T.C.
KONYA TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

TVİTER VERİLERİ ÜZERİNDE SINIFLANDIRMA ALGORİTMALARI KULLANARAK HİSSE SENEDİ DEĞERLERİ İÇİN YÖN TAHMİNİ

Mustafa Vehbi TÜRKALP

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

Temmuz - 2019
KONYA

TEZ KABUL VE ONAYI

Mustafa Vehbi TÜRALP tarafından hazırlanan "Tviter Verileri Üzerinde Sınıflandırma Algoritmaları Kullanarak Hisse Senedi Değerleri İçin Yön Tahmini" adlı tez çalışması 22/07/2019 tarihinde aşağıdaki jüri tarafından oy birliği / oy çokluğu ile Konya Teknik Üniversitesi Lisansüstü Eğitim Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı'nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Jüri Üyeleri	**İmza**
Başkan Dr. Öğr. Üyesi Mehmet Hacıbeyoğlu	.....
Danışman Doç. Dr. Barış Koçer	.....
Üye Dr. Öğr. Üyesi Ersin Kaya	.....

Yukarıdaki sonucu onaylarım.

Prof. Dr. Hakan KARABÖRK
Enstitü Müdürü

TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Mustafa Vehbi TÜRKALP
22.07.2019

ÖZET

YÜKSEK LİSANS TEZİ

TVİTER VERİLERİ ÜZERİNDE SINIFLANDIRMA ALGORİTMALARI KULLANARAK HİSSE SENEDİ DEĞERLERİ İÇİN YÖN TAHMİNİ

Mustafa Vehbi TÜRKALP

Konya Teknik Üniversitesi
Lisansüstü Eğitim Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Barış KOÇER

2019, 83 Sayfa

Jüri
Doç. Dr. Barış KOÇER
Dr. Öğr. Üyesi Ersin KAYA
Dr. Öğr. Üyesi Mehmet HACIBEYOĞLU

Borsa, gerek hisselerin kolay alınıp satılması, gerek sağladığı gelir ve gerekse verilere kolay erişim bakımından sağladığı avantajlarla her zaman yatırımcıların gözde yatırım aracı olagelmiştir. Yatırımcılar da bu platformda daha fazla gelir elde edebilmek adına, hisse senetlerinin ileriye dönük yön tahmini ile sürekli ilgilenmişlerdir. Bunun için de çok farklı teknikler geliştirilmiştir. Biz de bu çalışmamızda, hisse senetlerinin fiyatlandırılmasında asıl kriter olarak düşündüğümüz "arz-talep" ilişkisinden yola çıkarak, hisseye olan talebin artması veya azalmasının önceden tahminini yapmak için, günümüzde en fazla kullanılan sosyal medya paylaşım platformlarından biri olan Twitter mesajlarının sınıflamasını yaptık.

Araştırmamızda, Amerikan Dow Jones (DJIA) borsasında işlem gören Apple, Facebook, General Electric, General Motors, The Coca-Cola Company, McDonald's, Microsoft, Netflix, Pfizer Corporation, Tesla Motors gibi dünya çapında firmaların hisse senetleri için atılan tivitleri analiz ettik.

Sınıflandırma işlemini, naive bayes, rastgele orman, destek vektör makinesi, karar ağacı, k-en yakın komşu ve yapay sinir ağları sınıflandırma algoritmalarını kullanarak gerçekleştirdik ve bu algoritmaların başarım sonuçlarını karşılaştırdık.

İlgili hisse senetleri için Nisan 2019 – Mayıs 2019 tarihlerini kapsayacak şekilde iki aylık veri, twitter web arayüzü kullanılarak elde edildi. Benzer şekilde yön tahmini başarı seviyesinde test amacı ile kullanılan hisse senedi değerlerini içeren dosyalar da yine www.eoddata.com web adresi üzerinden elde edildi.

Tivitleri etiketleme işlemi, borsa bilgileri birbirinden farklı, 75 farklı katılımcının tivitleri tek tek okuyup eli ile pozitif, negatif veya nötr olarak işaretlemesi ile yapıldı. Herhangi bir duygu belirtmeyen tivitlerin yanı sıra, anlaşılmayan, reklamdan ibaret olan, sadece bir link verilmiş olan v.b. tivitlerin tamamı nötr sınıfına dahil edildi. Algoritmaların sınıflandırma başarılarının ölçülmesinde değil ancak hisse senedinin yön tahmininin başarısı hesaplanırken, çok fazla çöp tivit içerdiği için nötr sınıfı göz ardı edilidi.

Sınıflandırma için öncelikle tivitleri, noktalama işareleri, hyperlinkler ve web adresleri, "tab" karakteri, hashtaglar, retweetler, birbiri ile aynı olan tekrarlanmış tivitler v.b. fazlalıkları silerek temizledik. Temizlenen bu veri seti üzerinde sınıflandırma için, makine öğrenmesi teknikleri kullanılarak sınıflandırma yaptık. TF-IDF yöntemi kullanılarak her bir veri setinde her bir tivit için geçen tüm kelimelerin frekans ağırlıklarını hesaplayarak vektör haline dönüştürüp sayısallaştırdık. Sayısallaştırdığımız bu veriler üzerinde 6 farklı sınıflandırma algoritması kullanılarak başarım sonuçlarını elde ettik.

Tivitlerin sınıfını tahmin etmek için yaptığımız sınıflandırma işlemi için belirlediğimiz algoritmaları kullanırken, homojenliği sağlamak için corss validation yöntemi ile veri setlerini 10 parçaya ayırdık. Bu parçalardan her birisini sıra ile doğrulama maksadı için kullandık, diğer parçaları ise sistemin

eğitimde kullandık. En sonunda da tüm parçalar bittiği zaman, 10 parçanın ortalamasını alınarak genel tahmin başarısını elde ettik.

Araştırma sonunda sınıflandırma işlemini gerçekleştirdiğimiz veri seti üzerinde %77,37 ile en başarılı sonucu rastgele orman algoritması verirken, %61,41 ile en kötü sonucu destek vektör makinesi verdi. Yine sınıflarını tahmin etmeye çalıştığımız hisse senetleri tivitleri için en iyi tahmin başarısı %83,3 ile GM'a ait iken en kötü tahmin başarısı ise %62,15 ile GE'ye ait olarak bulundu.

Hisse senetlerinin yön tahmin başarı sonuçlarının değerlendirilmesinde ise en başarılı tahmin %96,5 ile KO için yapılırken, en kötü tahmin ise %66,7 ile TSLA için yapılmıştır. Bu tahmin başarılarının hesaplanmasında nötr tivitler göz ardı edilerek sadece pozitif ve negatif etiketli tivitler dikkate alınmıştır. Pozitif olarak işaretlenmiş olan bir tivit, ertesi gün hisse senedi negatif yönlü bir hareket yapmadıkça başarılı bir tahmin yapmış olarak alınmıştır, aynı şekilde negatif olarak işaretlenmiş olan bir tivit, ertesi gün ilgili hisse senedi yükselmemişse başarılı olarak alınmıştır.

Elde edilen bulgular sonucu hisse senetlerinin yön tahminlerinin yapılmasında twitter verilerinin kullanılabileceği, gayet başarılı sonuçlar elde edilebileceği görülmüştür.

Anahtar Kelimeler: Sınıflandırma, Twitter, Borsa, Yön tahmini, Naïve Bayes, Destek Vektör Makinesi, Karar Ağacı, Yapay Sinir Ağları, k-En Yakın Komşu, Rastgele Orman, Genetik Algoritmalar.

ABSTRACT

MS THESIS

DIRECTION ESTIMATION FOR STOCK VALUES BY USING CLASSIFICATION ALGORITHMS ON WITTER DATA

Mustafa Vehbi TÜRKALP

Konya Technical University
Institute of Graduate Studies
Department of Computer Engineering

Advisor: Assoc. Prof. Dr. Barış KOÇER

2019, 83 Pages

Jury
Assoc. Prof. Dr. Barış KOÇER
Asst. Prof. Dr. Ersin KAYA
Asst. Prof. Dr. Mehmet HACIBEYOĞLU

The stock market has always been the favorite investment instrument of investors with the advantages it provides. Some of them are easy buying and selling of the shares, size of proceed and the easy access to the data. In order to generate more revenue on this platform, investors were also constantly interested in the forward-looking direction forecasting of stocks. Many different techniques have been developed for this reason. In this study, we made the classification of twitter messages -which is one of the most widely used social media sharing platforms- in order to predict the increase or decrease of the demand for the stock, because of we considered that the main criterion of pricing of stock is "supply-demand" relationship.

In our research, we analyzed the tweets for stocks of global companies such as Apple, Facebook, General Electric, General Motors, The Coca-Cola Company, McDonalds, Microsoft, Netflix, Pfizer Corporation, Tesla Motors, which are traded on the American Dow Jones (DJIA) stock exchange.

We performed classification, using naive bayes, random forest, support vector machine, decision tree, k-nearest neighbor and artificial neural network classification algorithms and compared the performance results of these algorithms

Bi-monthly data for the related stocks -covering April 2019 - May 2019- were obtained using the twitter web interface. Similarly, files containing the stock values used for determination of success level of direction estimation and testing purposes were also obtained from the www.eoddata.com web address.

The labeling process was performed by 75 different participants which have various stock market knowledge, by reading the tweets one by one and marking them manually as positive, negative and neutral. In addition to senseless tweets, the tweets consisting of a link, consisting of an advertisement, vague tweets etc. were labeled as neutral. Because of containing a lot of rubbish tweets, the neutral class was ignored when calculating the success of direction estimation of the stock values. But they were not ignored the measuring the success of the classification algorithms.

For the classification, firstly we cleaned the tweets by removing the inessential factors as punctuation marks, hyperlinks and web addresses, "tab" characters, hashtags, re-tweets, repeating similar tweets etc. We classified using machine learning techniques for classification on this clear dataset. Using the TF-IDF method, we digitized by calculating the frequency weights of all words in each data set for each tivit and converted it into a vector. We obtained the performance results by using 6 different classification algorithms on these digitized data.

While we used the classification algorithms which we determined for classification process with machine learning method to estimate the class of tweets, we divided the data sets into 10 parts with the

corss validation method in order to make it homogenous. We used each of these parts in order for verification purposes, and used the other parts in the training of the system. Finally, when all the parts were finished, we achieved the overall prediction success by averaging 10 tracks.

At the end of the research, random forest algorithm gave the most successful result with 77.37% and the support vector machine gave the worst result with 61.41%. Also the best predictive success for the stocks we tried to predict the class belongs to GM with 83.3%, while the worst predicted success belongs to GE with 62.15%.

In the evaluation of success results of the direction estimation of the stocks, the most successful prediction was made for KO with 96.5%, and the worst estimation was made for TSLA with 66.7%. In the calculation of these predictive successes, only positive and negative tagged tweets were taken into account, while neutral tweets were ignored. A positive tweet was considered as a asuccessful estimate unless the stock made a negative move the following day, and a tivit marked as negative was considered as successfully if the corresponding stock did not rise the following day.

As a result of the findings, it was seen that twitter data can be used to make direction estimations of stocks and very successful results can be obtained.

Keywords: Classification, Twitter, Stock, Direction Estimation, Naïve Bayes, Support Vector Machine, Decision Tree, Artificial Neural Networks, k-Nearest Neighbour, Random Forest, Genetic Algorithms.

ÖNSÖZ

Çocuk gelişimine önem veren ailelerin çocuklarının her zaman hayata diğerlerine nazaran daha önde başladıklarını ve tüm yaşantıları boyunca da daha avantajlı olduklarını düşünüyorum. Ben de tüm hayatım boyunca böyle bir avantaja sahip olduğum için kendimi gerçekten şanslı hissediyorum. Bunun için benden desteğini hiçbir zaman esirgemeyen annem Keziban TÜRKALP ve babam Yusuf TÜRKALP'e sonsuz teşekkür ederim.

Bir öğrencinin vizyonunu belirleme, potansiyelini ortaya koymasında en önemli etkenlerden bir tanesi ilkokul öğretmenidir. İlkokul öğretmeniniz eğer sizin için bir sınır belirlemişse, bu sınırı geçmek bazen gerçekten çok zor olabilir. Bazen sizin kendinizi nerede gördüğünüz kadar, ilkokul öğretmeninizin sizi nerede gördüğü de önemlidir. Ben de hayatımın şekillenmesinde çok önemli bir yere koyduğum rahmetli ilkokul öğretmenim Şükrü GAGACI hocama ve şahsında üzerimde az veya çok emeği olan tüm öğretmenlerime ve bu tezi ortaya koymamda beni motive eden, yardımlarını esirgemeyen danışmanım Doç. Dr. Barış KOÇER hocama da sonsuz teşekkür ederim.

Sevgili eşim Gülhan TÜRKALP ve çocuklarım Türkü Efdal, Öykü Melda ve Bengü Meyra'dan da hem onlara çok fazla zaman ayıramadığım ve bazen ihmal ettiğim özür diler, hem de her zaman desteklerin en büyüğünü onlardan gördüğüm için çok teşekkür ederim.

Mustafa Vehbi TÜRKALP
KONYA - 2019

İÇİNDEKİLER

ŞEKİLLER

ÇİZELGELER

SİMGELER VE KISALTMALAR

Simgeler

AAPL	Apple firmasının DJIA sembolü
FB	Facebook firmasının DJIA sembolü
GE	General Electric firmasının DJIA sembolü
GM	General Motors firmasının DJIA sembolü
KO	The Coca-Cola Company firmasının DJIA sembolü
MCD	McDonald's firmasının DJIA sembolü
MSFT	Microsoft Corporation firmasının DJIA sembolü
NFLX	Netflix firmasının DJIA sembolü
PFE	Pfizer Corporation firmasının DJIA sembolü
TSLA	Tesla Motors firmasının DJIA sembolü

Kısaltmalar

AID	Automatic Incident Detector
API	Application Programming Interface
CVV	Card Validation Value
DA	Duygu Analizi
DJIA	Dow jones Industrial Avarage
DNA	Deoksiribo Nükleik Asits
DT	Decision Tree
DVM	Destek Vektör Makinesi
F/K	Fiyat / Kazanç
GA	Genethic Algorithms
GPOMS	Google Profile of Mood States
LDA	Latent Dirchlet Allocation
ME	Maksimum Entropi
ML	Machine Learning
NB	Naive Bayes
NKB	Nokta Tabanlı Karşılıklı Bilgi
OM	Opinion Mining
OOB	Out of Bag
PD/DD	Piyasa Değeri / Defter Değeri
POS	Part of Speech
RF	Random Forest
RO	Rastgele Orman
SA	Sentiment Analysis
SVM	Support Vector Machine
TF-IDF	Term Frequency - Inverse Document Frequency
YSA	Yapay Sinir Ağları

1. GİRİŞ

Günümüzde sosyal medya, her geçen gün çeşitliliği, kullanım alanları, kullanım amaçları ve kullanıcı sayısı artan en önemli iletişim araçlarından biri haline gelmiştir. Hayatımızın içine bu kadar girmiş olan bu olgu, başlangıçta farklı amaçlar için düşünülmüş olsa da bugün sosyal, ekonomik, ticari, dini, bilimsel, kültürel ve daha birçok alana yön verir hale gelmiştir.

Ekonomi, çift yönlü etkileşime sahip olması itibariyle sosyal medyadan, diğer alanlara göre daha fazla etkilenebilmektedir. Örneğin menkul kıymetler borsasındaki bir hisse senedinin değerinde herhangi bir hareket olması sosyal medyada daha fazla konuşulmasına, sosyal medyada çok konuşuluyor olması da hisse senedinin değerinin değişmesine sebep olabilmektedir.

Bu araştırmada, sosyal medya ile menkul kıymetler borsası arasındaki bu çift yönlü etkileşimi sağlayan sosyal medya mesajlarını eş zamanlı olarak, yapay zekâ teknikleri ile analiz eden ve hisse senedi değerinin yükseleceği veya düşeceği yönünde tutarlı tahminlerde bulunabilen bir sistem tasarımı amaçlanmaktadır.

Bu tez yedi temel bölümden oluşmaktadır. Bu bölüm, tezin konusu, muhtevası ve tanıtımın yapıldığı giriş bölümdür. İkinci bölümde üzerinde çalıştığımız konu ile ilgili daha önceden yapılmış olan araştırmalar incelenmiştir. Üçüncü bölümde bizim tezimizin ne gibi farklılıklar ortaya koyduğu ve önemi ele alınmıştır. Dördüncü bölümde çalışmamızda kullanacağımız verilerin temini, içeriği ve ayrılması ile ilgili konular anlatılmaktadır. Beşinci bölümde metin sınıflandırma açıklanmış, metin sınıflandırmada kabul görmüş ve uygulanmakta olan yöntemler, algoritmalar ve teknikler ayrıntılı olarak anlatılmıştır. Altıncı bölüm, uygulamamızı gerçekleştirdiğimiz, gerçekleştirirken izlediğimiz yol ve elde ettiğimiz sonuçların sunulduğu bölümdür. Yedinci ve son bölümde ise, çalışmamızda elde ettiğimiz bulgular üzerine çıkardığımız sonuçlar ve konu ile ilgili yapılabilecek benzer çalışmalar irdelenmiştir

2. KAYNAK ARAŞTIRMASI

Metin sınıflandırma, uygulama alanı ve kapsamı çok geniş olan bir konudur. Bunun yanında metin sınıflandırma için kullanılabilecek teknikler de çeşitlilik arz edebilmektedir. Ayrıca kaynak olarak kullanılacak olan verilerin elde edileceği platform olarak da çok zengin bir konudur. Yapay zekâ da son dönemde yazılımcıların ilgi odağı haline gelmiş bir araştırma alanı olmuştur. Tüm bunları göz önünde bulundurursak, benzer konular üzerine yapılmış gerek Türkçe gerek İngilizce ve gerekse başka dillerde yazılmış sayısız araştırmaya ulaşmamız mümkündür.

Bu çalışmalar genel olarak metin sınıflandırma düzeyi (belge düzeyi, cümle düzeyi, ifade düzeyi), veri toplama şekli, kullanılan sınıflandırma tekniği, sınıflandırmada kullanılan algoritmalar, uygulama alanları bakımından farklılık göstermektedir. Bunlardan bazılarını bu bölümde inceleyeceğiz.

Metin sınıflandırma işlemi, birçok parçalı yapı seviyesinde doğal bir dil işleme aracı olarak ele alınmıştır. Belge düzeyi sınıflandırma problemlerinden başlanmış (Turney, 2002), (Pang ve Lee, 2004), cümle düzeyinde (Hu ve Liu, 2004), (Kim ve Hovy, 2004) ve daha yakın zamanlarda ifade düzeyinde ele alınmıştır (Wilson ve ark., 2005), (Agarwal ve ark., 2011). Kullanıcıların gerçek zamanlı tepkileri ve akıllarına gelen her şey hakkındaki görüşlerini yayınladığı Twitter gibi mikroblog verileri, daha yeni ve farklı zorluklar ortaya koymaya başlamıştır. Twitter verilerinin duygu analizi ile ilgili yapılan araştırmaların bazıları Go ve ark. (2009), Bermingham ve Smeaton (2010) ve Pak ve Paroubek (2010)'a ait olanlardır. Go ve ark. (2009), duygu verilerini elde etmek için uzaktan öğrenmeyi kullanır. ":)" ":-)" gibi pozitif ifadelerle biten tivitleri pozitif, ":(" ":-(" gibi negatif ifadelerle biten tivitleri ise negatif olarak işaretlerler. Naïve Bayes, Maxent ve destek vektör makineleri (SVM) kullanarak modeller oluştururlar ve SVM'nin diğer sınıflandırıcılardan daha iyi performans gösterdiği sonucuna varırlar. Özellik seti içinde, cümlenin ögeleri (*Parts Of Speech* - POS) özellikleri ile bir unigram ve bigram modeli denerler. Sonuçların karşılaştırılmasında unigram modeli, diğer modellerden daha iyi bir performans gösterir. Özellikle, bigram ve POS kabul edilebilir bir performans ortaya koymaz. Pak ve Paroubek (2010) de benzer şekilde bir uzaktan öğrenmeyi kullanarak veri toplar. Fakat onlar verileri daha farklı sınıflara ayırırlar: öznel ve nesnel. Öznel veriler için, emojilerle biten tivitleri Go ve ark. (2009) ile aynı şekilde toplarlar. Nesnel veriler için "New York Times", "Washington Posts" v.b. gibi popüler gazetelerin Twitter hesaplarını tararlar. POS ve bigram'ın her ikisinden de kabul edilebilir sonuçlar elde

ederler (Go ve ark., 2009). Burada kullanılan her iki yaklaşım da temelde n-gram modellerine dayanmaktadır. Ayrıca, eğitim ve test için kullandıkları veriler arama sorgularıyla toplanmıştur ve bu nedenle objektif değildir. Buna karşılık, bir unigram modeli ile önemli bir performans ortaya koymuşlardır. Buna ek olarak, farklı bir veri gösterimi yöntemi araştırmışlar ve unigram modelleri üzerinde önemli bir gelişme ortaya koymuşlardır. Bu makalenin bir diğer katkısı da verilerin, belirli sorguları kullanarak toplanan verilerin aksine, akış tivitlerinden rastgele alınmış bir örneklem kümesi olmasıdır. Twitter verilerinde duygu analizi (DA) için bir diğer önemli çalışma da Barbosa ve Feng (2010)'in yaptığı çalışmadır. Bir modeli eğitmek ve test etmek için 1000'er adet el ile etiketlenmiş tivit toplamışlardır. Retweet, hashtag, link, noktalama işaretleri ve ünlem işaretleri gibi tivitlerin sözdizimi özelliklerinin, kelimelerin polaritesi ve kelimelerin cümle içindeki görevleri gibi özelliklerle birlikte kullanılması yaklaşımını gerçekleştirmişlerdir. Gamon (2004) da küresel destek hizmetleri anketi verileri üzerinde duyarlılık analizi yapmıştır. Bu çalışmasının amacı kelimelerin cümle içindeki görevleri gibi dilsel özelliklerin rolünü analiz etmekti. Kapsamlı özellik analizi ve özellik seçimi yaparak soyut dilsel analiz özelliklerinin sınıflandırıcı doğruluğuna katkıda bulunduğunu gösterdi.

Bir başka araştırmada, Saif ve ark., (2012), duygu analizi için eğitim setine ek olarak semantik (anlambilim) ekleme konusunda yeni bir yaklaşım sunmuşlardır. Tivitlerden çıkarılan her kelimeye (örneğin iPhone) o kelimeyi anlamlandıran bir kavram eklemiş (örn. "Apple ürünü") ve ek bir özellik olarak ve bu temsilci kavramın da Negatif/Pozitif duygu ile ilişkisini değerlendirmişlerdir. Bu yaklaşımı, üç farklı twitter veri kümesi için duyguları tahmin etmek amacıyla uygulamışlardır. Araştırmanın sonucunda, bu yeni geliştirdikleri yaklaşımla pozitif ve negatif duyguları belirlemede n-gram F harmonik doğruluk skorunda ortalama %6,5 seviyelerinde bir artış sağlamışlardır.

