159-190. DOI:10.1007/s10462-007-9052-3

Lantz, B. (2013). "Machine Learning with R." Erişim tarihi: 21.9.2022 https://books.google.com.tr/books?id=ZQu8AQAAQBAJ. adresinden alındı.

Lee, S. J., ve Siau K. (2001). "A review of data mining techniques." Erişim tarihi: 10.01.2023.https://www.emerald.com/insight/content/doi/10.1108/02635570110365989/full/html adresinden alınmıştır.

Liu, Y. ve Wang, Y. (2012). New machine learning algorithm: Random forest. International Conference on Information Computing and Applications, *Springer*.

Osisanwo, F., Akinsola, J. A., Hinmikaiye, O., Olakanmi J., Akinjobi, O. (2017). "Supervised machine learning algorithms: classification and comparison." 48(3): 128-138. DOI:10.14445/22312803/IJCTT-V48P126

Patel, K. A. ve P. Thakral (2016). The best clustering algorithms in data mining. 2016 International Conference on Communication and Signal Processing (ICCSP), *IEEE*.

Science, (2018). "What is logistic regression?". Erişim tarihi: 15.01.2023 https://www.mastersindatascience.org/learning/machine-learning-algorithms/logistic-regression/ adresinden alınmıştır.

Sharifzadeh M., Alexandra Shah, Nilay (2019). "Machine-learning methods for integrated renewable power generation: A comparative study of artificial neural networks, support vector regression, and Gaussian Process Regression." 108: 513-538. Erişim tarihi : 21.12.2022 https://www.sciencedirect.com/science/article/abs/pii/S1364032119301807 adresinden alınmıştır.

Sharma, H. ve Kumar S. (2016). "A survey on decision tree algorithms of classification in data mining." 5(4): 2094-2097. Erişim tarihi: 10.12.2022 https://www.researchgate.net/publication/324941161_A_Survey_on_Decision_Tree_Algorithms_of_Classification_in_Data_Mining adresinden alınmıştır.

Soofi, A. A. ve Awan A. (2017). "Classification techniques in machine learning: applications and issues." 13: 459-465. DOI:10.6000/1927-5129.2017.13.76

Şirin, E. (2017). "Büyük Veri Ön-İşleme." Erişim tarihi: 11.12.2022 https://www.veribilimiokulu.com/buyuk-veri-on-isleme-makale-notlari/. adresinden alındı.

Vatansever, M. (2009). "Görsel Veri Madenciliği." Erişim tarihi: 05.10.2022 http://visualdatamining.blogspot.com/2009/06/veri-madenciligi-yontemlerinin.html adresinden alınmıştır.

Velmurugan, (2014). "Performance based analysis between k-Means and Fuzzy C-Means clustering algorithms for connection oriented telecommunication data." 19: 134-146. DOI:10.1016/j.asoc.2014.02.011

Wei, L. ve Y. Yang (2005). "A study on several machine-learning methods for classification of malignant and benign clustered microcalcifications." 24(3): 371-380. DOI:10.1109/TMI.2004.842457

Wolff, R. (2020). "5 Types of Classification Algorithms in Machine Learning." Erişim tarihi: 10.01.2023 https://monkeylearn.com/blog/classification-algorithms/ adresinden alınmıştır.

ÖZGEÇMİŞ

İhtiman Emre BİLGE

Eğitim

Derece	**Yıl**	**Üniversite, Enstitü, Anabilim**
Y. Lisans	2021	Maltepe Üniversitesi, Lisansüstü Eğitim Enstitüsü
Lisans	2007	Bilgisayar Mühendisliği Anabilim Dalı Ankara Üniversitesi/ Fen Fakültesi
Lise	2003	Ankara Kılıçarslan Lisesi, Fen-Matematik

İş/İstihdam

Yıl	**Görev**
2018- -	İş Analisti Turkcell
2014-2018	İş Analisti Ericsson
2013-2013	Fatih Projesi Takım Lideri Askom
Bilgisayar	
2010-2013	Kurucu/Yönetici Anka Planetaryum