Borsa yön tahminini twitter verilerinin duygu analizi ile yapmaya çalışan Bollen ve ark. (2010), büyük ölçekli twitter yayınlarından elde ettikleri toplumsal ruh hali durumlarının ölçümlerinin zaman içinde Dow Jones Industrial Average (DJIA) değeriyle ilişkili olup olmadığını araştırmışlardır. Günlük twitter beslemelerinin metin içeriğini analiz etmek için iki farklı araç kullanmışlardır (*OpinionFinder* ve *Google Profile Of Mood States* - GPOMS). Toplumsal ruh halinin Amerikan Dow Jones (DJIA) borsası üzerindeki etkilerini araştırmak için bahsedilen araçlardan elde ettikleri ortalama duyguları kullanmışlar ve %80,7 doğrulukla tahmin edebilmişlerdir. Ancak burada tüm duyguların bu tahminde kullanamayacağı, sadece bazılarının doğru sonuç verdiğini

görmüşleridir. Daha sonra bu duyguları kendi kendini eğiten bir bulanık - yapay sinir ağı ile analiz etmiş ve ortalama %6 lık bir iyileşme kaydederek %86,7 doğruluk seviyesine ulaşmışlardır. Bu çalışmanın üzerine Mittal ve Goel (2012) de aynı çalışmayı farklı veri kümeleri üzerinde yapmışlar ve genel olarak benzer sonuçlara varmışlarsa da bazı büyük farklılıklar da ortaya çıkmıştır. Bollen ve ark. (2010) yaptıkları çalışmada sadece "sakin" ruh hali ile DJIA arasında yüksek bir ilişki tespit etmelerine rağmen, Mittal ve Goel (2012), farklı olarak "sakin" ve "mutlu" ruh hallerinin baskın şekilde DJIA yönü ile ilişkili olduğunu tespit etmişlerdir. Ayrıca farklı olarak yön tahmininde %86,7 oranında bir başarı sağlayamamış, %75,56 oranında bir başarı sağlamışlardır.

Shehu (2019), Türkçe twitter verileri üzerinde duygu analizini kutupsallık sözlüğü ve yapay zekâ teknikleri kullanarak gerçekleştirmiştir. Twitter üzerinden çektiği 13.000 adet tiviti, kutupsallık sözlüğü ve makine öğrenmesi yöntemleri ile sınıflandırmıştır. Sınıflandırıcı olarak rastgele orman (*random forest* - RF) ve destek vektör makineleri (*support vector machines* - SVM) kullanmıştır. Bu çalışmasının sonunda DVM'nin işlenmiş veriler üzerinde daha hızlı bir şekilde sonuca ulaştığını, RO algoritmasının ise ham veriler üzerinde daha doğru sonuç verip daha iyi bir performans gösterdiği sonucuna varmıştır. Kutupsallık sözlüğü kullanılarak yapılan sınıflandırmada ise tivitlerin temizlenip köklerine inildikçe performansın da arttığını tespit etmiştir.

Kang ve Park (2014) ise duygu analizini daha başka bir alanda kullanarak, mobil hizmetlerde müşteri memnuniyetini ölçme problemini incelemiştir. Bu çalışmada veri toplama – ön işleme ve müşteri memnuniyetinin ölçülmesi olarak birbirini takip eden iki ana aşamayı kapsayan VIKOR yaklaşımını ortaya koymuşlardır. Duygu analizinin gücünü ve VIKOR yaklaşımını bir arada kullanarak önerilen yaklaşım, gerçek müşteri incelemelerini kullanarak müşteri memnuniyetini ölçmektedir. Bu yönetim çalışmalarında mobil uygulama hizmetleri üzerinde ampirik bir vaka üzerinde uygulamışlar ve çok çarpıcı, başarılı sonuçlar elde etmişlerdir. Hao ve ark. (2011) de benzer bir çalışmada, gerçek dünyadaki twitter veri akışlarında müşteri memnuniyetini araştırmak için duyarlılık ve akış analizlerini kullanmışlardır. Coğrafi ve zamana dayalı etkileşimli görselleştirmelerle birleştirilen twitter zaman serilerinin görsel bir analizini yapmışlar, duygu analizi tekniklerini müşterilerin geri bildirimlerini tanımlayan satın alma sonrası web anketi ve eğlence parkı twitter verilerine uygulamışlardır.

Filmlerin incelemelerinin otomatik duygu analizi ile sınıflandırılması problemi üzerine yapılan bir başka çalışmada da Ohana ve Tierney (2009), SentiWordNet sözlük kaynağını kullanmışlardır. Bu yaklaşımda, duyarlılık yönünü belirlemek için pozitif ve

negatif terim puanlarını saymışlar ve SentiWordNet'i kaynak olarak kullanan bir ilgili özellikler veri seti oluşturup iyileştirme sağlayarak makine öğrenme sınıflandırıcısına uygulamışlardır. SentiWordNet ile elde edilen sonuçların literatürde görülen manuel sözlükleri kullanarak benzer yaklaşımlarla uyumlu olduğu sonucuna varmışlardır. Bunun yanında, özellik seti yaklaşımının, temel terim sayma yöntemine göre iyileştirme sağladığını görmüşlerdir. SentiWordNet'in duygu sınıflandırma problemleri için önemli bir kaynak olarak kullanılabileceğini göstermişlerdir ve yöntemin, diğer tekniklerle birlikte kullanılmasının olası iyileştirmelere katkı sağlayabileceği çıkarımına ulaşmışlardır.

Bir başka çalışma da (Kaya ve ark., 2012), Türkçe siyaset haberleri üzerinde duygu analizi yapılmıştır. Bu çalışmada, haber sitelerinden kopyalanan haberler ile bir veri seti oluşturulmuştur. Veri setinde yalnızca siyasi içerikli haberler kullanılmıştır. Bu veriler üzerinde makine öğrenmesi algoritmaları uygulanmıştır. Kullanılan tüm algoritmalar, %65-%70 doğruluk seviyelerine ulaşarak başarılı olmuş, ancak sonuçlar karşılaştırıldığında en iyi sonuçları n-gram ve maksimum entropi (ME) modellerinden elde etmişlerdir. Destek vektör makinelerinin (DVM) ve Naïve Bayes (NB) yöntemlerinin ise performans olarak biraz daha geride kaldıklarını gözlemlemişlerdir.

Aynı grup duygu analizi konusunda yaptıkları bir sonraki çalışmalarında ise (Kaya ve ark., 2013) bu kez performansı arttırmak için yapay zekâ teknikleri ile etiketlenmemiş olan tivitleri, etiketli politik verilere çevirmişler, sonrasında da bu tüm veri setini pozitif ve negatif olarak sınıflandırmışlardır. Sınıflandırıcı olarak da bir önceki çalışmalarına benzer olarak ME, DVM ve NB kullanmışlardır. Çalışmanın sonunda performansın %26'ya varan oranlarda arttığını gözlemlemişlerdir.

3. TEZİN AMACI VE ÖNEMİ

3.1. Tezin Amacı

Zaman içerisinde borsa, yatırımcıların en gözde yatırım aracı haline gelmiştir. Bunu sağlayan en önemli sebep olarak da diğer yatırım araçlarının aksine, borsada, daha kısa zamanda daha yüksek karların elde edilebilmesi, yani daha spekülatif olmasını gösterebiliriz. İnsanlar, altın, döviz, gayrimenkul gibi diğer yatırım araçlarında, daha uzun sürelerde elde edebilecekleri karları, borsada, biraz daha fazla risk alarak çok daha kısa sürede elde edebilmektedir. Bu da yatırım aracı olarak borsayı daha popüler hale getirmektedir.

Borsadaki hisse senetlerinin yön tahmini üzerine daha önceden pek çok yaklaşım ortaya konmuştur. Bunlardan bazıları, şirketlerin reel verileri üzerinden, şirketin genel durumunu değerlendirmiş, buna göre bir çıkarımda bulunmuştur. Bu değişkenleri Özçam (1990), mikro değişkenler ve makro değişkenler olmak üzere iki ana başlığa ayırmıştır. Demir (2001) ise yaptığı çalışmasında hisse senetleri üzerinde etkili olan mikro değişkenleri ele almış ve bu değişkenleri hisse senedi fiyatları, kaldıraç oranı, temettü ödeme oranı, hisse başına kar, öz sermaye karlılığı oranı, fiyat kazanç (F/K) oranı, net kar, büyüme hızı, öz sermaye artış hızı, işlem görme oranı, piyasa değeri defter değeri oranı (PD/DD) olarak ele almıştır. Albeni ve Demir (2005) yaptıkları çalışmada hisse senedi fiyatını etkileyen makro değişkenlerin fiyatlar genel düzeyi (enflasyon), kamu harcamalarındaki farklılıklar, gayri safi milli hasıladaki değişmeler, döviz kurundaki değişmeler, altın fiyatları, faiz oranları, uluslararası portföy yatırımları, para arzındaki değişmeler (emisyon) ve özelleştirme uygulamaları olduğunu ortaya koymuşlardır.

Hisse senedi değerlerinin yönleri ile ilgili bir başka yöntem olarak da yukarıda bahsedilen reel verilerden bağımsız olarak hisse senedi ve borsa grafiklerinin analizini ekleyebiliriz. Bu teknik, genel olarak yatırımcıların tercih akışlarının değişimlerine, olabilecek en hızlı şekilde tepki vererek, yön değişimin başlama anından bitiş anına kadar olan hareketi, en efektif şekilde kullanma prensibine dayanır. Bu yaklaşımın, yukarıda bahsedilen reel yöntemlere göre daha başarılı sonuçlar verdiği de söylenebilir. Orçun (2010), 4 farklı grafik üzerinde uygulanabilecek teknik analiz formasyonlarını incelemiştir.

1. Çubuk grafik
2. Çizgi grafik

3. Mum grafik
4. Nokta ve şekil grafikleri.

Bu grafik çeşitleri üzerinde uygulanabilecek teknik analiz formasyonlarını da omuz – baş – omuz formasyonu, ters omuz – baş – omuz formasyonu, çift tepe formasyonu, çift dip formasyonu, üçgen formasyonları, takoz formasyonları, dikdörtgen formasyonları, elmas formasyonu, kama formasyonları ve boşluklar olarak belirleyip tek tek ele almıştır. Tüm bu formasyonların temel amacı, yukarıda da bahsedildiği gibi, kullanıcı tercih değişimlerinin mümkün olan en kısa sürede saptanması ve ona göre tepki verilmesidir.

Bizim bu çalışmamızın amacını da yatırımcı tercihlerindeki değişimlerin saptanmasının bir adım daha önceden, yani henüz yatırımcı yatırımını yapmadan, hisse senedi yukarı veya aşağı yönlü hareketine başlamadan önce, henüz daha sosyal medyada konuşulurken yapılıp yapılamayacağı veya hangi doğrulukla yapılabileceğinin tespiti olarak ifade edebiliriz. Böyle bir tahmin yapılabiliyorsa, bunu en başarılı şekilde yapabilecek tekniğin belirlenmesi de yine bu çalışmamızın amaçlarından bir tanesi olarak eklenebilir.

Analiz verilerini elde etmek için, günümüzde en çok kullanılan sosyal medya platformlarından biri olan *Twitter* tercih edilmiştir. Ayrıca, bu çıkarımı yaparken, atılan tivitlerin pozitif mi, negatif mi yoksa nötr mü olduğunun sınıflandırılması işinde 6 farklı sınıflandırma algoritmasının başarım sonuçlarını karşılaştırarak, kullandığımız yöntemde hangi algoritmanın ne kadar başarılı olduğu ortaya konulmuştur.

3.2. Tezin Önemi

Yatırımcı tercihleri, sosyal, ekonomik, siyasi, Türk ve yabancı merkez bankalarının ve hükümetlerinin önlemleri, kuraklık, hükümet güven endeksi, seçimler v.b. birçok parametreye bağlı olarak değişiklik gösterse de yapılan araştırmalara göre Türkiye'de yatırımcıların %25 – %30'u, Amerikada ise %34'ü hisse senetlerini tercih etmektedir. Bu kadar popüler olan bir yatırım aracı için de sadece şirketin karlılığını değil, bunun yanında işlem hacmi, hükümet güven endeksi, faiz artırma veya düşürme kararları, mevsimsel etkiler gibi birçok parametreyi de göz önünde bulundurarak, daha sağlıklı tahminler yapmak için bugüne kadar sayısız çalışmalar yapılmış, pek çok farklı yöntem geliştirilmiştir.

Bu çalışmamızda, son dönemde popülaritesi giderek artan ve birbirleriyle sarmal bir ilişkiye sahip olan bu iki konuyu ortak bir şekilde ele alarak, daha etkin sonuçlar elde edilmesi düşünülmektedir. Bu çalışmamızda, hem yapay zekâ teknikleri kullanılarak hisse senetleri fiyatlandırmaları üzerine tahminde bulunma konusunda, hem de sosyal medya mesajları üzerinde metin madenciliği yapılarak, toplumu ilgilendiren pek çok konu üzerinde önsezi geliştirme konularında, farklı bir bakış açısı ortaya konulması amaçlanmaktadır. Benzer konularda daha sonra çalışma yapacak olan arkadaşlara bir kaynak oluşturması veya bir fikir vermesi ümit edilmektedir.

4. VERİ SETİ OLUŞTURULMASI

Tezimize konu olan araştırma ve uygulamanın yapılmasında öncelikli olarak üzerinde çalışma yapılacak olan veri setinin hazırlanması gerekir. Bu veri seti, programın işleyeceği, sınıfları tespit etmek için girdi olarak kullanacağı twitter verileri ve tivitlerin hisse senedi yön tahmininde pozitif, negatif ve nötr sınıfları temsil eden, Dow Jones borsasının ilgili hisselerine ait günlük en yüksek, en düşük ve kapanış değerlerini içeren yön tahmini test verileri olmak üzere iki tip veriden oluşmaktadır.

Çalışmamız, 1 Nisan 2019 tarihinden 31 Mayıs 2019 tarihine kadar olan iki aylık periyodu kapsamaktadır.

4.1. Twitter Verileri

Sistemin eğitilmesinde ve test edilmesinde girdi olarak kullanılacak twitter mesajlarıdır.

4.1.1. Twitter verilerinin elde edilmesi

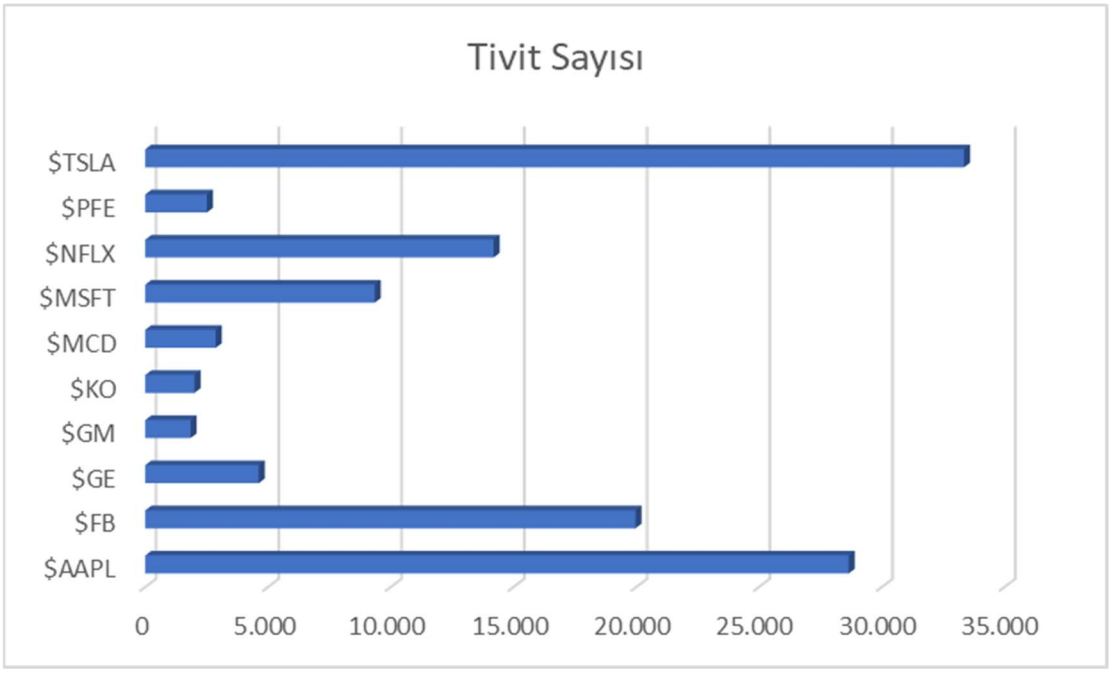
Twitter firması, yazılımcıların rahatça twitter verilerine ulaşabilmeleri, tivit atabilmeleri, twitter verileri içerisinde arama yapabilmeleri v.b. konularda zorluk çekmemeleri için Twitter API olarak bilinen ve pek çok platform üzerinde kullanılabilen bir API hazırlayarak yazılımcıların kullanımına sunmuştur. Ancak kendi sistemini çok fazla yormamak adına da birtakım kullanım kısıtları getirmiştir. Örneğin bu API kullanılarak twitter verileri içerisinde yapılan aramalarda en fazla 1000 tivit dönmektedir. Bu da bizim çalışmamız için yeterli değildir. Bunun için araştırmamızda, twitter verilerinin elde edilmesinde bu API kullanılmamış, bunun yerine twitter web ara yüzü kullanılarak arama yapılmıştır. Bu aramalar Apple ($AAPL), Facebook ($FB), General Electric ($GE), General Motors ($GM), The Coca-Cola Company ($KO), McDonald's ($MCD), Microsoft Corporation ($MSFT), Netflix ($NFLX), Pfizer Corporation ($PFE), Tesla Motors ($TSLA) olmak üzere toplam 10 tane hisse senedi etiketi için iki aylık periyodu kapsayacak şekilce ve her biri için günlük olarak ayrılarak yapılmıştır. 1 Nisan 2019 tarihli borsa değerlerinin tahminini yapmak için 31 Mart 2019 tarihli tivitler kullanılmış, 31 Mayıs 2019 tarihli borsa değerlerinin tahminini yapmak için 30 Mayıs 2019 tarihli tivitler kullanılmıştır. Bunun için de 31 Mart 2019 ile 30 Mayıs 2019 tarihleri

arasındaki twitter verileri alınmıştır. Bu verilerin alınması için, twitter web arayüzünü kullanarak arama yapan bir ara program yazılmıştır. Çekilen verilerdeki hisse senedi başına tivit sayısını gösteren tablo Çizelge 4.1'de sunulmuştur.

Çizelge 4.1. Hisse senetlerine göre atılan tivit sayıları

Hisse Senedi	Tivit Sayısı
$AAPL	28.649
$FB	19.969
$GE	4.619
$GM	1.846
$KO	2.005
$MCD	2.868
$MSFT	9.344
$NFLX	14.187
$PFE	2.508
$TSLA	33.348

Şekil 4.1'de hisse senetlerine göre tivit sayıları grafiksel olarak gösterilmiştir.



Şekil 4.1. Hisse senetlerine göre atılan tivit sayıları

4.1.2. Twitter verilerinin gruplanması

Elde edilen bu tivitler, farklı şirketlerin farklı günlerdeki değerlerinin analizinde kullanılacağından, her bir hisse senedi için atılmış günlük tivitleri içerecek şekilde dosyalara ayrılmıştır. Böylece 10 hisse senedi için toplam 61 günlük verileri içeren 610 adet dosyaya ayrılmıştır.

Örneğin Apple firması için 31 Mart 2019 tarihinde atılan tivitler $AAPL-2019-3-31.txt isimli dosyada, Tesla Motors firmasi için 20 Nisan 2019 tarihinde atılan tivitler $TSLA-2019-4-20.txt isimli dosyada, Facebook firması için 15 Mayıs 2019 tarihinde atılan tivitler $FB-2019-5-15.txt isimli dosyada olacak şekilde gruplanmıştır.

Gruplandırılan bu tivitler, el ile, okunarak tek tek etiketlenmiştir. Etiketleme esnasında atılan tivitlerinin tamamının düzenli cümlelerden oluşmadığı, çoğunluğunun herhangi bir duygu ifade etmediği, çok fazla reklam içerdiği gözlemlenmiştir. Bahsedilen bu satırların tamamı NÖTR olarak etiketlenmiştir. Bunun dışında atılan tivitte, ilgili hisse senedi için olumlu bir kanaat bildiriliyorsa POZİTİF, olumsuz bir kanaat bildiriliyorsa NEGATİF olarak etiketlenmiştir. Anlamsız tivitlerinin sayısının çokluğu ve bu tivitlerin tamamının nötr olarak etiketlenmesinden dolayı, sonuç kısmında bahsedilecek, ortaya konulacak olan sonuçlarda, sistemin tahmin başarısı ölçülürken, atılan tivitin, bir gün sonraki hisse senedi hareketi ile aynı değeri taşımasına bakılmamış, sonuçlar iki sınıfa indirgenerek nötr olan tivitlerin başarıyı olumsuz etkilemesinin önüne geçilmiştir. Yani sistemin başarısının ölçümünde, pozitif olarak etiketlenmiş bir tivitin ertesi gününde hisse senedinin negatif yönlü bir hareket yapmamış olması başarı olarak kabul edilmiş, aynı şekilde negatif olarak işaretlenmiş olan bir tivitin ertesi gününde hisse senedinin pozitif yönlü bir harekete yapmamış olması başarı olarak kabul edilmiştir.

4.2. Borsa Verileri

Hisse senetlerinin yön tahmininin başarı değerinin ölçülmesinde ve test edilmesinde kullanılacak, Apple, Facebook, General Electric, General Motors, The Coca-Cola Company, McDonald’s, Microsoft Corporation, Netflix, Pfizer Corporation ve Tesla Motors olmak üzere toplam on adet şirketin 1 Nisan 2019 – 31 Mayıs 2019 tarihleri arasındaki en yüksek, en düşük ve kapanış değerlerini içeren arşiv verileridir.

4.2.1. Borsa verilerinin elde edilmesi

Borsa verileri, www.eoddata.com internet sitesi ara yüzü kullanılarak günlük olarak elde edilmiştir. Bu siteden arşiv verilerine erişmek için üyelik gerekmektedir. Site içerisinden ücretsiz olarak yapılacak üyelik bilgileri ile sisteme giriş yaptıktan sonra "Download" linkine tıklanarak arşiv verilerinin bulunduğu bölüme geçilebilir. Buradan da her bir tarih için tek tek arşiv verileri indirilebilir.

4.2.2. Borsa verilerinin temizlenmesi

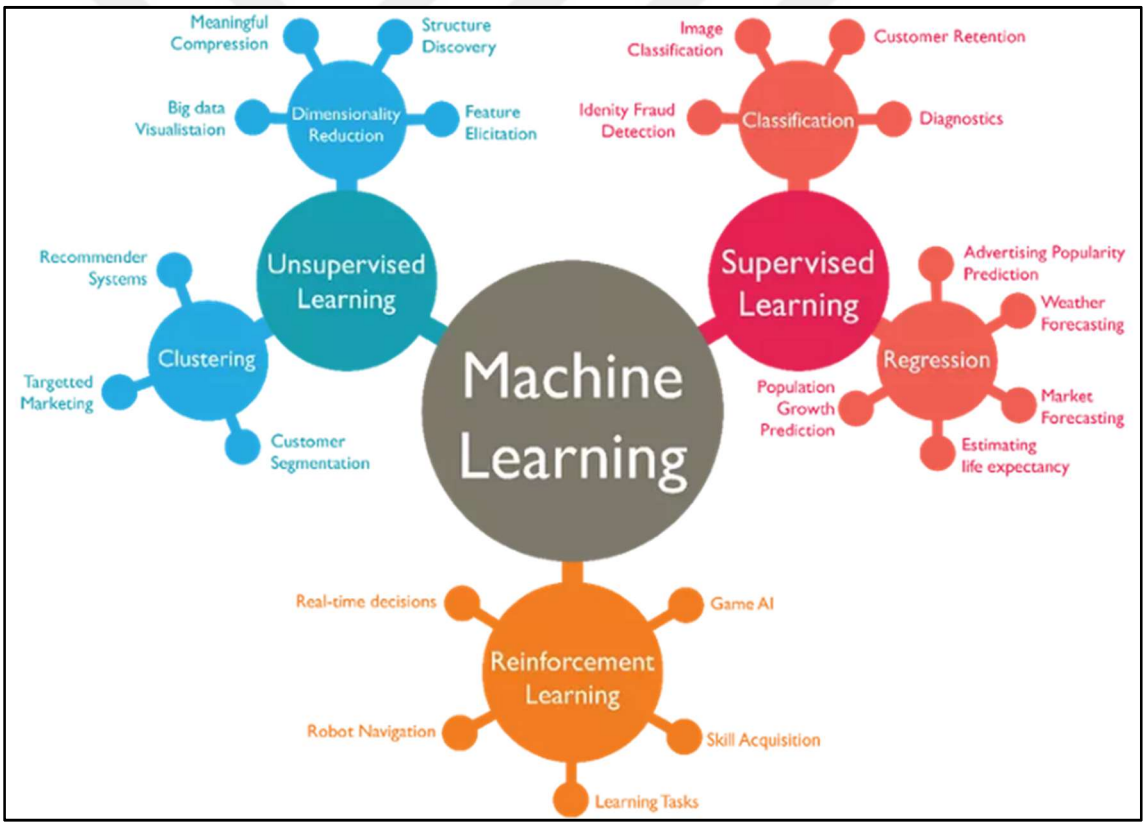
Elde edilen bu arşiv dosyaları, Dow Jones borsasında işlem gören tüm şirketlere ait günlük verileri içermektedir. Bu da gereksiz yere fazladan pek çok işlem yapacağımız anlamına gelmektedir. Onun için bu verileri kullanmadan önce, gereksiz olan kısımlarını silmemiz gerekmektedir. Her bir dosyanın içerisinde sadece bizim ilgilendiğimiz on şirketin verileri kalacak şekilde diğer veriler silinmelidir. Böylece elimizde sadece bizim ilgilendiğimiz hisse senetlerinin günlük verilerini içeren, 1 Nisan 2019 – 31 Mayıs 2019 tarihleri arasını kapsayan 61 adet dosya olması gerekir. Ancak Dow Jones borsası Cumartesi ve Pazar günleri kapalı olduğundan toplam 45 dosya elde ettik.

Cumartesi ve Pazar günleri Dow Jones borsası kapalı olduğundan, bu günlerde atılan tivitler aslında Pazartesi gününün tahmininde kullanılacağı düşünülmüştür. Bu düşünceyle de Cuma, Cumartesi ve Pazar günü atılan tivitlerin tahmin başarısını ölçmede Pazartesi gününe ait borsa değerleri, Pazartesi günü atılan tivitlerin tahmin başarısını ölçmede Salı gününe ait borsa verileri, Salı günü atılan tivitlerin tahmin başarısını ölçmede Çarşamba gününe ait borsa verileri şeklinde tahmin başarısı değerlemesi yapılmıştır. Ayrıca Dow Jones borsası yerel saat ile sabah 09:00'da işlem görmeye başladığından dolayı, ilgili güne ait borsa verileri, bir gün önce saat 09:00 ile o gün saat 09:00 arasında atılan tivitlerin tahmin başarısını ölçmede kullanılmıştır.

5. METİN SINIFLANDIRMA

5.1. Makine öğrenmesi

Machine Learning (ML) kavramının Türkçe karşılığıdır. Makine öğrenmesi, günümüzde oldukça popüler olan makine algılaması, bilgisayarlı görme, doğal dil işleme, sözdizimsel örüntü tanıma, arama motorları, tıbbi tanı, biyoinformatik, beyin-makine arayüzleri ve kiminformatik, kredi kartı dolandırıcılığı denetimi, borsa çözümlemesi, DNA dizilerinin sınıflandırılması, konuşma ve elyazısı tanıma, bilgisayarlı görmede nesne tanıma, oyun oynama, yazılım mühendisliği, uyarlamalı web siteleri, fen bilimleri ve robot gezisi gibi pek çok alanda kullanılan, yapay zekanın bir alt koludur. Makine öğrenmesi kullanım alanları Şekil 5.1'te gösterilmiştir.



Şekil 5.1. Makine öğrenmesi uygulama alanları (Dayıbaşı, 2017)

Temel olarak, otomatik öğrenme ve geliştirme ilkesine dayanır. Makine öğrenmesi, standartlaşmış çeşitli algoritmalar ve yöntemler ile veride bazı kalıpları arar ve bu kalıplara karşılık gelen etiketlere bakarak önce öğrenir, daha sonra (öğrendiklerine

benzer durumla karşılaştığında) deneyimlerinden yararlanarak çıkarım yapabilen sistemler geliştirmeye imkân sağlar.

Bu imkânı, çeşitli matematiksel ve istatistiksel yöntemlerin kullanıldığı birçok algoritma ile sağlamaktadır. Bu yöntem ve algoritmaların bir veya birkaçı bir arada kullanılarak model(ler) oluşturulur ve bu model(ler), tahmin edilmesi istenilen şeyi, en verimli, en kesin en hızlı biçimde tahmin etmeyi amaçlar (Gülcan, 2019). Şekil 5.2'de makine öğrenmesi için farklı alanlarda kullanılmak üzere ortaya atılmış ve standartlaşmış bazı algoritmalar gösterilmiştir.

Şekil 5.2. Makine öğrenmesi için geliştirilmiş algoritmalar (Dayıbaşı, 2017)

Makine öğrenmesi temelli yöntemlerin birçoğunda verilerin bir kısmı önceden sınıflandırılmıştır. Bu sınıflandırılmış veriler sistemin eğitilmesinde kullanılır. Önceden etiketlenmiş ve sistemin eğitilmesinde kullanılan bu verilere eğitim verileri denir. Burada eğitilen sistemin test edilmesinde kullanılan, etiketlenmemiş verilere ise test verileri denir. Bu yöntemle sınıflandırma yapabilmek için daha önceden geliştirilmiş birtakım sınıflandırma algoritmaları vardır. İlerleyen konularda, bu çalışmamızda karşılaştırma amaçlı kullanacağımız destek vektör makinesi (SVM), genetik algoritmalar (GA), yapay

sinir ağları (ANN), Naïve Bayes (NB), karar ağaçları (DT), k-en yakın komşu (k-NN) ve rastgele orman (RF) algoritmaları detaylı şekilde anlatılacaktır.

Metin sınıflandırmada makine öğrenmesi yöntemleri ile özellik seçimi, ilk olarak Pang ve ark. (2002) tarafından sinema filmlerine ait yorumları, olumlu ve olumsuz olarak sınıflandırmak için kullanılmış bir yöntemdir.

ML, sözlük tabanlı yöntemlerin aksine, cümle içerisinde geçen kelimelerin semantik ve sentaktik (kelimelerin anlamları ve cümle içerisindeki görevleri) özellikleri ile ilgilenmez, onları birer büyüklük olarak görür ve bu büyüklüklerden oluşan vektörler üzerinde işlem yapar. ML yöntemleri, sözlük tabanlı yöntemlere göre daha başarılı sonuçlar verebilmektedir. Özellikle sözlük tabanlı yöntemlerin daha yetersiz kaldığı dolaylı kurulmuş cümlelerde çok daha başarılı sonuçlar verebilmektedir. Bunun yanında sistemin iyi sonuç verebilmesi için daha önceden etiketlenmiş bir eğitim veri setine çoğu zaman ihtiyaç vardır. Ayrıca bu eğitim veri seti, çalışmanın yapıldığı alana özgü olmalıdır. Başka alanlarda çok fazla işe yaramaz.

5.1.1. Terminoloji

Gözlemler (Samples, Rows, Instances, Observations, Records, Examples): Öğrenmek için kullanılan veri parçasına denir. Örneğin; Kredi Kartı Bilgileri

Özellikler (Variables, Features, Attributes, Fields, Columns, Dimensions): Gözlemi temsil eden verinin değerleridir. Örneğin; Kart üzerindeki isim, kartın numarası, geçerlilik tarihi, CVV kodu.

Etiketler (Labels): Gözlemler sonuçu ortaya çıkan kategoriler. Örn: fraud (sahtekarlık), not-fraud

Eğitim Verisi (Training Data): Gözlemlerden oluşan diziler, algoritmaya öğrenmesi için gönderilir. Algoritma bu veri dizisinden çıkarımlarda bulunur ve bir model oluşturur. *True* ya da *False* diye etiketler oluşturur.

Test Verisi (Test Data): Elde edilen modelin gerçek değerlere ne kadar yakın olduğunu saptamak için kullanılan test verileridir. Eğitim sırasında algoritma test verilerini görmez. Eğitim verilerinde ürettiği çıkarımı test verileri üzerinde kullanır. Aslında burada sistemin ne kadar başarı sağladığı saptanır. (Akın, 2017)

5.1.2. Öğrenme türleri

Gözetimli (Supervised) Öğrenme: Eğitim verileri algoritmaya gönderildikten sonra etiketlenir. Böylece hangi girdi için hangi çıktıyı alacağımızı öğrenmiş oluruz. Bu öğrenme türüne gözetimli öğrenme (*Supervised Learning*) diyoruz.

Mesela son zamanlarda insan olup olmadığınızı anlamak için resimdeki binaları vs. işaretlememiz isteniyor. Eğer böyle bir sistem tasarlamak istiyorsak ilk önce farklı bina tiplerini bulup bunları bina şeklinde işaretlememiz aksi durumlarda da tam tersi olarak işaretlememiz gerekiyor. Bu da gözetimli (*supervised*) öğrenme şekline bir örnektir. Bu öğrenme türü de yaptığı işe göre kendi arasında *sınıflandırma (classification)* ve *regresyon (regression)* olmak üzere ikiye ayrılır.

- *Sınıflandırma (Classification):* Sınıflandırmada amaç kategoriyi tahmin etmektir. Örneğin: Ferrari pahalı araçlar sınıfına aitken, Renault clio ucuz araçlar sınıfındadır.
- *Regresyon (Regression):* Regresyonda amaç sayısal bir değeri tahmin etmektir. Örneğin Ferrari 430 Spider 2010 model 15.000 km'deki bir aracın fiyatı 250.000$ civarı olmalıdır.

Gözetimsiz (Unsupervised) öğrenme: Etiketsiz verilerle algoritmanın öğrenmesidir. Algoritma veriler arasında kendi kendine ilişki kurar ve bilinmeyen yapıları keşfetmesi beklenir. Yaptığı işe göre ikiye ayrılır:

- *Kümeleme (Clustering):* Benzer özelliklerdeki verileri bir grupta toplar. Örneğin; yazılım mühendislerinin *Senior* ve *Junior* olarak iki ayrı grupta toplanması. Homojen bir dağılım yapılır.
- *İlişki Analizi (Association Analysis):* Öğeler arasındaki ilişkileri yakalamak için kurallar bulunur. Örneğin, bir e-ticaret sitesinden alışveriş yaparken, "*bu ürünü alanlar bunlara da baktı*" şeklinde ilişkilendirme yapılması. (Akın, 2017)

5.1.3. Özellik seçimi

Makine öğrenmesi tekniklerini başarılı bir şekilde uygulayabilmek için, özellik vektörünü başarılı bir şekilde oluşturmak gerekir. Bu işlem, sınıflandırma performansını doğrudan etkiler. Başarıyı arttırabilmek için özellikleri en doğru şekilde tespit etmek ve

kullanmak gerekir. Özellik seçiminde kullanılan, birtakım standartlaşmış yöntemler vardır. Bu yöntemleri şöyle sıralayabiliriz:

Kelime Torbası (Bag of Words): Özellik seçiminde en çok kullanılan yöntemdir. Kelimelerin veya kelime gruplarının, metin içerisinde geçip geçmediğine bakılır. Her bir kelime metin içerisinde tek tek aranabildiği gibi, ikili gruplar veya üçlü ve daha fazla gruplar şeklinde de aranabilir. Her bir kelime ayrı ayrı aranıyorsa buna *unigram*, ikişerli gruplar halinde aranıyorsa buna *bigram*, n tane kelime bir arada aranıyorsa buna *n-gram* modeli denir.

Kelimelerin metin içerisinde kaç defa geçtiklerine bakılabildiği gibi, kaç defa geçtiği ile ilgilenilmeden, sadece geçip geçmediği de kontrol edilebilir. Kaç defa geçtiği sayılıyorsa buna *terim frekans ağırlığı* yöntemi ile özellik seçimi denir. Kaç defa geçtiği sayılmıyorsa, sadece geçip geçmediğine bakılıyorsa, buna *ikili* özellik seçimi denir.

Terim frekansı – ters doküman frekansı (Term Frequency – Inverse Document Frequency): *TF-IDF* yöntemi olarak da bilinir. *Term Frequency – Inverse Document Frequency* kelimelerinin baş harflerinden oluşur. Her bir kelimenin ayırt edicilik özelliğini tespit etmek için kullanılır. Terim frekansı, her bir kelimenin doküman içerisinde kullanılma sayısının, doküman içerisinde en fazla kullanılan kelimenin kullanılma sayısına oranını ifade eder. Terim frekansı (TF), formül (5.1)'de gösterildiği şekilde hesaplanır.

TF = (terim tekrarlama sayısı / en sık kullanılan terimin tekrarlama sayısı) (5.1)

Örneğin bizim ilgilendiğimiz kelime, doküman içerisinde 5 kez geçiyorsa ve doküman içerisinde en sık kullanılan kelime toplamda 40 kez geçiyorsa, terim frekansımız 5/40 yani 0,125 olarak hesaplanır. IDF değeri ise toplam doküman sayısının, terimi içeren toplam doküman sayısına oranının logaritmasıdır. Ters doküman frekansı da (IDF), formül (5.2)'de gösterildiği şekilde hesaplanır.

IDF = log (Toplam Doküman Sayısı / Terimi İçeren Doküman Sayısı) (5.2)

Şeklinde ifade edilebilir. Örneğin bizim ilgilendiğimiz kelime toplam 80 tane dokümanın 8'inde geçiyorsa

IDF = log (80 / 8) = log10 = 1

olarak bulunur.

Sözcük türü (Part of speech): Cümlenin ögelerine ayrılması ve her bir kelimenin türüne göre işleme tabi tutulması esasına dayanır. Metin madenciliği ve doğal dil işleme çalışmalarında sıkça kullanılan önemli bir tekniktir. İsim, fiil, sıfat ve zarf türündeki kelimelerle daha çok ilgilenilir. Özellik vektörünü oluşturmada kullanılır.

Nokta tabanlı karşılıklı bilgi - NKB (Pointwise Mutual Information): İki terim arasındaki anlamsal yakınlığı ölçmek için ortaya atılmış bir skorlama yöntemidir. İlk kez Church ve Hanks (1990) tarafından kullanılmıştır. Makine öğrenmesi, doğal dil işleme, metin işleme, anlamsal çıkarım çalışmalarında sıklıkla kullanılmaktadır. NKB'nin formülü (5.3)'teki gibidir:

$$NKB\ (x,y) = log\ (p\ (x,y) / (p\ (x) * p\ (y))) \quad (5.3)$$

Bu formülde (5.3), *p(x)*, *x*'in metin içerisinde tek başına yer alma sayısı, *p(y)*, metin içerisinde *y*'nin tek başına yer alma sayısı, *p(x,y)* ise metin içerisinde *x* ve *y*'nin birlikte yer alma sayısını ifade etmektedir. Terimlerin birlikte yer alma sayıları genellikle eldeki tüm dokümanlar üzerinden hesaplanmaktadır. Bazı çalışmalarda ise bu sayılar web aramalarından elde edilmektedir. NKB skoru ne kadar büyükse iki terimin anlamsal yakınlığı da o kadar yüksektir. (Özyurt ve Akcayol, 2018)

5.2. Sınıflandırma Algoritmaları

Makine öğrenmesi başlığı altında Şekil 5.2 (Dayıbaşı, 2017)'de gösterilen algoritmalar, farklı farklı problemlerin çözümlerinde kullanılan standartlaşmış algoritmalardır. Bu, algoritmanın sadece o problemin çözümünde kullanılabileceği anlamına gelmez. Başka bir problem, algoritmanın işleyişine uygun şekilde modellendiği zaman, farklı problemlerin çözümlerinde de kullanılabilir. Bu sebeple, bizim bu başlık altında anlatacağımız algoritmalar, genel olarak sınıflandırma amaçlı olarak önerilmiş algoritmalardır. Bunun yanında burada anlatılmayan başka algoritmalar da sınıflandırma amacıyla kullanılabilir.

Sınıflandırma problemi için geliştirilen algoritmalar, adından da anlaşılacağı üzere verileri belli özelliklerine göre sınıflandırırlar. Sınıflandırma yapısal (*structure*) veya yapısal olmayan (*unstructure*) veriler üzerinde yapılabilir. Eğer sistem, hangi verinin, hangi koşullarda, hangi sınıfa ait olacağı bilgisi ile sınıflandırılarak eğitilirse,

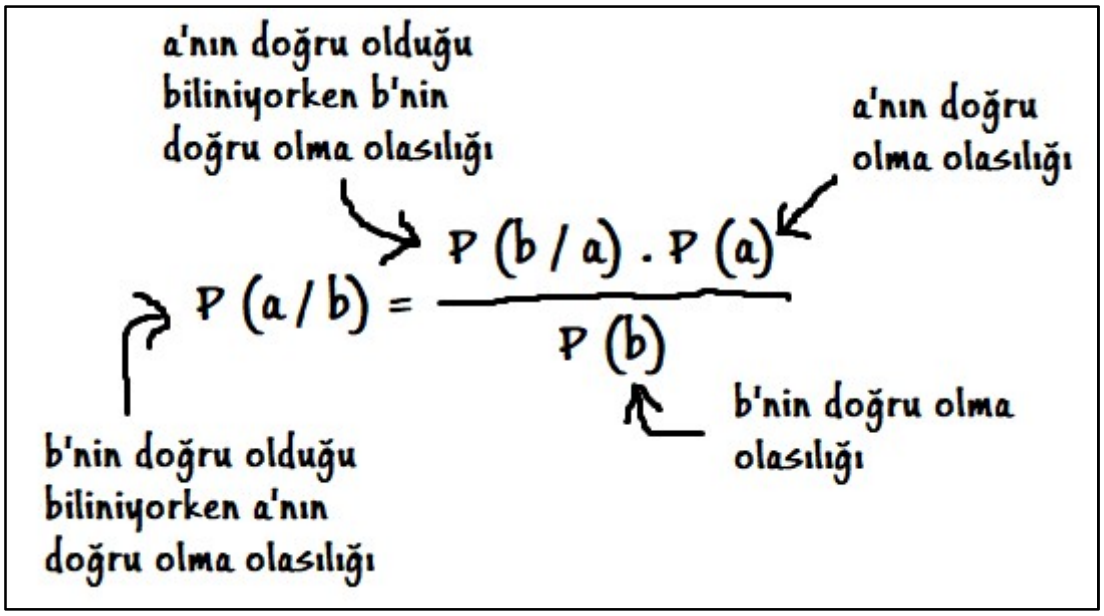
yeni veri setindeki veriyi de öğrendiklerine benzer biçimde sınıflandırabilir (Gülcan, 2019).

5.2.1. Naïve bayes

1812 yılında Thomas Bayes tarafından bulunan koşullu olasılık hesaplama formülüdür. Bayes teoremi, olasılık kuramı içinde incelenen önemli bir konudur. Makine öğrenmesinde bayes sınıflandırıcılar, özellikler arasında güçlü bir bağımsızlık varsayımı içeren basit bir olasılıksal sınıflandırma ailesidir.

Naïve Bayes sınıflandırıcılar üzerine 1960'lı yıllarından beri çalışılmaktadır. Aslında bu konuyla ilgili olmamasına rağmen, 1960'ların ilk yıllarında metin işleme ile uğraşan topluluklara duyuruldu ve o yıllardan beri metin kategorizasyonu için temel metotlardan biri olarak kullanılmaktadır. Bunun sebebi biraz da metin kategorizasyonunun halen temel bir problem olarak çözülmeye devam edilmesi olabilir (mailler spam mı, değil mi? Yorum politik mi spor ile mi ilgili? v.s.). İyi bir ön-işleme yapıldığı taktirde SVM gibi çok daha gelişmiş algoritmalar ile yarışabilecek seviyede sonuçlar verebilir. Ayrıca otomatik tıbbi tanı koyma problemlerinin çözümünde de sıklıkla kullanılır (wikipedia, 2019).

5.2.1.1. Naïve bayesin yapısı



Şekil 5.3. Bayes formülü

Naive Bayes sınıflandırıcısının temeli Bayes teoremine dayanır. Bayes teoreminin formülü, Şekil 5.3'te açıklamalarıyla gösterilmiştir.

Bayes sınıflandırma algoritması, tembel (*lazy*) bir öğrenme algoritmasıdır. Aynı zamanda dengesiz veri kümelerinde de çalışabilir. Algoritmanın çalışma şekli şöyledir: Bir eleman için her durumun olasılığı hesaplanır ve olasılık değeri en yüksek olana göre sınıflandırılır. Az bir eğitim verisiyle çok yüksek başarı performanslarına ulaşılabilir. Test kümesindeki bir kelimenin eğitim kümesinde gözlemlenemeyen bir değeri varsa olasılık değeri olarak 0 verir. Bu tahmin yapılamayacağı anlamına gelir. Bu durum genellikle Sıfır Frekans (*Zero Frequency*) adıyla bilinir. Bu durumu gidermek için düzeltme teknikleri kullanılabilir. En basit düzeltme tekniklerinden biri *Laplace* tahmini olarak bilinir.

Kullanım alanlarına örnek olarak gerçek zamanlı tahmin, çok sınıflı tahmin, metin sınıflandırması, spam filtreleme, duyarlılık analizi, tıbbi tanı koyma ve öneri sistemleri verilebilir. (Hatipoğlu, 2018)

Çizelge 5.1'de havanın durumuna göre tenis oynama veya oynamama verileri gösterilmektedir.

Çizelge 5.1. Naive bayes sınıflandırma örneği (Yalçın, 2019)

Gün	Hava Durumu	Sıcaklık	Nem	Rüzgar	Tenis Oynama
1	Güneşli	Sıcak	Yüksek	Zayıf	Hayır
2	Güneşli	Sıcak	Yüksek	Şiddetli	Hayır
3	Bulutlu	Sıcak	Yüksek	Zayıf	Evet
4	Yağmurlu	Ilık	Yüksek	Zayıf	Evet
5	Yağmurlu	Serin	Normal	Zayıf	Evet
6	Yağmurlu	Serin	Normal	Şiddetli	Hayır
7	Bulutlu	Serin	Normal	Şiddetli	Evet
8	Güneşli	Ilık	Yüksek	Zayıf	Hayır
9	Güneşli	Serin	Normal	Zayıf	Evet
10	Yağmurlu	Ilık	Normal	Zayıf	Evet
11	Güneşli	Ilık	Normal	Şiddetli	Evet
12	Bulutlu	Ilık	Yüksek	Şiddetli	Evet
13	Bulutlu	Sıcak	Normal	Zayıf	Evet
14	Yağmurlu	Ilık	Yüksek	Şiddetli	Hayır

Çizelge 5.1'deki verilerde özellikleri, tenis oynama veya oynamama olasılıklarına göre gruplandırırsak:

Çizelge 5.2. N.B. – Verilerin gruplandırılması (Yalçın, 2019)

Hava Durumu	
P(Güneşli | Evet) = 2/5	P(Güneşli | Hayır) = 3/5
P(Bulutlu | Evet) = 4/4	P(Bulutlu | Hayır) = 0/4
P(Yağmurlu | Evet) = 3/5	P(Yağmurlu | Hayır) = 2/5

Sıcaklık		
P(Sıcak | Evet) = 2/4	P(Sıcak | Hayır) = 2/4	P(Evet) = 9 / 14
P(Ilık | Evet) = 4/6	P(Ilık | Hayır) = 2/6	P(Hayır) = 5 / 14
P(Serin | Evet) = 3/4	P(Serin | Hayır) = 1/4	

Nem Oranı	
P(Yüksek | Evet) = 3/7	P(Yüksek | Hayır) = 4/7
P(Normal | Evet) = 6/7	P(Normal | Hayır) = 1/7

Rüzgar	
P(Şiddetli | Evet) = 3/6	P(Şiddetli | Hayır) = 3/6
P(Zayıf | Evet) = 6/8	P(Zayıf | Hayır) = 2/8

Çizelge 5.2'deki gibi bir sınıflandırma yaptıktan sonra yeni gelen verilerin sınıfları, buradaki olasılıklara göre tahmin edilebilir.

Örneğin yeni değerimiz x = <*yağmurlu, sıcak, yüksek, zayıf*> şeklinde olsun.*P(Evet | x)* değerini hesaplayalım.

$$P(Evet \mid x) = P(yağmurlu \mid evet) \cdot P(sıcak \mid evet) \cdot P(yüksek \mid evet) \cdot P(zayıf \mid evet)$$
$$= 3/5 \cdot 2/4 \cdot 3/7 \cdot 6/8 \cdot 9/14$$
$$= 0{,}062$$

P(Hayır | x) değerini hesaplayalım.

$$P(Hayır \mid x) = P(yağmurlu \mid Hayır).P(sıcak \mid Hayır).P(yüksek \mid Hayır).P(zayıf \mid Hayır)$$
$$= 2/5 \cdot 2/4 \cdot 4/7 \cdot 2/8 \cdot 5/14$$
$$= 0{,}01$$

Olarak bulunur. *P(Evet | x) > P(Hayır | x)* olduğundan ***x** örneğinin sınıfı **evet*** olarak öngörülür. (Yalçın, 2019)

5.2.1.2. Naïve bayesin avantajları

1. Kolay uygulanabilir
2. Üstün performans gösterir
3. Az eğitim verisiyle bile kabul edilebilir sonuçlar elde edilebilir. (Yalçın, 2019)

5.2.1.3. Naïve bayesin dezavantajları

1. Varsayım: Sınıf bilgisi verildiğinde nitelikler bağımsız olması
2. Gerçek hayatta değişkenler birbirine bağımlı olması
3. Değişkenler arası ilişkilerin modellenememesi
4. Test verisinin işlem zamanı uzundur. (Yalçın, 2019)

5.2.1.4. Naïve bayesin uygulama alanları

1. Metin sınıflandırma
2. Konuşmacı tanıma sistemleri
3. Şifre kontrolü uygulamaları
4. Orta veya geniş eğitim kümesinin mevcut olduğu sınıflandırma problemlerinde
5. Örnekleri tanımlayan niteliklerin sınıflandırmadan bağımsız olarak verildiği sınıflandırma problemlerinde kullanılır. (Yalçın, 2019)

5.2.2. Destek vektör makinesi

Değişkenler arasındaki örüntülerin bilinmediği veri setlerindeki sınıflandırma problemleri için önerilmiş bir makine öğrenmesi yöntemidir. Sınıflama, regresyon ve aykırı değer belirleme için kullanılabilen gözetimli (*supervised*) bir öğrenme yöntemidir. Eğitim verisinde öğrenme yaparak, test verileri üzerinde doğru tahmin yapmaya ve

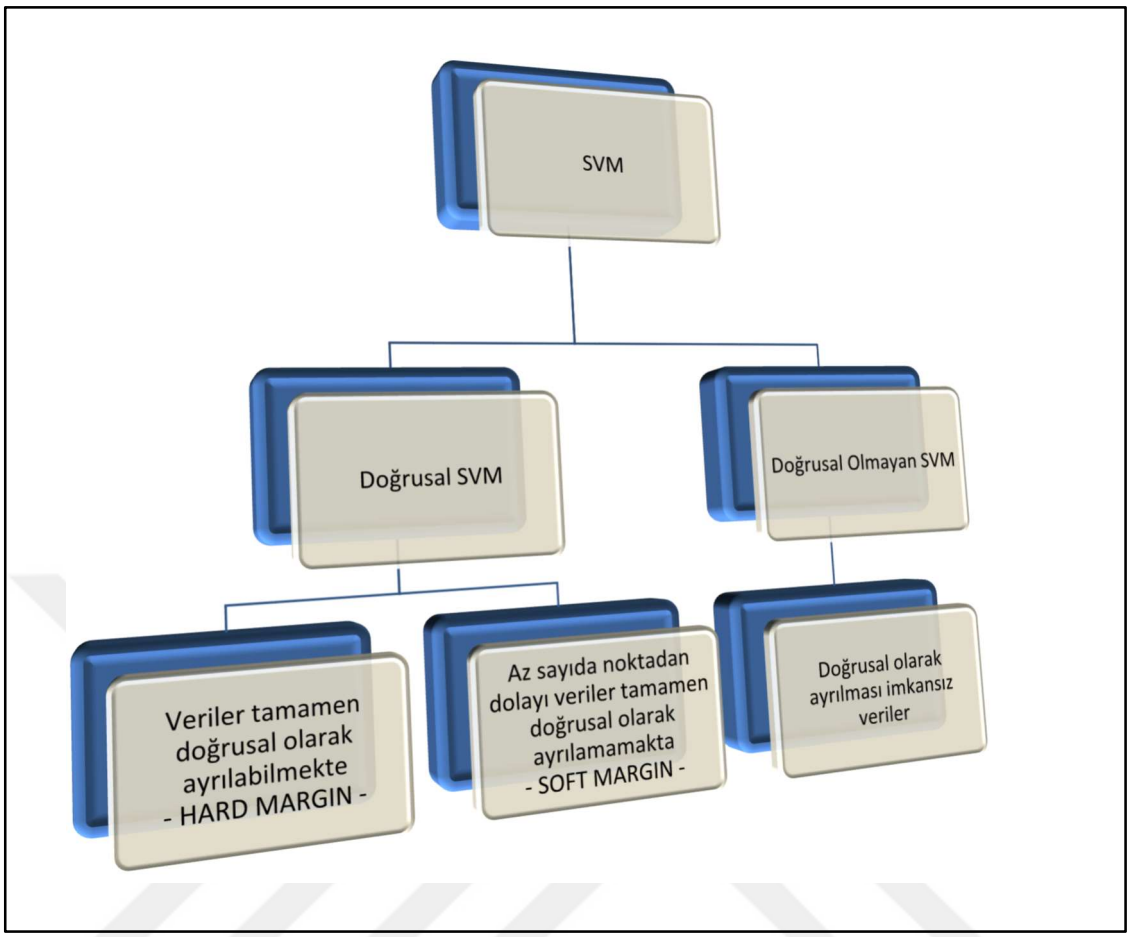
genelleştirmeye çalışma temeline dayanır. İstatistiksel öğrenme teorisine ve yapısal risk minimizasyonunu kullanır.

SVM'ler parametrik olmayan sınıflandırıcılardır. Yani dağılım hakkında herhangi bir ön-bilgi paylaşımı yoktur. Eğitim veri setlerinde girdi ve çıktılar eşlenir. Bu eşlemeler aracılığıyla test veri setinde ve yeni veri setlerinde girdi değişkenini sınıflayacak karar fonksiyonları elde edilir.

Girdi verileri doğrusal olarak ayrılabildiğinde verileri ayırabilecek sonsuz sayıdaki doğru içerisinden marjini en yüksek yapacak olan doğruyu seçmek hedeflenir. Girdi verileri doğrusal olarak ayrılamadığında çalışma verisini yüksek boyuta dönüştürmek için doğrusal olmayan haritalama (*mapping*) kullanılmaktadır. Verinin taşındığı bu yeni boyutta marjini en büyük (*optimal*) ayrıcı düzlemi araştırılır.

Yüksek doğrulukta sınıflandırma yapabilmesi, karmaşık karar sınırlarının modellenebilmesi, çok sayıda bağımsız değişkenle çalışabilme hem doğrusal olarak ayrılabilen hem doğrusal olarak ayrılamayan veriler üzerine uygulayabilme ve diğer birçok yönteme göre *overfitting* sorununun daha az olması, SVM'lerin avantajları olarak sıralanabilir. Olasılıksal tahminler üretememe, çekirdek fonksiyonlar için *Mercer Koşulu* zorunluluğu (çekirdek fonksiyonlar pozitif tanımlı sürekli simetrik fonksiyonlar olmalı) ise SVM'lerin dezavantajları olarak gösterilebilir.

SVM, nesne tanıma (yüz tanıma, parmak izi tanıma v.b.), el yazısı tanıma, zaman serisi tahmin testleri, biyoinformatik (microarray dizilerin analizi) gibi birçok alanda kullanılmaktadır. (Dolgun, 2013)



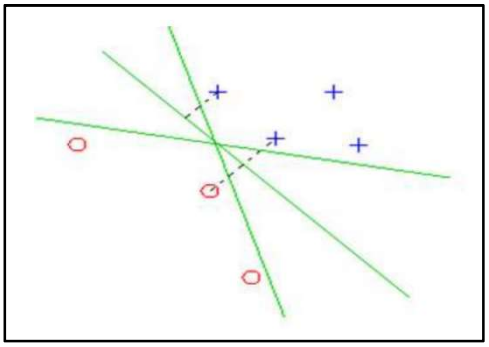
Şekil 5.4. SVM'lerin nokta dağılımlarına göre sınıflandırılması (Dolgun, 2013)

Şekil 5.4'te SVM'lerin nokta dağılımlarının şekline göre kategorizasyonu görülmekteir.

5.2.2.1. Doğrusal SVM

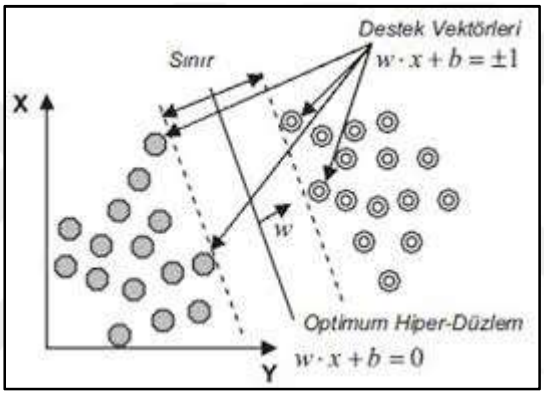
Doğrusal SVM'ler verilerin tamamının doğrusal olarak ayrılabildiği (*hard margin*) ve verilerin tamamının doğrusal olarak ayrılamadığı (*soft margin*) destek vektör makineleri olmak üzere iki kısımda incelenebilir.

Hard Margin: Verilerin tamamının doğrusal olarak ayrılabildiği durumdur. En temel SVM uygulamasıdır. Bir veri setini iki sınıfa ayırabilecek sonsuz sayıda doğru çizilebilir. (Dolgun, 2013)



Şekil 5.5. SVM - Verilerin tamamının doğrusal olarak ayrılabildiği durum (Ayhan ve Erdoğmuş, 2014)

Şekil 5.5'te görülen, noktaları ayıran sonsuz doğru içerisinde amaç bilinmeyen veri seti ile karşılaşıldığında sınıflama hatasını minimize edecek doğruyu seçmektir.



Şekil 5.6. SVM – Doğrusal tam ayırma (Küçüksille ve Ateş, 2013)

$$D = \{(\vec{x}_i, y_i), i = 1 \dots N\} \tag{5.4}$$

N adet elemandan oluşan eğitim veri kümesi formül (5.4)'deki gibi olduğu kabul edilirse buradaki $y_i \in \{-1, +1\}$ sınıf etiketi, $\vec{x}_i \in R^n$ olup n boyutlu uzayda herhangi bir örnektir.

$$f(\vec{x}) = \vec{w}^T \vec{x} + b \tag{5.5}$$

Formül (5.5) ifadesindeki $\vec{w}^T$ karar fonksiyonun normalini, $\vec{x}$ ifadesi bu doğru üzerinde bulunan noktaları, b ise eğilim değerini göstermektedir. Amaç $\vec{w}^T$ ve b' yi eğitim verileri yardımıyla bulmaktır. Yani sistemi eğitmektir. Tüm destek vektör makinelerinde amaç Şekil 5.5 ve Şekil 5.6'da olduğu gibi verileri 2 sınıfa ayırmaktır. Şekil 5.6'de kesikli çizgiler ile ifade edilen doğrular üzerindeki vektörler, *destek vektör* olarak isimlendirilir ve yumuşak ayrım çizgisi bu vektörler üzerinden geçer. İki yumuşak ayrım

çizgisinin ortasındaki doğru ise sert ayrımıdır ve formül (5.6) fonksiyonu ile çizilir. (Küçüksille ve Ateş, 2013)

$$f(\vec{x}) = \vec{w}^T\vec{x} + b = 0 \tag{5.6}$$

Formül (5.5) ifadesindeki $\vec{w}^T$ ve $\vec{x}$ vektörel büyüklükler olup, formül (5.7) fonksiyonunun sade halini ifade eder.

$$f(\vec{x}) = w_1x_1 + w_2x_2 + w_3x_3 + \cdots + w_nx_n + b \tag{5.7}$$

Formül (5.5) deki fonksiyonun 1'den büyük veya 1'e eişit olması durumunda $y_i = 1$ veya aynı fonksiyonun -1'den küçük veya -1'e eşit olması durumunda $y_i = -1$ olur. Bu iki fonksiyonu formül (5.8)'deki gibi kısaltabiliriz.

$$y_i\,(\boldsymbol{w^T x_i}) + b \geq 1 \tag{5.8}$$

Bu ayırma işleminin püf noktası sınır değerini maksimum yapmak ve bu sayede en iyi ayrıma sahip olmaktır. Veri setini sınıflara ayırabilecek sonsuz sayıda çoklu düzlem çizilebilmesine karşın, amaç bilinmeyen veri seti ile karşılaşıldığında sınıflama hatasını en küçük yapacak aşırı düzlemi seçmektir. Bunun için maksimum sınırlı aşırı düzlem tekniği önerilmiştir. Sınır değerinin büyüklüğü genelleme SVM'nin sınıflandırma kabiliyetini arttırır. x_1 değeri $f(\vec{x}) = \vec{w}^T\vec{x} + b = 1$ fonksiyonu üzerinde bir nokta ve x_3 değeri $f(\vec{x}) = \vec{w}^T\vec{x} + b = -1$ fonksiyonu üzerinde bir noktadır. Sınır değerini bulmak için,

$\vec{w}^T x_1 + b = +1$

$\vec{w}^T x_3 + b = -1$

$\vec{w}^T x_3 + b = -1$ ifadesini -1 ile çarpıp $\vec{w}^T x_1 + b = +1$ ifadesi ile toplarsak formül (5.9) ifadesi bulunur. Bu ifade de eşitlikte yerine yazılırsa, formül (5.10) ifadesi elde edilir.

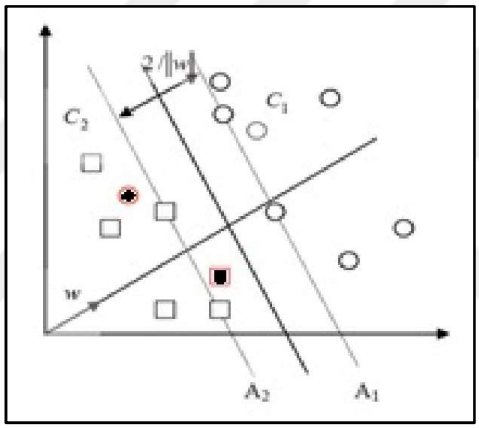
$$x_1 = x_3 + \lambda w \tag{5.9}$$

$$\lambda = 2 \,/\, \vec{w}^2 \tag{5.10}$$

Hedef, λ değerini maksimum yapmak olduğu için 1/ λ ifadesi minimum olmalıdır. Buna bağlı sınırlama ise formül (5.11)'de gösterilmiştir. (Küçüksille ve Ateş, 2013)

$$y_i\,(\boldsymbol{w^T x_i} + b) - 1 \geq 0\ ,\ y_i \in \{-1, +1\} \tag{5.11}$$

Soft Margin: Veriler bazı durumlarda doğrusal olarak %100 performansla ayrılamayabilir. Şekil 5.7'ye baktığımız zaman içi dolu karenin destek vektörü ile sert ayırma çizgisinin arasına düştüğünü görüyoruz. Bu durum *ayrılamama* olarak adlandırılır. Yine Şekil 5.7'de içi dolu dairenin, sert ayrılma çizgisinin karşı tarafına düştüğü görülmektedir. Bu durum da *yanlış ayrılma* olarak adlandırılır. Bu durumlarda, doğruların minimum hata ile ayırma sağlayacak şekilde ayarlanması gerekir. (Küçüksille ve Ateş, 2013)



Şekil 5.7. SVM – Doğrusal tam ayıramama (Küçüksille ve Ateş, 2013)

Soft margin yaklaşımı bu tip problemler için deneme hatalarını tolere edebilecek bir yaklaşımdır.

Gevşek değişken adında negatif olmayan bir değişken tanımlanır. Gevşek değişken, bir x_i değerinin sınırdan olan sapma uzaklığı olarak ifade edilebilir. Hard margin'de elde edilen kısıtlara bu gevşek değişken eklenir. Gevşek değişken eklendikten sonra formüllerin son durumu formül (5.12) ve formül (5.13)'te gösterilmektedir.

$$f(\vec{x}) = \vec{w}^T\vec{x} + b \geq 1 - \xi_i\ ,\ \xi_i\ \geq 0\ ,\ \forall_i \tag{5.12}$$

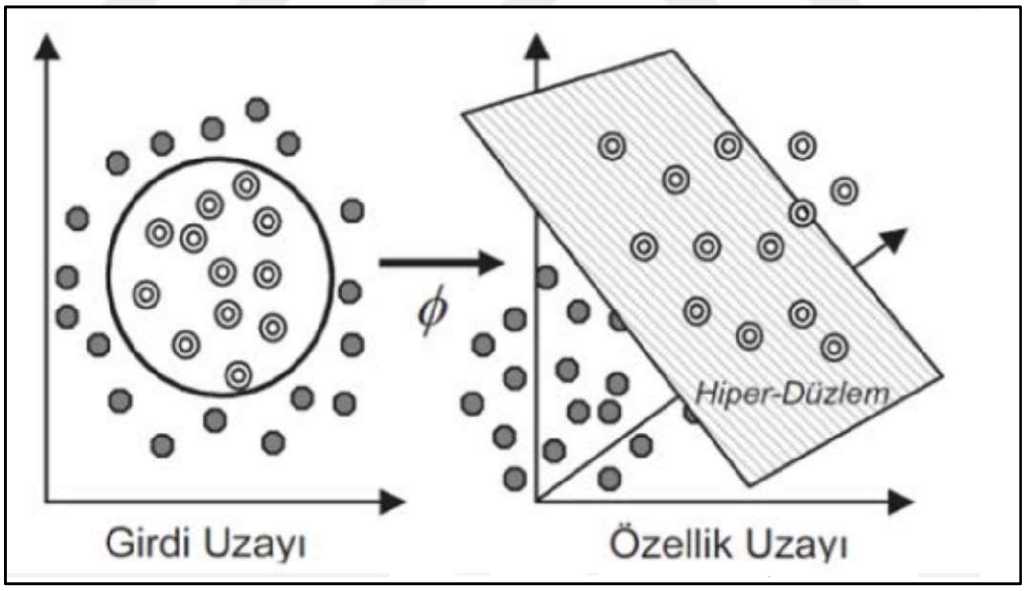
$$f(\vec{x}) = \vec{w}^T\vec{x} + b \leq 1 + \xi_i\ ,\ \xi_i\ \geq 0\ ,\ \forall_i \tag{5.13}$$

Ayrılamama durumunda $0 \leq \xi_i \leq 1$ olur. *Yanlış ayrılma* durumunda da $\xi_i \geq 1$ olur. Minimizasyona aynen devam edilir. Kullanıcı tarafından belirlenen *hata maliyeti (C)* de eklenir. Bu hata maliyeti, marjin maksimizasyonu ile deneme hatası minimizasyonu arasındaki ödünleşimi belirler. Yüksek hata maliyeti değeri, yüksek hata beklentisi anlamına gelir. Fonksiyona hata maliyeti değeri de eklendiği zaman minimizasyon formülü, formül (5.14)'deki gibi gibi olur. (Küçüksille ve Ateş, 2013)

$$^1/_2 \ \|w\|^2 + C \ \sum_i \xi_i^{\ 2} \tag{5.14}$$

5.2.2.2. Doğrusal olmayan SVM

Veriler doğrusal olarak ayrılamadığında, veriyi doğrusal olmayan haritalama (ϕ) yaparak orijinal girdi uzayından daha yüksek daha yüksek boyuttaki bir uzaya aktarılır. Bu yeni boyuttaki veriyi en iyi ayıracak düzlem araştırılır. (Dolgun, 2013). Şekil 5.8'de doğrusal olmayan bir SVM'nin ayrılması gösterilmiştir.



Şekil 5.8. Doğrusal olmayan SVM (Küçüksille ve Ateş, 2013)

Doğrusal SVM'lerden farklı olarak x yerine $\phi(x)$ kullanılır. Dönüştürülmüş uzayda karar fonksiyonu formül (5.15) olarak ifade edilebilir. (Dolgun, 2013)

$$< w, \phi(x) > + b = 0 \tag{5.15}$$

Şekil 5.8'de iki boyutlu uzaydaki düzlemlere x_1 ve x_2, üç boyutlu uzaydaki düzlemlere de z_1, z_2 ve z_3 diyelim. Bu durumda fonksiyonların özellik uzayı formül (5.16)'daki gibi olur.

$$\phi: R^2 \rightarrow R^3 \quad (x_1, x_2) \rightarrow (z_1, z_2, z_3) => (x_1^2, \sqrt{2}x_1x_2, x_2^2) \tag{5.16}$$

Özellik uzayında haritalanmış girdi vektörlerinin iç çarpımı formül (5.17) şeklinde ifade edilebilir. (Soman ve ark., 2011)

$$(x_1^2\ddot{x}_1^2 + 2x_1\ddot{x}_1x_2\ddot{x}_2 + x_2^2\ddot{x}_2^2)^2 \tag{5.17}$$

Böylece veri seti iki boyutlu uzaydan üç boyutlu uzaya taşınarak haritalama işlemi gerçekleştirilmiş olur (Soman ve ark., 2011). Sonuç olarak doğrusal olmayan SVM için özellik uzayında tanımlı ayırma hiper düzlemine bağlı olarak sınıflandırıcı karar fonksiyonu formül (5.18)'deki eşitlik ile gösterilebilir. (Ayhan ve Erdoğmuş, 2014)

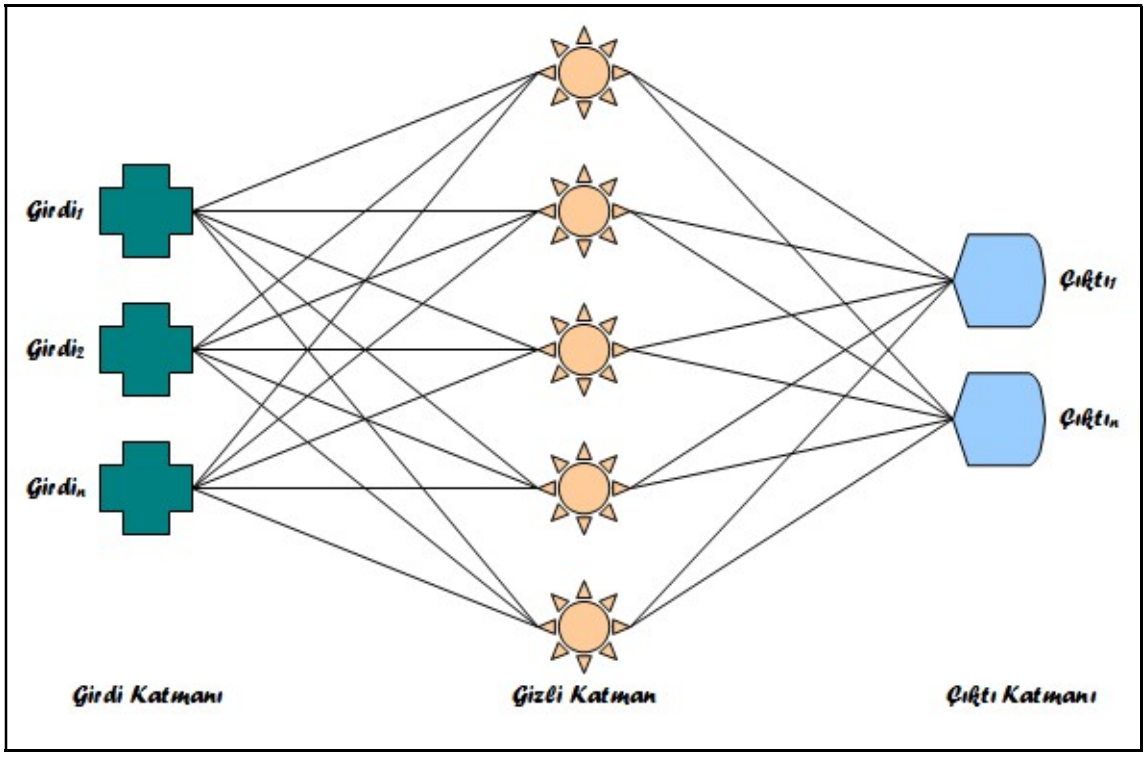
$$f(x) = sign\big((w^2, x_i) + b\big) = sign(\textstyle\sum_{i=1}^{n} y_1\alpha_1(\phi(x_1), \phi(x_i))) \tag{5.18}$$

5.2.3. Yapay sinir ağları

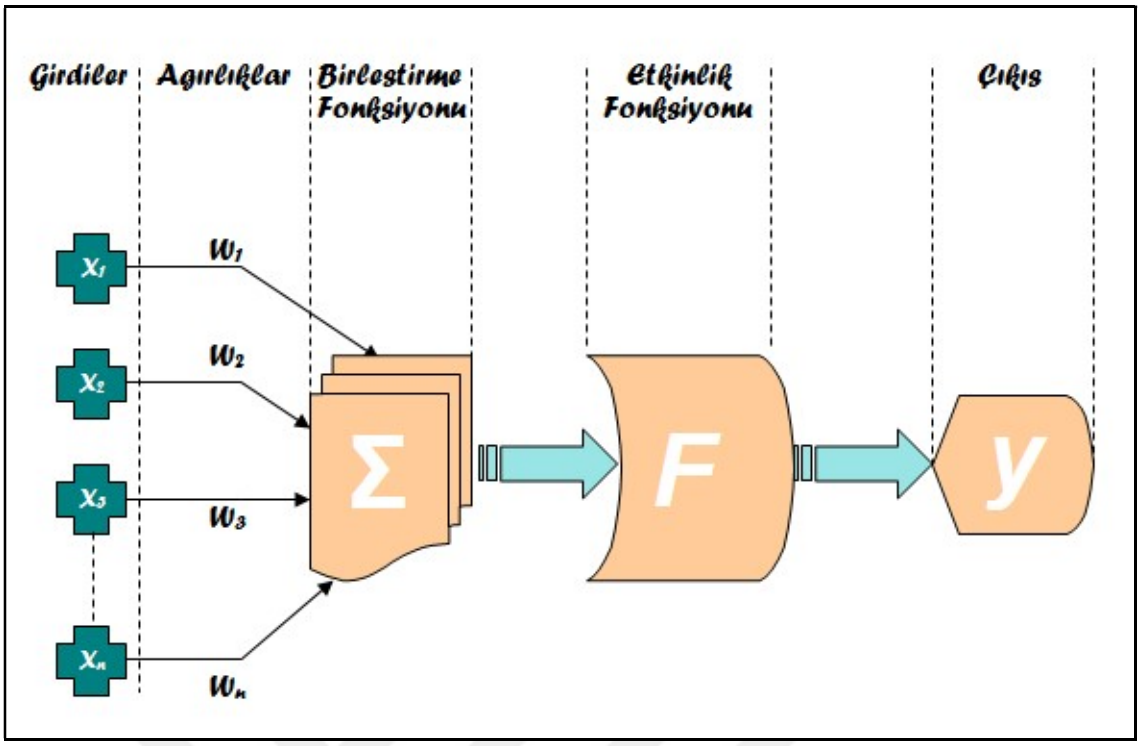
YSA, insan beyninin çalışma sisteminin yapay olarak benzetimi çabalarının bir sonucu olarak ortaya çıkmıştır. YSA, genel olarak insan beyninin ya da merkezi sinir sisteminin çalışma prensiplerini taklit eden bilgi işleme sistemleridir. YSA, yapay zekâ çalışmaları kapsamında, insan beynindeki birçok işlem elemanının veya basit işlemcilerin çalışma prensiplerine göre tasarlanmıştır. Olaylar arasındaki ilişkileri bilinen örnekleri kullanarak öğrenme, karar verme, sonuç çıkarma davranışlarının yapay olarak modellenmesi esasına dayalı, paralel çalışma özelliği olan bir bilgi işleme sistemidir. Bu yüzden YSA bazen, *paralel dağıtılmış işleme sistemleri* (*Parallel Distributed Processing Systems*) ya da *bağlantıcı sistemler* (*Connectionist Systems*) olarak da adlandırılırlar (Şahin, 2008).

5.2.3.1. Yapay sinir ağlarının yapısı

Yapay sinir hücreleri, YSA' nın çalışmasına esas teşkil eden en küçük bilgi işleme birimidir. Geliştirilen hücre modellerinde bazı farklılıklar olmakla birlikte genel özellikleri ile bir yapay sinir modeli, girdiler, ağırlıklar, birleştirme fonksiyonu (toplama), aktivasyon (etkinleştirme) fonksiyonu ve çıktılar olmak üzere 5 bileşenden meydana gelir. Şekil 5.9'da bu bileşenler gösterilmektedir. Şekil 5.10'da ise yapay bir sinirin temel bileşenleri gösterilmektedir (Şahin, 2008).



Şekil 5.9. Yapay bir sinirin katmanları



Şekil 5.10. Yapay bir sinirin temel bileşenleri (Şahin, 2008)

Şekil 5.10'teki girdiler ($x_1, x_2, ..., x_i$), diğer hücrelerden ya da dış ortamlardan hücreye giren bilgilerdir. Bilgi, bir önceki sinirlerden veya dış dünyadan gelir. Bir sinir genellikle gelişigüzel birçok girdileri alır (Engin ve Fığlalı, 2002).

Bilgiler, bağlantılar üzerindeki ağırlıklar ($w_1, w_2, ..., wi$) üzerinden hücreye girer ve ağırlıklar, ilgili girişin hücre üzerindeki etkisini belirler. Öğrenme esnasında sürekli değişerek girdi ve çıktı arasındaki ilişkiyi yakalamaya (optimize etmeye) çalışır. Her bağlantının bir ağırlığı vardır. Ağırlık büyüdükçe etki de büyür. Ağırlığın sıfır olması, hiçbir etkinin olmaması; negatif olması ise etkinin ters yönde olması demektir (Şahin, 2008).

Birleştirme fonksiyonu, bir hücreye gelen net girdiyi hesaplayan bir fonksiyondur. Net girdi genellikle girişlerin ilgili ağırlıkla çarpımlarının toplamıdır. Toplama yanında, çarpım, maksimum, minimum, çoğunluk ve kümülatif toplam fonksiyonları da birleştirme fonksiyonu olarak kullanılmaktadır (Öztemel, 1992).

Aktivasyon (etkinlik) fonksiyonu, birleştirme fonksiyonundan elde edilen net girdiyi bir işlemden geçirerek hücre çıktısını belirleyen ve genellikle doğrusal olmayan bir fonksiyondur. Bir etkinlik fonksiyonunun kullanım amacı, zaman söz konusu olduğunda toplama fonksiyonunun çıkışının değişmesine izin vermektir (Engin ve Fığlalı, 2002).

Çıkış Fonksiyonu, etkinlik fonksiyonu sonuçlarının gönderildiği yerdir. Çıkış fonksiyonu, etkinlik fonksiyonundan aldığı sonuçları ya bir sonraki işlem elemanına (sinirlere) veya ağın dışına gönderir. Bir çıkış, kendinden sonra gelen herhangi bir sayıdaki diğer sinirlere giriş olabilir (Şahin, 2008).

Yukarıda bahsedilen beş bileşen arasında yapılan işlemler kısaca şu şekildedir: x_i sembolü ile gösterilen girişlerin her biri ağırlık ile (w_i) çarpılır. Formül (5.19)'da gösterilen çarpımların toplamı ile elde edilen sonuç birleştirme (toplama) fonksiyonu sonucudur.

$$\Sigma(x_i \times w_{i)} \tag{5.19}$$

Elde edilen bu değer giriş değerleri ile karşılaştırılır. Eğer toplam başlangıç değerinden büyükse doğrusal olmayan (F) etkinlik fonksiyonunu kullanarak bir çıkış (y_j) hesaplar. y_j çıkış değeri, bu toplam ile başlangıç değeri arasındaki farkın doğrusal olmayan (F) fonksiyonudur. Aktivasyon (Etkinlik) fonksiyonu formül (5.20) olmak üzere çıkış fonksiyonu formül (5.21) şeklinde gösterilir.

$$F(s)= 1 /(1+e^{-s}) \tag{5.20}$$

$$y = F (\Sigma (x_i \, x \, w_{i)} - t) \tag{5.21}$$

Burada t başlangıç değeridir. Doğrusal olmayan F etkinlik fonksiyonu, bir modelleme seçimi ve yapay sinir ağı modelinde istenen çıkış işareti cinsinden bir fonksiyondur. Bu fonksiyon için en fazla kullanılan seçimler ise *sigmoid, basamak ve rampa* fonksiyonlarıdır. Tüm yapay sinir ağları bu temel yapıdan türetilmiştir (Şahin, 2008).

5.2.3.2. Yapay sinir ağlarının avantajları

1. Uzman sistemler gibi bilgiyi kurallar halinde istemezler,
2. Öğrenebilir ve hiç karşılaşmadıkları bir problemi çözebilirler,
3. Paralel yapıları nedeniyle çok hızlı çalışırlar,
4. Matematiksel modele ihtiyaç duymazlar,

5. Sistemin herhangi bir işlem noktasında ortaya çıkabilecek bir hatanın sistemi çökertmemesi için hataya izin verebilecek bir tarzda tasarlanabilir. (Engin ve Fığlalı, 2002)

5.2.3.3. Yapay sinir ağlarının dezavantajları

1. Çıkardıkları sonuçları nasıl ve neden çıkardığını açıklayamaz (kapalı kutu),
2. Eğitimleri oldukça zaman alıcı ve zordur,
3. Bazı ağlar hariç kararlılık analizleri yapılamaz,
4. Farklı sistemlere uyarlanması zor olabilir. (Engin ve Fığlalı, 2002)

5.2.3.4. Yapay sinir ağlarının uygulama alanları

Genel olarak YSA model seçimi ve sınıflandırılması, işlev tahmini, en uygun değeri bulma ve veri sınıflandırması gibi işlerde başarılıdır. Bugün ise sayısal optimizasyon tasarımlarını da içine alacak şekilde, klasik yöntemlerle çözülemeyen problemlere başarılı çözümler getirmektedir. Böylece matematik ve fizik gibi temel bilimlerle, elektrik, bilgisayar ve makine mühendisliği gibi uygulamalı alanlarda kullanılır hale gelmiştir. Bunların yanında YSA'nın uygulama alanları aşağıdaki şekilde sayılabilir (Şahin, 2008):

- Arıza Analizi ve Tespiti
- Tıp Alanında
- Savunma Sanayi
- Haberleşme
- Üretim
- Otomasyon
- Kalite kontrol
- Görüntü tanıma

5.2.4. Genetik algoritmalar

Genetik algoritmalar (*Genetic Algorithms - GA*), doğada gözlemlenen evrimsel sürece benzer bir şekilde çalışan arama ve eniyileme yöntemidir. Karmaşık çok boyutlu arama uzayında en iyinin hayatta kalması ilkesine göre bütünsel en iyi çözümü arar (Wikimedia, 2016).

GA, problemlere tek bir çözüm üretmek yerine farklı çözümlerden oluşan bir çözüm kümesi üretir. Böylelikle, arama uzayında aynı anda birçok nokta değerlendirilmekte ve sonuçta bütünsel çözüme ulaşma olasılığı yükselmektedir. Çözüm kümesindeki çözümler birbirinden tamamen bağımsızdır. Her biri çok boyutlu uzay üzerinde bir vektördür (Wikimedia, 2016).

5.2.4.1. Terminoloji

Seleksiyon (Seçim): İki ebeveyn kromozomun *f(x)* fonksiyonuna göre seçimidir. Burada uygunluk derecesi yüksek olanın seçilme şansı yüksektir.

Çaprazlama: Yeni bir fert oluşturmak için ebeveynlerin bir çaprazlama olasılığına göre çaprazlanmasıdır. Eğer çaprazlanma uygulanmazsa, bireyler atalarının birer kopyası olacaktır.

Mutasyon: Kromozom üzerindeki bazı genlerin değerleri değiştirilerek nesillerin yozlaşması önlenir.

Ekleme: Yeni bireyin yeni topluma eklenmesi işlemidir.

Değiştirme: Algoritmanın yeniden çalıştırılmasında oluşan yeni toplumun kullanılmasıdır.

Yeni Uygunluk Değerinin Hesaplanması: *Ekleme* ve *Değiştirme* işlemleri tamamlandıktan sonra yeni değerlerle yeniden uygunluk değerlerinin hesaplanması.

Test: Eğer sonuç tahmin ediliyorsa, algoritmanın sona erdirilmesi ve son toplumun çözüm olarak sunulması adımıdır. Sonuç tahmin edilemiyorsa *Seleksiyon* adımından başlanarak algoritma tekrarlanır.

5.2.4.2. Genetik algoritmaların yapısı

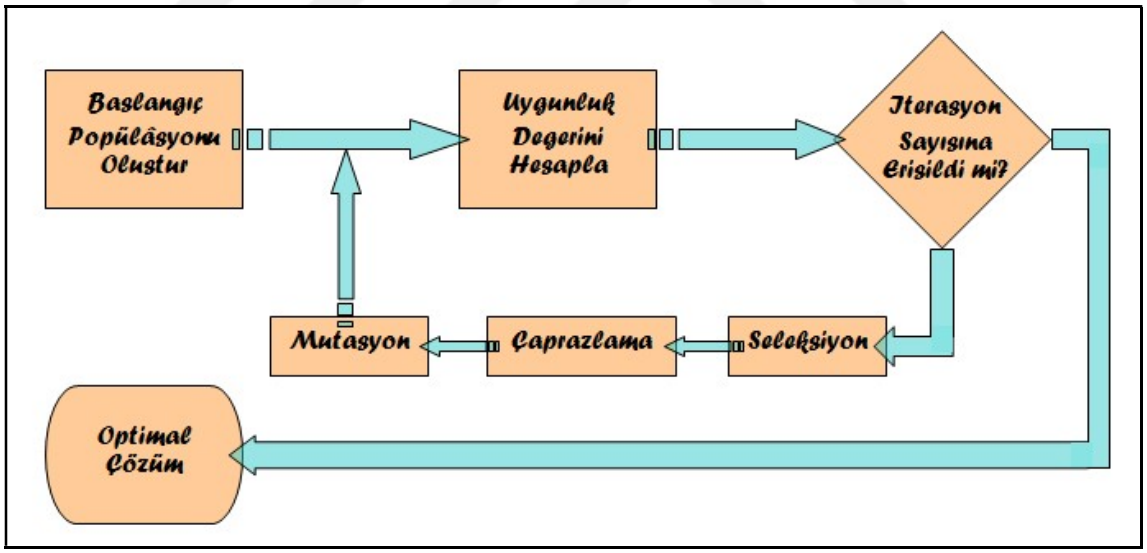
GA, evrim sürecinden etkilenerek, canlılarda yaşanan genetik sürecin bilgisayar ortamında gerçekleştirilmesi işlemidir. İşlemler bilgisayar hafızasına depo edilmiş

kromozomlar üzerinde icra edilmektedir. Çaprazlama operatörü vasıtasıyla, kromozomlar arasındaki genetik bilgi sürekli olarak değişmekte ve topluluğun başarısı artmaktadır (Yurtcu ve İçağa, 2006).

GA, doğal seçim ilkesine dayanan bir sayısal optimizasyon yöntemidir. Genetik algoritma, çözüm dizilerinden oluşan bir başlangıç nesliyle, çaprazlama ve mutasyon gibi doğal seçim operatörlerini kullanmaktadır (Yurtcu ve İçağa, 2006, Papadrakakis ve Lagaros, 1994).

GA, bağımsız parametrelerin kromozomlar içinde kodlanması gerekmektedir. Yığındaki her birey, ikili düzende veya tamsayı olarak kodlanmaktadır (Yurtcu ve İçağa, 2006).

GA, oldukça genel prensiple, Şekil 5.11'deki akış şemasında görüldüğü gibi çalışmaktadır. Öncelikle ele alınan problem için bir rastgele *n* kromozomlu popülasyon oluşturulur. Daha sonra popülasyondaki her bir kromozom için *f(x)* uygunluk fonksiyonu hesaplanır. Yeni bir popülasyon oluşuncaya kadar Şekil 5.11'deki adımlar tekrar edilir (Yurtcu ve İçağa, 2006).



Şekil 5.11. GA – Genel akış şeması

Genetik algoritmada, probleme ait en optimal çözümün bulunabilmesi,

- Bireylerin gösteriminin doğru bir şekilde yapılabilmesine,
- Uygunluk fonksiyonunun etkin bir şekilde oluşturulabilmesine ve
- Doğru genetik işlemcilerin seçilebilmesine bağlıdır.

GA ancak

- Arama uzayının büyük ve karmaşık olduğu,

- Mevcut bilgiyle sınırlı arama uzayında çözümün zor olduğu,

- Problemin belirli bir matematiksel modelle ifade edilemediği,

- Geleneksel eniyileme yöntemlerinden istenen sonucun alınamadığı durumlarda etkili ve kullanışlıdır (Wikimedia, 2016).

5.2.4.3. Genetik algoritmaların avantajları

1. Tek çözüm değil, birden fazla optimum çözüm elde edilebilir.
2. Çok sayıda parametre ile çalışabilme imkanı vardır.
3. Amaç fonksiyonunu geniş bir spektrumda araştırır.
4. Karmaşık amaç fonksiyonu parametrelerini optimize eder.
5. Kısa sürede iyi sonuçlar alınabilir.

5.2.4.4. Genetik algoritmaların dezavantajları

1. Son kullanıcının modeli anlaması güçtür.
2. Problemi GA ile çözmeye uygun hale getirmek zordur.
3. Uygunluk fonksiyonunu belirlemek zordur.
4. Çaprazlama ve mutasyon tekniklerini belirlemek zordur.
5. Elde ettiği sonuçlar her zaman optimum çözüm olmayabilir.

5.2.4.5. Genetik algoritmaların uygulama alanları

- Optimizasyon (bakım, servis, depo, toplama)

- Otomatik programlama ve bilgi sistemleri

- Makine öğrenmesi

- Ekonomik ve sosyal sistem modelleri (böcek kolonileri, çok etmenli sistemlerde iş birliği)

- Görüntü işleme

- Popülasyon genetiği

- Evrim ve öğrenme

- Montaj hattı dengeleme problemleri

- Çizelgeleme problemleri

- Tesis yerleşim problemleri

- Atama problemi

- Hücresel üretim problemi

- Sistem güvenilirliği problemi

- Taşıma problemi

- Gezgin satıcı problemi

- Araç rotalama problemi

- Minimum yayılan ağaç problemi

5.2.5. Karar ağacı

Karar ağacı (Decision Tree – DT) algoritmalarının kullanımı ilk olarak Morgan ve Sonquist tarafından Michigan Üniversitesi'nde 1970'li yılların başlarında kullanılan AID karar ağacı ve algoritması ile başlamıştır. Yöntem, 60 yılı aşkın süredir kullanılmaktadır. (Koç, 2016)

Ağaç tabanlı öğrenme algoritmaları, en çok kullanılan ve denetimli öğrenme yöntemlerinden biri olarak düşünülmektedir. Ağaç tabanlı yöntemler, yüksek doğruluk, kararlılık ve yorumlanma kolaylığına sahiptir. Doğrusal modellerin aksine doğrusal olmayan ilişkileri de oldukça iyi eşleyebilirler. Hem sınıflandırma ve hem de regresyon problemlerinin çözümünde kullanılabilirler. DT, her türlü veri bilimi probleminde yaygın şekilde kullanılmaktadır.

Karar ağacı öğrenmesi, *endüktif (inductive)* çıkarım için en yaygın kullanılan pratik yöntemlerden birisidir. Karar ağacı öğrenmesi, öğrenilen fonksiyonun bir karar

ağacı tarafından temsil edildiği, kesikli değerli hedef fonksiyonlarını yaklaştırmak için kullanılan bir yöntemdir. DT, sınıflandırma problemlerinde çoğunlukla kullanılan bir denetimli (supervised) öğrenme algoritmasıdır. Hem kategorik hem de sürekli giriş ve çıkış değişkenleri için çalışır. Öğrenilen ağaçlar insan okunabilirliğini artırmak için *if-then* kural setleri olarak temsil edilebilirler. DT, bir ağaç yapısı biçiminde sınıflandırma veya regresyon modelleri oluşturur. Bir veri kümesini daha küçük ve daha küçük alt kümelere bölerken, aynı zamanda ilişkili bir karar ağacı aşamalı olarak geliştirilir. Nihai sonucu, karar düğümleri ve yaprak düğümleri olan bir ağaçtır. Bir karar düğümü, iki veya daha fazla dallara sahiptir. Yaprak düğüm bir sınıflandırma veya kararı temsil eder. Bir ağaçtaki en üstteki karar düğümü, kök düğüm olarak adlandırılan en iyi belirleyiciye karşılık gelir. Karar ağaçları hem kategorik hem de sayısal verileri işleyebilir. Karar ağaçlarını şöyle ikiye ayırabiliriz:

Kategorik Değişken Karar Ağacı: Kategorik hedef değişkeni olan karar ağacı, kategorik değişken karar ağacı olarak adlandırılır. *Sınıflandırma karar ağaçları* da denilebilir.

Sürekli Değişken Karar Ağacı: Karar ağacı sürekli hedef değişkenine sahipse, Sürekli değişken karar ağacı olarak adlandırılır. *Regresyon karar ağaçları* da denilebilir. (Anonim, 2018)

5.2.5.1. Terminoloji

Kök Düğüm: Tüm örneği temsil eder ve bu düğüm daha sonra iki veya daha fazla kümeye ayrılır.

Parçalama: Bir düğümün iki veya daha fazla alt düğümlere bölünmesi işlemidir.

Karar Düğümü: Bir alt düğüm başka alt düğümlere bölünürse, karar düğümü olarak adlandırılır. Giriş verilerinin test edildiği, soruların sorulduğu ve hangi yöne yöneleceklerini belirleyen karar düğümleridir.

Dal: Düğümler arası ayrımlardır. Soruların cevaplarını temsil eder.

Yaprak Düğümü: Bölünmeyen düğümlere *yaprak* veya *terminal* düğümü denir. Kategorilerin bulunduğu sınıf etiketleridir.

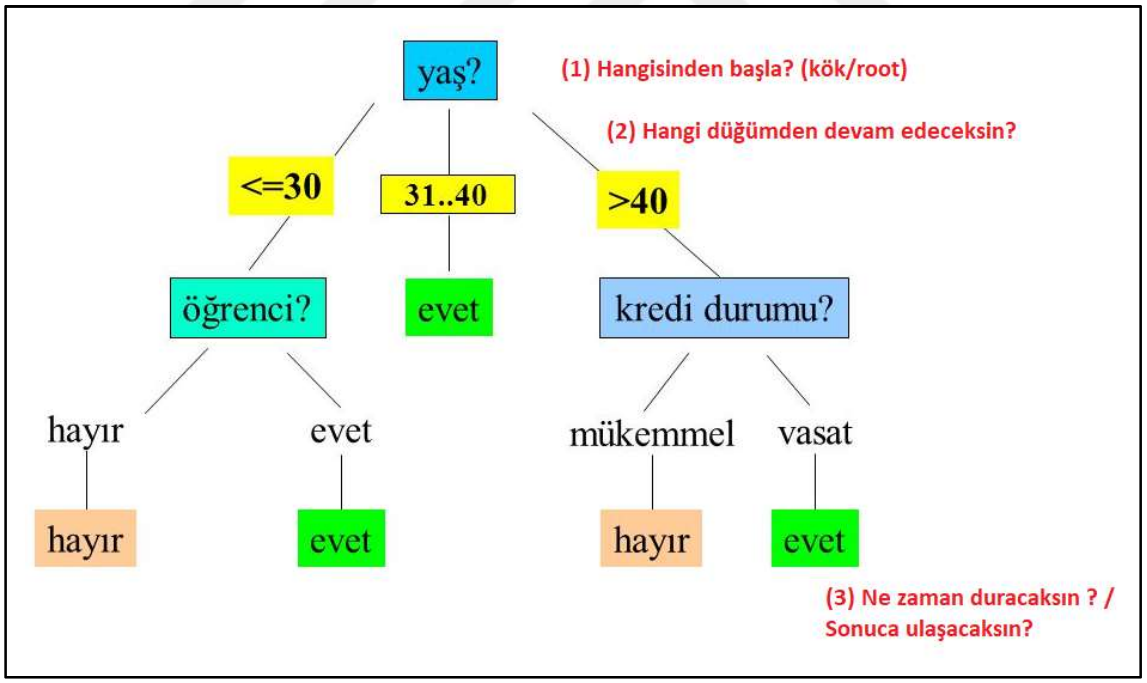
Budama: Karar düğümünün alt düğümlerini kaldırdığımızda, bu işleme budama denir. Yani parçalama işleminin tersi diyebiliriz.

Alt Ağaç: Tüm ağacın bir alt kısmı *şube* veya *alt-ağaç* olarak adlandırılır.

Ana ve Çocuk Düğümü: Alt düğümlere ayrılmış olan bir düğüme, alt düğümlerin *ana düğümü* ve alt düğümlerine de *çocuk* düğümü adı verilir. (Anonim, 2018)

5.2.5.2. Karar ağacının yapısı

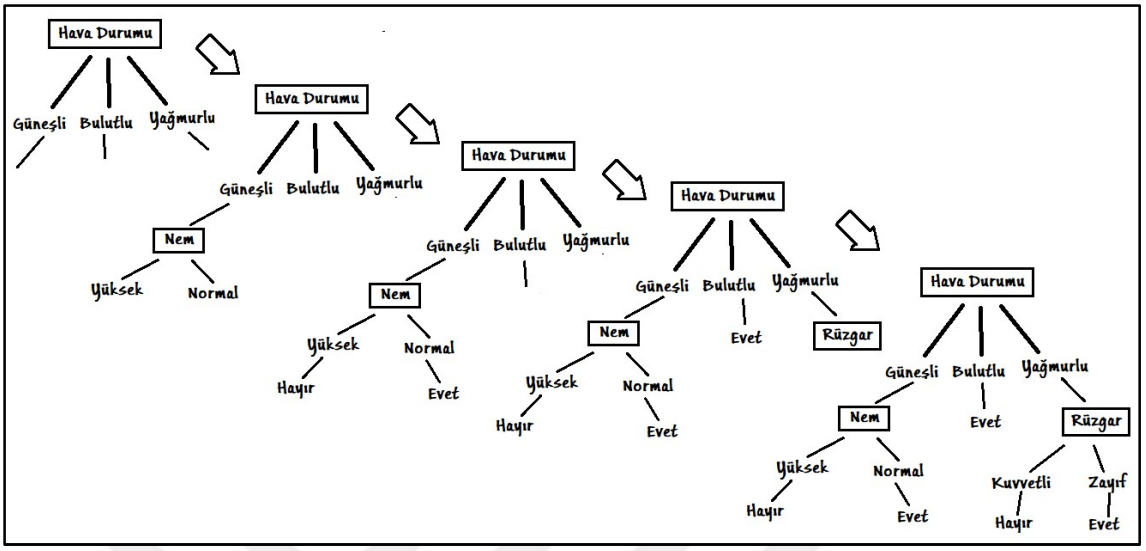
Ağaç yapısı şeklinde bir sınıflandırma algoritmasıdır. Tümevarım mantığının programlama ortamına taşındığı, basit ama çok yaygın bir metottur. Ayrık değerli parametrelerde çalışabilir. Gürültüye karşı dayanıklı algoritmalardır. En yaygın olarak kullanılanı *C4.5* algoritmasıdır. Karar ağacı algoritmalarının dayandığı tümevarımlı felsefeye dair temel sezgi, öğrenme özellikleriyle oluşturulacak karar ağacının olabildiğince küçük olmasının iyi olacağı yönündedir. Ürettikleri kuralların kolay anlaşılır olması, pek çok değişik problemde kolay uygulanabilir olmaları ve hem kategorik hem de sayısal verileri işleyebiliyor olması, karar ağaçlarının çok yaygın olarak kullanılmasını sağlamaktadır (Koç, 2016). Şekil 5.12'te rastgele oluşturulmuş bir karar ağacı örneği görülmektedir.



Şekil 5.12. Rastgele oluşturulmuş karar ağacı (Demiriz, 2006)

Algoritmanın uygulanmasında, öncelikle her öznitelik üzerinde bölme için bilgi kazancı hesaplanır (*entropy, variance, azalma*). Yeni bir düğüm için en fazla bilgi kazancını veren öznitelik seçilir. Bu özniteliğe dayalı olan eğitim verisi bölünür. En alt

düğümlere kadar tüm düğümler için bilgi kazancı hesaplamadan başlayarak aynı işlemler uygulanır. Bu adımlar grafiksel olarak Şekil 5.13'de gösterilmektedir.



Şekil 5.13. Karar ağacı algoritma adımları

Kök düğümde ağacın dengeli bir şekilde dallanması ve sınıflandırma algoritmasının verimli olması için özellikler içinden en uygun olanı seçilir. Kök düğüm için uygulanan algoritma, iç karar düğümleri için de söz konusudur.

Entropi

Rastgele bir değişkenin belirsizliğinin ölçüsüdür. Keyfi bir koleksiyonun saf olup olmadığını kategorize eder. *0* ile *1* arasında bir değerdir. Entropi ne kadar yüksek olursa elde edilen bilgi o kadar fazla olur. Örneklerin tamamı aynı sınıfa aitse *entropi = 0* olur. Örnekler sınıflar arasında tamamen eşit dağılmışsa *entropi = 1* olur. Örnekler, sınıflar arasında rastgele dağılmışsa entropi *0* ile *1* arasında bir değer alır. Entropi hesaplamasında kullanılacak fonksiyon, formül 5.22'de gösterilmiştir.

$$E(S) = -p(P)\log_2 p(P) - p(N)\log_2 p(N) \tag{5.22}$$

Bilgi Kazancı (Information Gain)

Entropi, tipik olarak eğitim örneklerini daha küçük alt gruplara bölmek için bir karar ağacında bir düğümü kullandığımızda değişir. Bilgi kazancı, işte bu değişimin bir ölçüsüdür. Entropideki bu değişim, normalde azalma olarak beklenir. Bu şekilde bizim için uygun olan özellik seçimini gerçekleştirebiliriz. Formül (5.23)'teki gibi hesaplanır.

$$BK = Sistemin\ Enformasyon\ Değeri - Özelliğin\ Enformasyon\ Değeri \quad (5.23)$$

Entropi hesapama ve özellik seçimi için kullanılacak örnek bir tablo, Çizelge 5.3'te gösterilmektedir.

Çizelge 5.3. Entropi hesaplama ve özellik seçimi – Örnek Tablo (Koç, 2016)

V_1	V_2	S
A	C	E
A	C	F
B	D	E
B	D	F

Bu örnekteki sınıf entropisini heaplamak için formül (5.22)'yi kullanıyoruz. Hesaplama şu şekilde olur:

$$H(S) = -\left(\frac{1}{2}\log_2\frac{1}{2} + \frac{1}{2}\log_2\frac{1}{2}\right) = 1$$

Aynı formülü (formül (5.22)) V_1 entropisini hesaplamak için kullanırsak:

$$H(V_1) = \frac{1}{4}H(A) + \frac{3}{4}H(B)$$
$$= \frac{1}{4}\,0 - \frac{3}{4}\left(\frac{1}{3}\log_2\frac{1}{3} + \frac{2}{3}\log_2\frac{2}{3}\right)$$
$$= 0 + \frac{3}{4}\,0{,}9183$$
$$= 0{,}6887$$

Değerini buluruz. Şimdi de V_2 entropisini yine formül (5.22) ile hesaplayalım:

$$H(V_2) = \frac{1}{2}H(C) + \frac{1}{2}H(D)$$
$$= \frac{1}{2} + \frac{1}{2}$$
$$= 1$$

Entropi seçimi için de formül (5.23)'ü kullanalım:

$$H(S) - H(V_1) = 1 - 0{,}06887 = 0{,}3113$$
$$H(S) - H(V_2) = 1 - 1 = 0$$

$BK(V_1) > BK(V_2)$ olduğundan, V_1 özelliği seçilir. (Koç, 2016)

Haftasonu Örneği

Karar ağacı algoritmasını bir örnek üzerinde uygulayalım. İlkönce veri setinden bir öğrenme kümesi oluşturuyoruz. Çizelge 5.4'te örnek için oluşturulmuş bir öğrenme kümesi görülmektedir.

Çizelge 5.4. Karar ağacı örnek öğrenme kümesi (Koç, 2016)

Haftasonu	Hava	Ebeveyn	Para	Karar (Kategori)
H1	Güneşli	Var	Çok	Sinema
H2	Güneşli	Yok	Çok	Tenis
H3	Rüzgârlı	Var	Çok	Sinema
H4	Yağmurlu	Var	Az	Sinema
H5	Yağmurlu	Yok	Çok	Evde
H6	Yağmurlu	Var	Az	Sinema
H7	Rüzgârlı	Yok	Az	Sinema
H8	Rüzgârlı	Yok	Çok	Alışveriş
H9	Rüzgârlı	Var	Çok	Sinema
H10	Güneşli	Yok	Çok	Tenis

Veri setindeki en ayırt edici özelliği belirleyip, ağacın kökü olarak alıyoruz.

Bunun için entropiyi formül (5.22)'de değerleri yerine koyarak hesaplıyoruz.

- 6 örnek için karar sinemaya gitmek
- 2 örnek için karar tenis
- 1 örnek için karar evde kalmak
- 1 örnek için karar alışverişe çıkmak

$$H(S) = -(\frac{6}{10})\log_2\frac{6}{10} - (\frac{2}{10})\log_2\frac{2}{10} - (\frac{1}{10})\log_2\frac{1}{10} - (\frac{1}{10})\log_2\frac{1}{10}$$

$$H(S) = 1{,}571$$

Olarak hesaplanır. Şimdi de bilgi kazançlarını hesaplayalım. Bunun için de formül (5.23)'ü kullanıyoruz:

BK(T, Hava) = ?

- Güneşli = 3 (1 sinema, 2 tenis)
- Rüzgarlı = 4 (3 sinema, 1 alışveriş)
- Yağmurlu = 3 (2 sinema, 1evde)

- Entropi($T_{güneşli}$)= $-\left(\frac{1}{3}\right)\log_2\frac{1}{3}-\left(\frac{2}{3}\right)\log_2\frac{2}{3}=0{,}918$
- Entropi($T_{rüzgarlı}$)= $-\left(\frac{3}{4}\right)\log_2\frac{3}{4}-\left(\frac{1}{4}\right)\log_2\frac{1}{4}=0{,}811$
- Entropi($T_{yağmurlu}$)= $-\left(\frac{2}{3}\right)\log_2\frac{2}{3}-\left(\frac{1}{3}\right)\log_2\frac{1}{3}=0{,}918$

BK(T, Hava) = entropi(T) – (P(güneşli) Entropi($T_{güneşli}$) + P(rüzgarlı) Entropi($T_{rüzgarlı}$) + P(yağmurlu) Entropi($T_{yağmurlu}$))

BK(T, Hava) = 1,571 – ((3/10).0,918 + (4/10).0,811 + (3/10).0,918)

BK(T, Hava = 1,571 – 0,2754 – 0,3244 – 0,2754

BK(T, Hava) = 0,70 olarak hesaplanır.

Diğer düğümler için de benzer hesaplamaları yapıyoruz:

BK(T, Ebeveyn) = ?

- Var = 5 (5 sinema)
- Yok = 5 (1 sinema, 2 tenis, 1 alışveriş, 1 evde)
- Entropi(T_{var})= $-\left(\frac{5}{5}\right)\log_2\frac{5}{5}=0$
- Entropi(T_{yok})= $-\left(\frac{1}{5}\right)\log_2\frac{1}{5}-\left(\frac{2}{5}\right)\log_2\frac{2}{5}-\left(\frac{1}{5}\right)\log_2\frac{1}{5}-\left(\frac{1}{5}\right)\log_2\frac{1}{5}=0{,}922$

BK(T, Ebeveyn) = entropi(T) – (P(var) Entropi(T_{var}) + P(yok) Entropi(T_{yok}))

BK(T, Ebeveyn) = 1,571 – ((5/10).0 + (5/10).0,922)

BK(T, Ebeveyn) = 1,571 – 0,961

BK(T, Ebeveyn) = 0,61 olarak hesaplanır.

BK(T, Para) = ?

- Çok = 7 (3 sinema, 2 tenis, 1 alışveriş, 1 evde)
- Az = 3 (3 sinema)
- Entropi($T_{çok}$)= $-\left(\frac{3}{7}\right)\log_2\frac{3}{7}-\left(\frac{2}{7}\right)\log_2\frac{2}{7}-\left(\frac{1}{7}\right)\log_2\frac{1}{7}-\left(\frac{1}{7}\right)\log_2\frac{1}{7}=1{,}842$
- Entropi(T_{az})= $-\left(\frac{3}{3}\right)\log_2\frac{3}{3}=0$

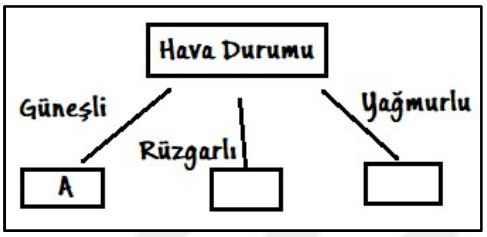
BK(T, Para) = entropi(T) – (P(çok) Entropi($T_{çok}$) + P(az) Entropi(T_{az}))

BK(T, Para) = 1,571 – ((7/10).1,842 + (3/10).0)

BK(T, Para) = 1,571 – 1,2894

BK(T, Para) = 0,2816 olarak hesaplanır.

BK(T, Hava) > BK(T, Ebeveyn) > BK(T, Para) olarak bulunmuş olur. *Hava Durumu* özelliği en büyük bilgi kazancını sağladığı için ağacın kökünde yer alacak özellik olarak seçilir. Bu özellik en ayırt edici özellik olarak bulunmuş olur. Bu özelliği ağacımızın köküne yerleştiriyoruz (Şekil 5.14). (Koç, 2016)



Şekil 5.14. DT – Kök belirleme (Koç, 2016)

Artık bundan sonra Şekil 5.14'de gösterilen ağacın çocuk düğümü olan *A* düğümüne ait alt veri kümesini belirliyoruz. Bu veri kümesi, Çizelge 5.5'te gösterilmektedir.

Çizelge 5.5. DT– A düğümüne ait alt veri kümesi (Koç, 2016)

Haftasonu	Hava	Ebeveyn	Para	Karar (Kategori)
H1	Güneşli	Var	Çok	Sinema
H2	Güneşli	Yok	Çok	Tenis
H10	Güneşli	Yok	Çok	Tenis

Her alt küme için tekrar bilgi kazancını hesaplayarak en ayırt edici özelliği belirliyoruz.

Bilgi kazançlarını hesaplıyoruz:

BK($T_{güneşli}$, Ebeveyn) = ?

- Var = 1 (1 sinema)
- Yok = 2 (2 tenis)
- Entropi(T_{var})= $-\left(\frac{1}{1}\right)\log_2\frac{1}{1}=0$
- Entropi(T_{yok})= $-\left(\frac{2}{2}\right)\log_2\frac{2}{2}=0$

$BK(_{Tgüneşli}$,Ebeveyn) = entropi($_{Tgüneşli}$) -

(P(var).Entropi(T_{var})+P(yok) .Entropi(T_{yok}))

$BK(_{Tgüneşli}$, Ebeveyn) = 0,918 – ((1/3).0 + (2/3).0)

$BK(_{Tgüneşli}$, Ebeveyn) = 0,918 olarak hesaplanır.

$BK(_{Tgüneşli}$, Para) = ?

- Çok = 3 (1 sinema, 2 tenis)
- Az = 0
- Entropi($T_{çok}$) = $-\left(\frac{1}{3}\right)\log_2\frac{1}{3} - \left(\frac{2}{3}\right)\log_2\frac{2}{3} = 0{,}918$
- Entropi(T_{az}) = 0

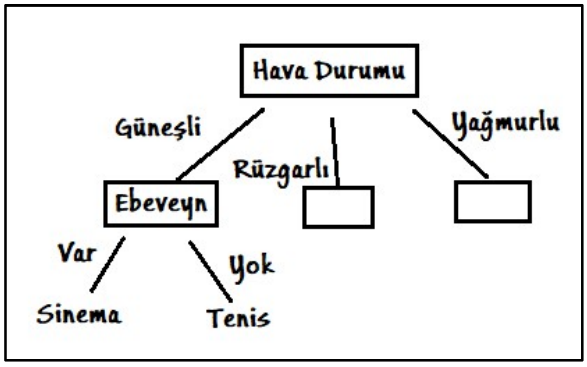
$BK(_{Tgüneşli}$, Para) = entropi($_{Tgüneşli}$) – (P(çok) Entropi($T_{çok}$) + P(az) Entropi(T_{az}))

$BK(_{Tgüneşli}$, Para) = 0,918– ((3/3).0,918+ (0/3).0)

$BK(_{Tgüneşli}$, Para) = 0,918 – 0,918

$BK(_{Tgüneşli}$, Para) = 0 olarak hesaplanır.

BK($_{Tgüneşli}$, Ebeveyn) > BK($_{Tgüneşli}$, Para) olduğundan yeni düğüm için en ayırt edici özellik *Ebeveyn* olarak belirlenmiştir. Ağacımızın yeni hali, Şekil 5.15'deki gibi olur.

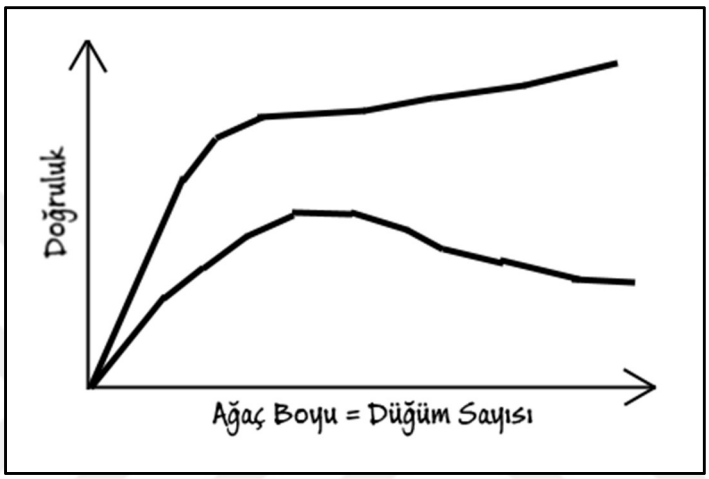


Şekil 5.15. DT– Ortaya çıkmış olan dal (Koç, 2016)

Yukarıda yapılan işlemler, her düğüm için aşağıdaki durumlardan biri oluşuncaya kadar tekrarlanır.

- Örneklerin hepsinin aynı sınıfa ait olması
- Örnekleri bölecek özellik kalmamış olması
- Kalan özelliklerin değerini taşıyan örneğin bulunmaması.

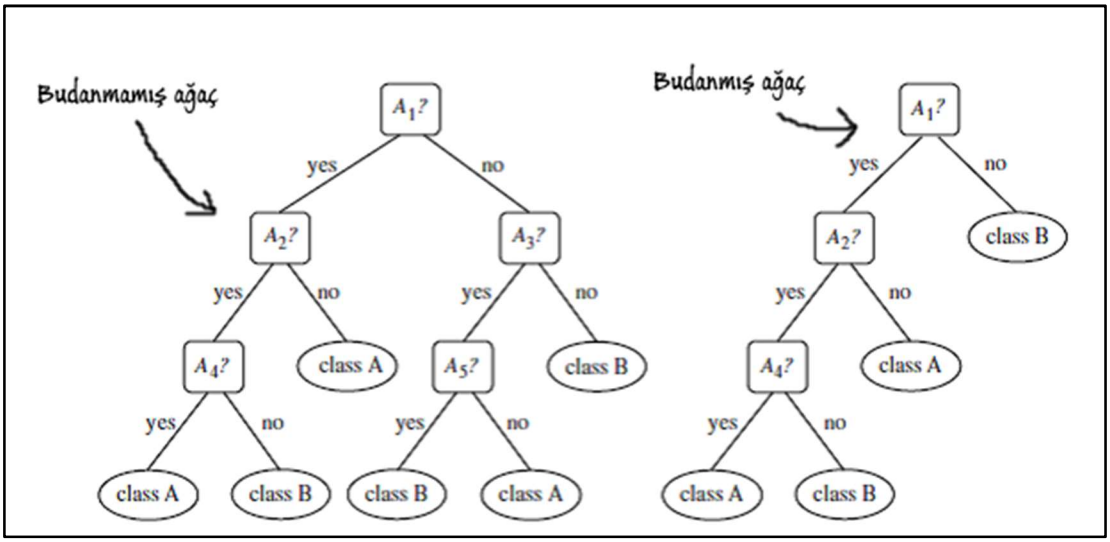
Eğitim verilerini en iyi sınıflandıracak şekle kadar her dal derinleştirildiği zaman, veriler arasında gürültü var ise, bu birtakım zorluklara sebep olabilir. Her yaprak *saf (pure)* olana dek bölmek, *veriyi ezberleyen (overfitting)* çok büyük bir karar ağacı oluşmasına sebep olabilir (Koç, 2016). Bu, istenmeyen bir durumdur çünkü overfitting durumunda başarım, zamanla azalmaya başlar. Bununla ilgili örnek grafik Şekil 5.16'da gösterilmiştir.



Şekil 5.16. Overfitting durumundaki bir DT başarımı (Koç, 2016)

Böyle bir durumda, ağaç doğru bir şekilde budanarak daha verimli hale getirilebilir. Doğru şekilde budamanın faydaları şöyle sıralanabilir:

- Budanmış ağaçlar daha kısa ve daha az karmaşık olma eğilimindedirler.
- Daha kolay anlaşılırlar
- Genellikle daha hızlıdırlar
- Test verilerini sınıflamada daha başarılıdırlar.



Şekil 5.17. Budanmamış ve budanmış ağaç (Koç, 2016)

Şekil 5.17'de budama işleminin nasıl yapıldığı, ağacın budamadan önce ve budamadan sonraki halleri ile örnek bir ağaç üzerinde gösterilmiştir.

5.2.5.3. Karar ağacının avantajları

- Karar ağacı oluşturmak zahmetsizdir.
- Küçük ağaçları hem bilgisayar programı ile hem de göz ile yorumlamak kolaydır.
- Anlaşılabilir kurallar oluşturulabilir.
- Ayrık nitelik değerleri için kullanılabilir.
- Yüksek doğrulukta sonuçlar üretirler.
- Kararlıdırlar.
- Doğrusal modellerin aksine, doğrusal olmayan ilişkileri de oldukça iyi eşleyebilirler.
- Sınıflandırma veya regresyon problemlerinin her ikisine de uyarlanabilirler.
- Çok çıktılı problemleri çözebilmektedir.
- İstatistiksel testler kullanılarak bir modelin doğrulanması mümkündür. (Koç, 2016)

5.2.5.4. Karar ağacının dezavantajları

- Sürekli nitelik değerlerini tahmin etmekte çok başarılı değillerdir.

- Sınıf sayısı fazla ve öğrenme kümesi örnekleri sayısı az olduğunda model oluşturmada çok başarılı değillerdir.
- Büyük öğrenme kümeleri için ağaç oluşturma karmaşıklığı fazladır.
- Büyük öğrenme kümeleri için ağaç budama karmaşıklığı fazladır. (Koç, 2016)

5.2.5.5. Karar ağacının uygulama alanları

- Belirli bir sınıfın olası üyesi olacak elemanları belirlenmesi
- Çeşitli vakaların, yüksek, orta, düşük risk grupları gibi kategorilere ayrılması
- Parametrik modellerin kurulmasında kullanılmak üzere çok sayıdaki değişkenden en önemlilerinin seçilmesi
- Gelecekte olayların tahmin edilebilmesi için kurallar oluşturulması
- Sadece belirli alt gruplara özgü ilişkilerin tanımlanması
- Kategorilerin birleştirilmesi ve sürekli değişkenlerin kesikli değişkenlere dönüştürülmesi tabanına dayanan problemlerin çözümü için tüm sahalarda kullanılmaktadır. (Albayrak ve Yılmaz, 2019)

5.2.6. Rastgele orman

Rastgele orman (*Radom Forest – RF*), ilk kez 2001 yılında Leo Breiman tarafından geliştirilmiştir. Breiman, 1996 yılında kendi geliştirdiği *bagging* yöntemi ile Kim Ho tarafından geliştirilen *random subspace* yöntemlerini birleştirmiştir. Amit ve Geman tarafından 1997 yılında tanımlanan, her düğüm için en iyi ayrımın rastgele bir seçim üzerinden belirlendiği bir çalışmadan da etkilenerek böyle bir yaklaşımı ortaya koymuştur.

Rastgele orman algoritması hem regresyon hem de sınıflandırma problemlerinde kullanılabilen gözetimli (supervised) bir öğrenme algoritmasıdır. Topluluk öğrenme yöntemi olan RF algoritması, sınıflandırma işlemi esnasında birden fazla DT üreterek sınıflandırma başarısını yükseltmeyi hedefler. Teker teker oluşturulan karar ağaçları bir araya gelerek karar ormanını oluşturur. Buradaki karar ağaçları, bağlı olduğu veri setinden rastgele seçilmiş birer alt kümedir. Ormandaki ağaç sayısı ve elde edilebilecek sonuç arasında doğrusal bir ilişki vardır. Ormandaki ağaç sayısı arttıkça daha kesin bir sonuç elde edilir.

Kök düğümü bulma ve düğümleri bölme işlemlerinin rastgele yapılıyor olması dışında DT algoritmaları ile tamamen aynıdır.

5.2.6.1. Rastgele ormanın yapısı

RF algoritması, bilinen ML yöntemleri içerisinde eşsiz bir tahmin geçerliliği ve model yorumlanabilirliği sağlar. Rasgele örneklemeyi ve topluluk yöntemlerdeki tekniklerin iyileştirilmiş özelliklerini içermesi nedeniyle RF yöntemi daha iyi genellemeler yapar ve daha geçerli tahminlerde bulunur. RF yöntemi ile yapılan tahminlerin daha kesin olmasının nedenleri, sapması düşük sonuçlar vermesi ve ağaçlar arasındaki düşük korelasyondur. Oldukça büyük ağaçların oluşturulması ile, daha düşük sapma miktarı elde edilebilir. Mümkün olduğu kadar birbirlerinden farklı ağaçlar oluşturarak da düşük korelasyon yapısında bir topluluk elde edilebilir. Birbirinden farklı olarak oluşturulan sınıflama ve regresyon karar ağaçları, bizi sonuca götürecek karar ormanı topluluğunu oluşturur. Karar ormanı oluşumu sırasında elde edilen sonuçlar bir araya getirilerek en son tahmin yapılır. (Fidancı, 2017)

RF yönteminde ağaçlar, seçilen *bootstrap* örneklemleri ve her düğüm ayrımında rastgele seçilen *m* adet tahminci ile oluşturulur. *m* adet tahmincinin toplam tahminci sayısından oldukça küçük olmasına dikkat edilir. Oluşturulan her bir karar ağacı, en geniş haliyle bırakılır ve budanmaz. Sınıflandırma için ağaçta her bir yaprak düğümü, sadece bir sınıfın üyelerini içerecek şekilde oluşturulur. Regresyon için ise, yaprak düğümde az sayıda birim kalana kadar ağaçlar bölünmeye devam edilir. (Fidancı, 2017)

RF algoritması, iki tane kullanıcı parametresi alır.

m: En iyi bölünmeyi sağlamak için her bir düğümde kullanılan değişken sayısı.

N: Geliştirilecek ağaç sayısı.

Algoritma uygulanırken, ilk olarak veri setinin 2/3'ünden önyükleme örnekleri oluşturulur. *Out of Bag (OOB)* verisi olarak da adlandırılan eğitim veri setinin 1/3'lük geri kalan kısmı da hataları test etmek için kullanılır. Her bir düğümde m değişkeni tüm değişkenler içerisinden rastgele seçilir ve bu değişkenler arasından en iyi dal belirlenir. *M* adet değişken sayısının kare köküne eşit olarak alınan *m* değişken sayısı genelde optimum sonuca en yakın sonucu verir. (Fidancı, 2017)

Oluşturulan sınıfın homojenliğini ölçmek için *Gini* indeksi hesaplanır. Gini indeksi ne kadar küçük olursa sınıfımız o kadar homojendir. Bir alt sınıfın Gini indeksi,

bir üst sınıfın Gini indeksinden daha az ise o dal başarılı bir şekilde oluşturulmuş anlamına gelir. (Fidancı, 2017)

GINI indeksini bulmak için (5.24) formülünden faydalanılır:

$$Gini(T) = 1 - \sum_{j=1}^{n}(p_j)^2 \qquad 5.24$$

Bu formülde (5.24) T, tüm veri setimizi, p_j, veri setindeki her bir düğümün kendisinden küçük ve kendisinden büyük eleman sayılarına bölümlerini, n ise seçilen verimizi ifade etmektedir.

Gini indeksi hesaplandıktan sonra, bu indeks kullanılarak test veri setinin sınıfları belirlenir. Çıkan sonuçları bütününde en iyi sınıflandırma yapılmış olur.

Örnek olarak kredi başvurusunda bulunan insanların bulunduğu bir veri seti üzerinde çalışarak, veri setindeki özellikler yardımı ile kredi verilecek ve kredi verilmeyecek şeklinde etiketleme yapan bir RF tasarlayalım. Çizelge 5.6'da kullanacağımız veri seti görülmektedir.

Çizelge 5.6. Rastgele orman örneğinde kullanılacak veri seti (Fidancı, 2017)

#	C	İş	Yaş	Eğitim	Malik	M.H.	Araba	Maaş	Kredi
1	E	Emekli	20-35	Lise	Ev Sah.	Evli	Var	<1500	Evet
2	K	İşsiz	35-45	Lise	Kiracı	Evli	Var	1500-3000	Evet
3	E	Çalışan	45-60	İlkokul	Kiracı	Bekar	Yok	3000>	Evet
4	K	Çalışan	60>	Lisans	Kiracı	Evli	Var	1500-3000	Hayır
5	K	İşsiz	45-60	Y.L.	Ev Sah.	Bekar	Yok	1500-3000	Hayır
6	E	İşsiz	45-60	Lise	Ev Sah.	Dul	Yok	3000>	Evet
7	E	Çalışan	35-45	İlkokul	Ev Sah.	Dul	Var	<1500	Evet
8	E	Emekli	35-45	İlkokul	Kiracı	Bekar	Var	1500-3000	Hayır
9	K	Emekli	20-35	İlkokul	Ev Sah.	Bekar	Var	3000>	Hayır
10	K	Emekli	35-45	Lise	Kiracı	Evli	Yok	1500-3000	Evet
11	E	Çalışan	45-60	Lisans	Kiracı	Bekar	Yok	1500-3000	Hayır
12	K	İşsiz	60>	Lisans	Ev Sah.	Evli	Yok	3000>	Evet
13	E	Çalışan	45-60	Y.L.	Ev Sah.	Bekar	Yok	<1500	Hayır
14	K	İşsiz	45-60	Lisans	Ev Sah.	Dul	Var	1500-3000	Hayır
15	K	Emekli	35-45	Lise	Kiracı	Dul	Var	3000>	Evet
16	K	Emekli	35-45	İlkokul	Ev Sah.	Bekar	Var	1500-3000	Evet
17	E	Çalışan	20-35	Lisans	Kiracı	Bekar	Yok	1500-3000	Hayır
18	E	Çalışan	35-45	T.L.	Kiracı	Bekar	Yok	3000>	Hayır

#	C	İş	Yaş	Eğitim	Malik	M.H.	Araba	Maaş	Kredi
19	K	İşsiz	45-60	Lise	Ev Sah.	Evli	Yok	<1500	Evet
…	…	…	…	…	…	…	…	…	…
9000	E	Çalışan	45-60	İlkokul	Kiracı	Dul	Var	3000>	Hayır

İlk iş olarak bu veri seti içerisinde oluşturacağımız karar ağacı sayısını belirliyoruz. Bu örneğimiz için 1200 tane karar ağacı seçtiğimizi varsayalım. Daha sonraki adımımız, her bir karar ağacı içerisinde yer alacak olan küme elemanlarının sayısını belirlemek olacaktır. Bunu da 3 tane olarak belirlemiş olalım.

Veri setimize bağlı kalarak 3 elemanlı 1200 adet karar ağacını rastgele seçiyoruz. Çizelge 5.7'de rastgele seçilmiş olan karar ağaçlarını görüyoruz.

Çizelge 5.7. RF - Rastgele seçilmiş orman (Fidancı, 2017)

#				Kredi
1	Kadın	İşsiz	45-60	Hayır
2	45-60	Ev Sahibi	Dul	Evet
3	Erkek	İlkokul	35-45	Evet
4	Erkek	Bekar	Bekar	Hayır
5	Kiracı	Ev Sahibi	Ev Sahibi	Hayır
6	Kadın	Emekli	Var	Evet
7	Erkek	Yok	Bekar	Hayır
8	Ev Sahibi	İşsiz	60>	Evet
…	…	…	…	…
1200	Kadın	İşsiz	45-60	Hayır

Her bir veri içi *random subspace* ile ağırlıklı ortalamasını hesaplıyoruz. Çizelge 5.8'de hesaplanmış olan ağırlıklı ortalamalar görülmektedir.

Çizelge 5.8. RF – Eğitici veri seti ağırlıklı ortalamaları (Fidancı, 2017)

$Ö_1$	$Ö_2$	$Ö_3$	Sınıf
0,4	0,33	0,05	$Sınıf_1$
0,12	0,66	0,4	$Sınıf_2$
0,2	0,2	0,12	$Sınıf_2$
0,3	0,1	0,39	$Sınıf_1$

Seçtiğimiz karar ağaçlarından 2/3'ü (800 tanesi) eğitici veri seti olarak kullanılır. Geri kalan 1/3'ü (400 tanesi) de test verisi olarak ayrılır. Formül (5.24)'teki ifade ile Gini indeksi hesaplanır. Çizelge 5.8'deki verileri formül (5.24)'te yerine koyarsak:

$$GI_1 = 1 - \left(\left(\frac{0,4}{0,78}\right)^2 + \left(\frac{0,33}{0,78}\right)^2 + \left(\frac{0,05}{0,78}\right)^2\right) = 0,55$$

Çizelge 5.9. RF – Test veri seti ağırlıklı ortalamaları (Fidancı, 2017)

$Ö_1$	$Ö_2$	$Ö_3$	Sınıf
0,8	0,75	0,25	$Sınıf_1$
0,13	0,33	0,5	$Sınıf_2$
0,66	0,65	0,7	$Sınıf_2$
0,25	0,32	0,22	$Sınıf_1$

Aynı işlemi Çizelge 5.9'da gösterilen test veri seti ağırlıklı ortalamaları için de yapalım:

$$GI_2 = 1 - \left(\left(\frac{0,8}{1,8}\right)^2 + \left(\frac{0,75}{1,8}\right)^2 + \left(\frac{0,25}{1,8}\right)^2\right) = 0,61$$

Gini indeksi en düşük çıkan satır, o ağaç için geçerli indeks değeridir. Bundan sonra son aşama olarak sınıflandırma işlemine geçilir. Ayırmış olduğumuz veri setimizin 1/3'lük kısmı olan test veri setimiz, 2/3'lük kısmı olan eğitim veri setimiz ile kıyaslanır. Kıyaslama işlemleri Gini indeksleri yardımı ile yapılır. Test veri seti değerleri, eğitici veri setinde kendi indeks değeri ile aynı olduğu sınıfa düşer. Yani indeks değerleri aynı olan veri setleri için sınıflandırma şu şekilde olur. Test veri setinin özellikleri (features) = Eğitici veri setinin sınıfı. (Fidancı, 2017)

5.2.6.2. Rastgele ormanın avantajları

- Mükemmel bir geçerlilik sunar. Pek çok veri seti için *adaboost* ve SVM'den daha kesin sonuçlar verir.
- Oldukça kısa sürede sonuç verir.
- Binlerce değişkene ve fazla sayıda sınıf etiketine sahip kategorik değişken içeren, kayıp verili veya dengesiz bir dağılım sergileyen veri setlerini kullanarak iyi sonuçlar verebilir.

- Girdi verilerinin ön hazırlık aşaması yoktur.
- Topluluğa ağaçlar eklendikçe, test setine ait hata tahmini, için yanlışlığı düşük sonuçlar vermeye başlar.
- Verilerin ölçeklendirilmesine gerek yoktur.
- Gürültülü verilerden arındırır.
- Son derece esnektir ve çok yüksek hassasiyete sahiptir.

5.2.6.3. Rastgele ormanın dezavantajları

- Yapıları biraz karmaşıktır.
- Karar ağaçlarına göre uygulanması biraz daha zor ve zaman alıcıdır.
- Hesaplama için daha fazla kaynak tüketir.
- Karar ağaçlarına göre daha az sezgiseldir.
- Bunun yanında DT algoritmalarının genelinde bulunan bir dezavantaj olarak, tahmin işlemi, diğer algoritmalardan daha fazla zaman alabilmektedir.

5.2.6.4. Rastgele ormanın uygulama alanları

- Astronomi
- Biyomedikal
- Kontrol sistemleri
- Finans analizi
- Sağlık
- Moleküler biyoloji
- Fizik
- Yazılım geliştirme. (Fidancı, 2017)

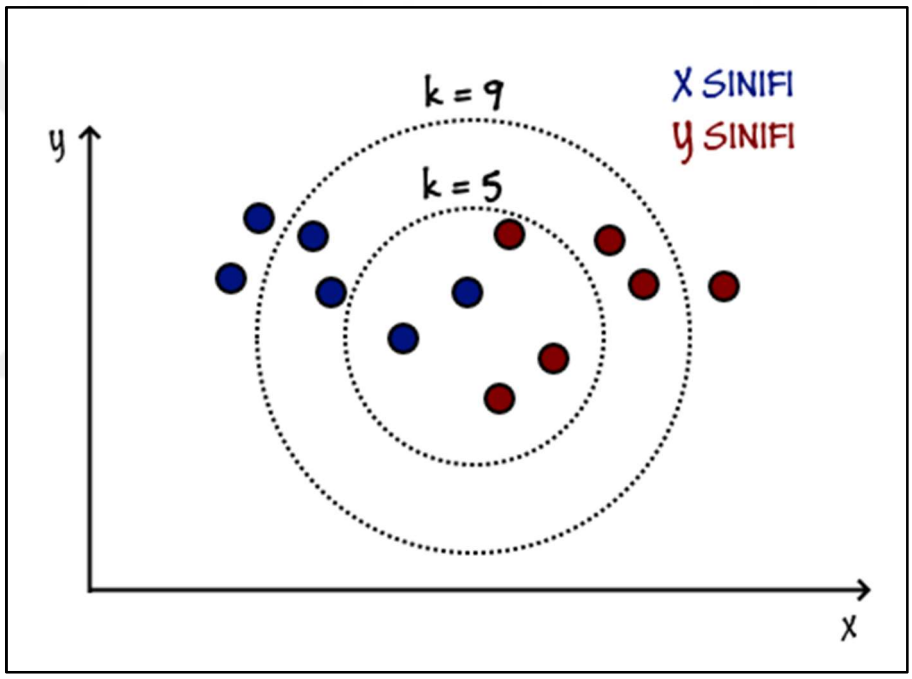
5.2.7. k-En yakın komşu

İngilizcedeki *k-Nearest Neighbour* kelimelerinin baş harfleri olan k-NN olarak anılır. Denetimli öğrenmede hem sınıflandırma ve hem de regresyon için kullanılan algoritmalardan biridir. En basit makine öğrenmesi algoritması olarak kabul edilir.

Diğer denetimli öğrenme algoritmalarının aksine, eğitim aşamasına sahip değildir. Eğitim ve test hemen hemen aynı şeydir. Tembel bir öğrenme türüdür. Bu nedenle, k-NN, geniş veri setini işlemek için gereken algoritma olarak ideal bir aday değildir.

5.2.7.1. k-NN'nin yapısı

k-NN ile temelde yeni noktaya en yakın noktalar aranır. K, bilinmeyen noktanın en yakın komşularının miktarını temsil eder. Sonuçları tahmin etmek için algoritmanın k miktarını (genellikle bir tek sayı) seçeriz. k-NN algoritmasının kabaca temel gösterimi Şekil 5.18'daki gibidir.



Şekil 5.18. k-NN algoritmasının temel gösterimi

Model tanımada, k-en yakın komşu algoritması (k-NN), sınıflandırma ve regresyon için kullanılan parametrik olmayan bir yöntemdir. Her iki durumda da girdi, özellik alanında k en yakın eğitim örneklerinden oluşur. Çıktı, k-NN'nin sınıflandırma için mi yoksa regresyon için mi kullanılacağına bağlıdır:

k-NN kullanılan sınıflandırma problemlerinde, çıktı sınıf üyeliği etiketidir. Bir nesne, kendisine en yakın komşularının çoğunluk oyuyla sınıflandırılır. Nesne, en yakın komşuları arasında en yaygın olan sınıfa ait olarak etiketlenir. Burada *k* ile bahsedilen, diğer noktaların işlenmekte olan noktaya olan uzaklığını ifade eder. Dolayısıyla negatif

bir değer olamaz. Ayrıca k, bir tam sayı olmak zorundadır. Eğer $k = 1$ ise, nesne basitçe o en yakın komşunun sınıfına atanır.

k-NN kullanılan regresyon problemlerinde, çıktı cismin özellik değeridir. Bu değer, en yakın komşularının değerlerinin ortalamasıdır.

k -NN, örüntü tabanlı öğrenme veya tembel öğrenme türüdür; burada işlev sadece yerel olarak yaklaştırılır ve tüm hesaplama, sınıflandırmaya kadar ertelenir. k-NN algoritması, tüm makine öğrenmesi algoritmalarının en basitleri arasındadır.

Hem sınıflandırma hem de regresyon için, komşuların katkılarına ağırlık koymak, böylece yakın komşuların ortalamaya daha uzak olanlardan daha fazla katkıda bulunmalarının daha yararlı olacağı temel anlayışı üzerine kurulmuştur. Örneğin, ortak bir ağırlıklandırma şeması, her komşuya $1 / d$ ağırlığının verilmesini öngörür; burada d komşuya olan uzaklıktır.

Komşular, sınıfın (k-NN sınıflaması için) veya nesne değerinin (k-NN regresyonu için) bilindiği diğer nesnelerden alınır. Bu, algoritma için yapılan bir eğitim olarak düşünülebilir, ancak diğer algoritmalardakine benzer direkt bir eğitim basamağı değildir ve böyle bir eğitim k-NN algoritmasında gerekli değildir.

k-NN algoritmasının bir diğer özelliği, verilerin yerel yapısına duyarlı olmasıdır. Algoritma, başka popüler bir makine öğrenme tekniği olan *k-means* ile karıştırılmamalıdır.

k komşuları, tüm mevcut vakaları depolayan ve bir benzerlik ölçüsüne (ör. Mesafe fonksiyonları) dayalı yeni vakaları sınıflandıran basit bir algoritmadır. k-NN, 1970'lerin başında halihazırda parametrik olmayan bir teknik olarak istatistiksel tahmin ve örüntü tanımada kullanılmıştır.

Her örnek, komşularının çoğunluk oyuyla sınıflandırılır. Bu olay, bir mesafe fonksiyonuyla ölçülen en yakın komşuları arasında en yakın olan sınıfa atanır. $k = 1$ ise, örnek yalnızca en yakın komşusunun sınıfına atanır. Mesafeyi ölçmede temel olarak üç temel fonksiyondan biri kullanılır.

$$\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2} \tag{5.25}$$

$$\sum_{i=1}^{k}|x_i - y_i| \tag{5.26}$$

$$\left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{1/q} \tag{5.27}$$

Bu formüller *Euclidean formülü* (5.25), *Manhattan formülü* (5.26) ve *Minkowski formülü* (5.27) olarak bilinir.

Bu üç mesafe formülünün de yalnızca sürekli değişkenler için geçerli olduğu unutulmamalıdır. Kategorik değişkenler söz konusu olduğunda, *Hamming mesafesi* (5.28) kullanılmalıdır. Ayrıca, veri kümesinde sayısal ve kategorik değişkenlerin bir karışımı olduğunda *0* ile *1* arasındaki sayısal değişkenlerin standardizasyonu meselesini ortaya çıkarmaktadır.

$$d_h = \sum_{i=1}^{k} |x_i - y_i| \qquad (5.28)$$

Hamming mesafesi ölçülürken, $x = y$ olduğunda $d_h = 0$, $x \neq y$ olduğunda $d_h = 1$ olarak hesaplanır. Bu durum Çizelge 5.10'de gösterilmektedir.

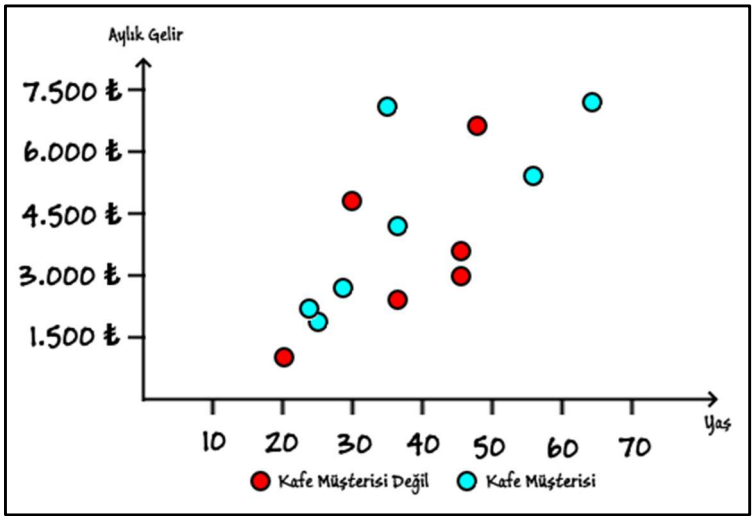
Çizelge 5.10. k-NN – x = y ve x ≠ y olma durumu

X	y	d_h
Kırmızı	Kırmızı	0
Kırmızı	Mavi	1

k'nın en uygun değerini seçmek için önce verileri incelemek gerekir. Büyük bir k değeri, genel gürültüyü düşürdüğü için daha hassas olacaktır ancak her zaman bu varsayım doğru olmaya da bilir. Çapraz doğrulama, k değerini doğrulamak için bağımsız bir veri kümesi kullanıp, geriye dönük olarak iyi bir k değeri belirlemenin başka bir yoludur. İstatistiksel olarak, çoğu veri kümesi için en uygun k değeri, 3 ile 10 arasında bulunmuştur.

Örnek olarak bir kafeye gelen devamlı müşterilerin gelir grubu ve yaş dağılımlarına bakılarak yeni bir örneğin, kafenin devamlı müşterisi olup olmayacağının tahmini yapılacaktır. Bu örnekte yaş ve aylık gelir olmak üzere iki sayısal değişken kullanılacak ve kafenin devamlı müşterisi olmak da hedef olarak araştırılacak sonuçtur.

Şekil 5.19'de örneğimize ait eğitim verilerinin dağılımı görünmektedir.



Şekil 5.19. k-NN – Kafe örneği örnek veri dağılımı

Grafikteki noktalar için euclidean formülünü (5.25) kullanarak *uzaklık (distance)* değerleri hesaplanır. Bu distance değerleri de bilinmeyen bir durumu (örneğimizde yaş = 40, aylık gelir = 5.000 ₺) sınıflandırmak için kullanılır. Çizelge 5.11'de örneğimiz için hesaplanmış olan distance değerleri dahil edilerek veri seti gösterilmektedir.

Çizelge 5.11. k-NN – Kafe örneği veri seti

Yaş	Aylık Gelir	Müşteri	$\Delta_{yaş}$	Δ_{gelir}	Distance
25	1.800 ₺	E	15	3.200	3.200,035
20	1.000 ₺	H	20	4.000	4.000,05
28	4.800 ₺	H	12	200	200,3597
45	3.500 ₺	H	5	1.500	1.500,008
28	2.800 ₺	E	12	2.200	2.200,033
22	2.200 ₺	E	18	2.800	2.800,058
63	7.200 ₺	E	23	2.200	2.200,12
55	5.500 ₺	E	15	500	500,2249
48	6.600 ₺	H	8	1.600	1.600,02
32	7.000 ₺	E	8	2.000	2.000,016
36	4.200 ₺	E	4	800	800,01
35	2.600 ₺	H	5	2.400	2.400,005
45	3.000 ₺	H	5	2.000	2.000,006
40	5.000 ₺	?			

Çizelge 5.11'de görüldüğü gibi, *Yaş* özelliğine baktığımız zaman en yakın komşu 36, *Aylık gelir* özelliğine baktığımız zaman da en yakın komşu 4.800 olarak görülüyor. Euclidean formülü (5.25) ile en yakın komşu hesaplandığı zaman da bulunan değerlerin tamamının *Aylık gelir* özelliği farkı ile çok yakın değerler olduğu görülüyor. Bunun sebebi de özelliklerin birimlerinden kaynaklanan büyüklük farklarıdır. Bu durum, doğrudan eğitim veri seti üzerinde yapılan uzaklık hesaplamalarında her zaman bir handikap olarak ortaya çıkar ve daha yüksek rakamlarla ifade edilen büyüklükler her zaman uzaklığa daha fazla etki eder. Bu durumu ortadan kaldırmak için, özellikler arasındaki mesafelerin belirli yöntemler kullanarak standartlaştırılması gerekir.

$$X_s = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{5.29}$$

Formül (5.29)'da standardizasyon için kullanılan yöntemlerden biri gösterilmiştir. Örneğimizdeki değerleri formül (5.29)'da yerine koyduğumuz zaman hesaplanan değerler ve Euclidean formülü (5.25)'nde bu değerler kullanılarak hesaplanan distance değerleri dahil edilirse, Çizelge 5.12'deki gibi bir tablo oluşur.

Çizelge 5.12. k-NN – Standartlaştırılmış kafe örneği veri seti

Yaş	$Yaş_s$	Aylık Gelir	$Gelir_s$	Müşteri	$\Delta_{yaş}$	Δ_{gelir}	Distance
25	0,116279	1.800 ₺	0,129032	E	0,348837	0,516129	0,622958
20	0	1.000 ₺	0	H	0,465116	0,645161	0,79534
28	0,186047	4.800 ₺	0,612903	H	0,27907	0,032258	0,280928
45	0,581395	3.500 ₺	0,403226	H	0,116279	0,241935	0,268428
28	0,186047	2.800 ₺	0,290323	E	0,27907	0,354839	0,451432
22	0,046512	2.200 ₺	0,193548	E	0,418605	0,451613	0,615779
63	1	7.200 ₺	1	E	0,534884	0,354839	0,641881
55	0,813953	5.500 ₺	0,725806	E	0,348837	0,080645	0,358038
48	0,651163	6.600 ₺	0,903226	H	0,186047	0,258065	0,318136
32	0,27907	7.000 ₺	0,967742	E	0,186047	0,322581	0,372386
36	0,372093	4.200 ₺	0,516129	E	0,093023	0,129032	0,159068
35	0,348837	2.600 ₺	0,258065	H	0,116279	0,387097	0,404184

Yaş	$Yaş_s$	Aylık Gelir	$Gelir_s$	Müşteri	$\Delta_{yaş}$	Δ_{gelir}	Distance
45	0,581395	3.000 ₺	0,322581	H	0,116279	0,322581	0,342898
40	0,465116	5.000 ₺	0,645161	?			

Çizelge 5.12'de de görüldüğü gibi, standartlaştırılmış veriler üzerinde yapılan uzaklık hesaplaması sonucu, değerini tahmin etmeye çalıştığımız 40 yaş ve 5.000 ₺ gelire sahip örneğin veri setindeki en yakın komşusu 36 yaş ve 4.200 ₺ gelire sahip örnek olduğu tespit edildi. Bu durumda tahmin etmeye çalıştığımız değer, en yakın komşusunun değeri ile aynı olmalıdır. Yani sonuç "*E*" olmalıdır.

5.2.7.2. k-NN'nin avantajları

- Temel problemlerin çözümünde kullanılabilecek en basit algoritmadır.
- Gürültülü veri setine karşı dayanıklıdır.
- Geniş eğitim seti üzerinde eğitilirse daha doğru sonuçlar verebilir.
- k-NN oldukça sezgisel ve basittir. k-NN algoritmasının anlaşılması ve uygulanması oldukça kolaydır. Yeni veri noktasını sınıflandırmak için k'nın en yakın komşularını bulmak için tüm veri setini okur.
- Varsayımı yoktur.
- Eğitim verisi yoktur. Çıkarımda bulunmak için sistemin bir eğitimden geçirilmesine gerek yoktur. Sadece eğitim verisi olarak adlandırılabilecek veri seti içerisinde arama yapar.
- Sürekli evrim geçirir. Her hesaplanan değer, bir sonraki hesaplanacak olan değer için bir giriş verisi olur.
- Çok sınıflı problemler üzerinde de kullanılabilir.
- Hem sınıflandırma hem de regresyon için kullanılabilir.
- Mesafe ölçümü yapacak olan tekniğin kullanımı anlamında esneklik sağlar.

5.2.7.3. k-NN'nin dezavantajları

Yukarıda bahsedilen avantajların bazıları aynı zamanda dezavantaj olarak da alınabilir. Örneğin sürekli evrim geçirmesi tutarlılık açısından avantajdır, ancak bu

durum, her seferinde yeniden hesaplama gerekliliği doğuracağı için aşırı bir yavaşlık oluşturacağından hız açısından dezavantaj olarak ele alınabilir. Aynı şekilde eğitim aşamasının olmaması tasarım açısından bir avantaj olarak görülebilir ancak her seferinde aynı verinin içerisinde arama yapacağından, yine hız açısından dezavantaj olarak görülebilir. Hem mesafe ölçümü hem de standartlaştırma için kullanılabilecek yöntemlerin çeşitliliği esneklik açısından olumlu bir durum gibi görünüyor olsa da, yapılacak olan tutarlılık açısından bir dezavantajdır. Algoritmanın dezavantajlarını sıralayacak olursak:

- Çok yavaş bir algoritmadır
- Boyut çıkmazı: Küçük boyutlu veri setlerinde hız kabul edilebilir olsa da başarı düşüktür, veri boyutu büyüdükçe başarı yükselir ancak bu kez de hız çok fazla düşer.
- Standardizasyon gerekliliği
- Mesafe hesaplama yöntemleri arasındaki farklılıklar
- Eksik veri ile başa çıkmada başarılı değildir.

5.2.7.4. k-NN'nin uygulama alanları

- k-NN algoritması genellikle arama yapmak için kullanılır. k-NN sınıflandırıcısı demek yerine k-NN arayıcısı bile denilebilir. Genellikle "*şu özelliğe en yakın özellikteki ögeyi bul*" şeklinde tanımlanabilen problemlerin çözümlerinde kullanılır.
- Anlamsal olarak benzer belgelerin tespitinde ve sınıflandırılmasında kullanılır.
- Tavsiye sistemlerinde sıklıkla kullanılır ("*bu ürünü alanlar şunlara da baktı*" gibi veya "*bunu mu demek istediniz?*" gibi, veya e-ticaret sitelerinde kullanıcıya gösterilecek olan reklamların seçimi gibi).
- Tıp alanında belirtilerin en fazla benzerlik gösterdiği hastalığın teşhisi uygulamalarında.

6. UYGULAMA VE SONUÇLARI

Çalışmamızı iki kategoriye ayırarak bunları ayrı ayrı inceleyebiliriz. Birinci kategori, hisse senetleri hakkında atılan tivitlerin etiketlenerek makineye öğretilmesi, farklı sınıflandırma algoritmaları kullanılarak makinenenin eğitilmesi ve bu algoritmalar ile test için kullanılan verilerin sınıflandırılmasıdır. İkinci aşama ise, birinci aşamadan tamamen bağımsız olarak, etiketlenmiş veri seti ile borsa yönünün tahmin edilmesine dayanmaktadır. Bu aşama için herhangi bir uygulamaya ihtiyaç duyulmamış, excel programı kullanılarak grafik olarak sunulmuştur.

6.1. Sınıflandırma

Bu başlık altında, önceden el ile etiketlenmiş olan veri seti ile altı farklı algoritmanın eğitilmesi ve cross-validation ile bu algoritmalarının test edilmesi, veri setimiz üzerinde bu algoritmaların başarı puanlarının tespit edilmesi işlemleri anlatılacaktır.

Sınıflandırma işlemi kendi yazdığımız uygulama ile gerçekleştirilmiştir. Uygulamamız, *Jupyter Notebook* geliştirme aracını kullanarak, *Python* dili ile gerçekleştirildi. Yaptığımız uygulama, önceki bölümlerde bahsedilen 10 adet hisse senedi için ve 6 ayrı sınıflandırma algoritması kullanılarak, TF-IDF özellik seçimi yöntemi kullanılarak gerçekleştirilmiştir. Bu başlık altında, yaptığımız uygulama ile ilgili örnek olarak sadece $AAPL (Apple hisse senedi) için atılmış olan tivitler üzerinde ve sadece NB algoritması ile yapılan sınıflandırma anlatılacaktır. Diğer algoritmalar ve diğer hisse senetleri de benzer şekilde yapılmıştır. Sonuçları da yine bu başlık altında tartışılacaktır.

6.1.1 Verilerin temizlenmesi

Uygulama geliştirilmeye başlamandan önce, twitter web arayüzünü kullanarak daha önceden indirmiş olan tivitler,, “,” karakteri ile ayrılacak şekilde “*.csv*” uzantılı *excel* dosyası olarak kaydedildi. Uygulamadaki ilk adım olarak da bu tivitler, panda kütüphanesi kullanılarak okundu. @AAPL tivitleri için dosyadan okunan ilk 5 tivit Şekil 6.1’de listelenmiştir.

	Tweet	Sınıf
0	FX Leadership http://dlvr.it/R1zrq3 $AAPL...	2
1	Get ready for the #Olympics #Tokyo2020 $SNE #S...	2
2	Report details Eddy Cue's failed negotiations ...	2
3	Get ready for the #Olympics #Tokyo2020 $SNE #S...	2
4	Imagine reading up on *Oil*AAPL $EURUSD concu...	2

(28649, 2)

Şekil 6.1. Dosyadan okunan ham tivitler

Twitter verileri, kullanıcıların denetimsiz olarak yazdıkları tivitleri içerdiği, aynı zamanda sosyal medyanın kendine has birtakım karakterlerinin barındırdığı için kendi içerisinde dahi bir standardı yoktur. Bunu sağlamak için de tivitleri işlemeye başlamadan önce bir ön temizleme işlemi gerekir. Temizleme işlemine öncelikle birebir aynı olan tivitleri silmek ile başlandı. Bunu yapmak için ".csv" dosyasından okunarak doldurulan dizinin "drop_duplicates()" fonksiyonu çağrıldı. Daha önce 28.649 olan tivit sayısı, bu işlem sonrasında 23.812'ye düşmüştür., Sonrasında elde kalan bu tekrarsız tivitlerin içersinde geçen, alfanumerik olmayan tüm karakterler silindi. *Tab* karakteri silindi, "[/(){}\[\]\|@,;]" karakterleri, *boşluk* karakteri ile değiştirildi. n*ltk* kütüphanesinin corpus sub-domaininden temin edilen *İngilizce* dili için "*stopwords*" listesi içerisinde bulunan tüm kelimeler de silindi. Bu listede, metin sınıflandırma için gerekli olmayan (bağlaçlar, ikilemeler v.b.) kelimeler bulunmaktadır. "*RT*" ile başlayan tivitler, daha önce başka bir tivit olarak ele alındığı için, bu şekilde belirtilmiş olan tivitler de silinerek göz ardı edildi. *Hashtag (#)* karakteri de bizim için anlamsız olduğu için tivitin içerisinden silindi. "@" karakteri ile başlayan kelimeler de metin sınıflandırma için *varlığı* ifade etmektedir. Her tivit zaten varlık ismi verilerek çekildiği için, sınıflandırmasını yapacağımız varlığı zaten bildiğimizden dolayı, bu kelimeler de bizim için gereksiz olarak kabul edildi ve silindi. Ayrıca tivit içerisinde geçen web adresleri ve linklerin de tivitin duygusuna bir katkısı olmadığından dolayı onlar da silindi. Son olarak da standart sağlamak amacıyla, tüm tivitler, tamamı küçük harf olacak şekilde değiştirildi. Bu işlemler, *sub()* ve *replace()* fonksiyonları kullanılarak yapıldı. Temizleme işlemi bittikten sonra veri setinde 21.265 adet tivit kaldı. Şekil 6.2'de temizlenmiş tivit örnekleri görülmektedir.

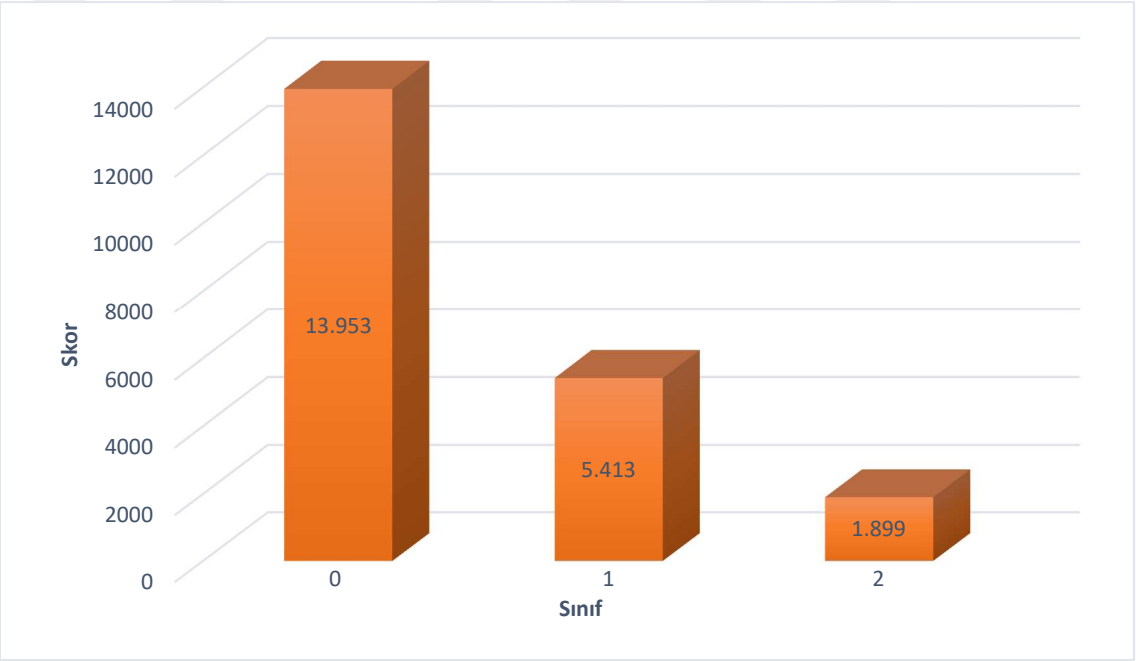
	Tweet	Sınıf
632	join robinhoodapp well get stock like aapl f f...	1
633	get globe mail subscription read peter miseks ...	1
634	handson apples new inch k core imac qtsnbsp aa...	1
635	traders sell shares apple aapl strength aapl nbsp	1
636	need know apples beastly x chip themotleyfool ...	1
637	dow aapl amzn googl nvda fb tsla twtr sq nflx ...	1

(21265, 2)

Şekil 6.2. Temizleme işleminden geçirilmiş olan tivitler

6.1.2. Vektörlerin Hazırlanması

matplotlib kütüphanesinin *pyplot* sınıfı fonksiyonları kullanılarak oluşturulan, eğitim veri setindeki tivitlerin görsel olarak dağılımını grafik, Şekil 6.3'te gösterilmiştir.



Şekil 6.3. Veri setinde bulunan tivitlerin sınıf dağılımları

Şekil 6.3'te de görüldüğü gibi, eğitim veri setindeki pozitif tivitlerin sayısı,*Sklearn* kütüphanesinin *feature_extraction.text* domainindeki TfidVectorizer sınıf fonksiyonları kullanılarak tüm kelimler ayrılıp harf sırasına göre *(tivit) X (veri setindeki tüm kelimeler)* olacak şekilde iki boyutlu (21.265 x 35.739) bir vektör haline dönüştürüldü. Şekil 6.4'te bu vektör gösterilmektedir.

```
array([[0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 3., 0., ..., 0., 0., 0.],
       ...,
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.]])
```

Şekil 6.4. Kelime frekans matrisi

Bu matristeki her bir satır, eğitim veri setindeki her bir tivite karşılık gelmektedir. Matrisin sütunları ise eğitim veri setindeki kullanılmış olan her bir kelimeye karşılık gelmektedir. Matris, kelimelerin tivit içerisindeki frekans değerlerinden oluşmaktadır. Yani her bir kelimenin her bir tivit içerisindeki kullanılma sayılarını göstermektedir. Elimizdeki bu matrisi, giriş ve çıkış olarak kullanılmak üzere iki farklı vektör haline dönüştürdük. Bu işlem için *sklearn* kütüphanesinin *model_selection* domainindeki *train_test_split()* sınıf fonksiyonu kullanıldı. Bu fonksiyon genel olarak eğitim ve test verileri setlerinin ayrılmasında kullanılmaktadır. Ancak biz coross validation yöntemi ile doğrulama yapacağımızdan dolayı bu fonksiyonu 100 parametresi ile çağırdık ve veri setinin tamamı test veri setine aktarılacak şekilde kullandık. Şekil 6.5'de veri seti içerisinden rastgele olarak seçilmiş eğitim ve test veri seti vektörleri ve bu vektörlerin sınıf-etiket dağılımları gösterilmektedir.

```
Eğitim Veri Seti Giriş Vektörü:   (21265, 35739)
Test Veri Seti Giriş Vektörü  :   (21265, 35739)
Eğitim Veri Seti Çıkış Vektörü:   (21265, )
Test Veri Seti Çıkış Vektörü  :   (21265, )

Eğitim Veri Seti Sınıf Etiketleri:
1    5413
0    13953
2    1899
Name: Class, dtype: int64

Test Veri Seti Sınıf Etiketleri:
1    5413
0    13953
2    1899
Name: Class, dtype: int64
```

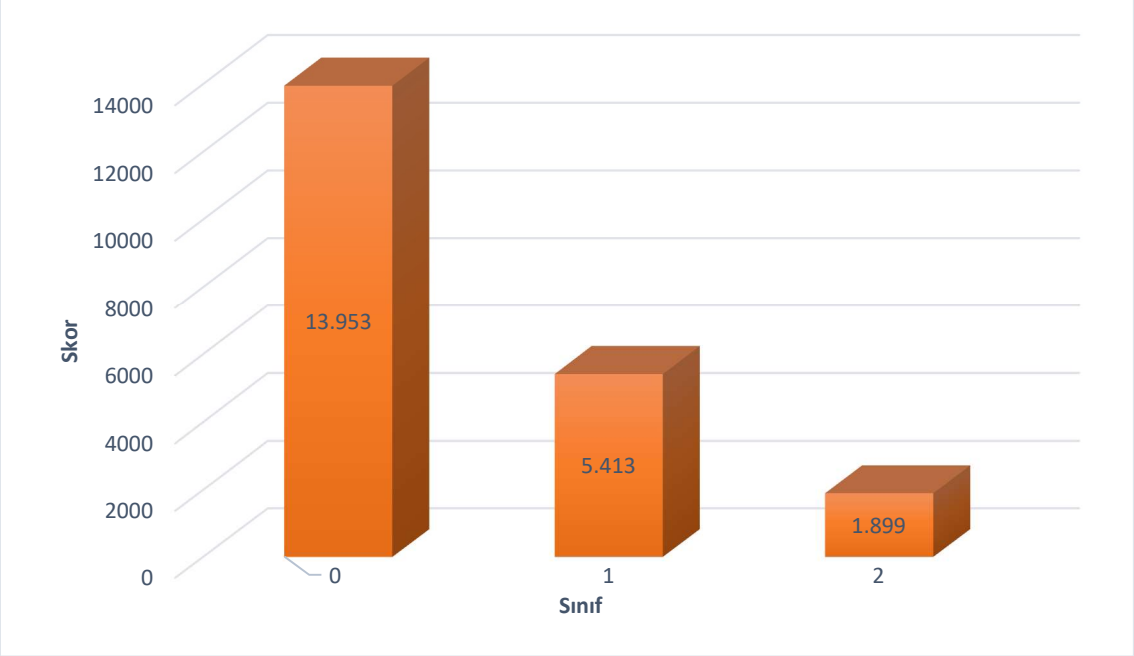
Şekil 6.5. Eğitim ve test veri setleri giriş-çıkış vektörleri

Kontrol amaçlı olarak kullanılacak olan test veri seti çıkış vektörünün görünümü Şekil 6.6'da görüldüğü gibidir.

Satır Numarası	Etiket
11896	0
3088	0
15837	1
2318	0
6897	2

Şekil 6.6. Test veri seti çıkış vektörü

Bu vektör, içerisinde tuttuğu tivitlerin, veri setindeki satır numaraları ve sınıf etiketlerinden oluşmaktadır. Bu çıkış vektörünün sınıf etiket dağılımları da Şekil 6.7'de görülmektedir.



Şekil 6.7. Test verisi çıkış vektöründeki sınıf etiket dağılımı

Test verisi çıkış vektöründeki pozitif tivitlerin sayısı 5.413, negatif tivitlerin sayısı 1.899 ve nötr tivitlerin sayısı is 13.953 olarak görülmektedir.

6.1.3. Eğitim ve Sınıflandırma

Veri setindeki tivitler temizlenip sayısal vektörler haline dönüştürüldükten sonra, eğitim ve sınıflandırma için hazır hale gelmiş oldu. Hazır halde bulunan bu veriler üzerinde tek tek sınıflandırma algoritmaları uygulandı. Algoritmaların başarımını homojen hale getirebilmek için *"cross validation"* metodu kullanıldı. Bu metodda veriler belli bir parçaya ayrılır. Bir parçası doğrulama amaçlı kullanılırken diğer parçalar eğitim amaçlı kullanılır. Doğrulama amaçlı olarak kullanılan parça her defasında bir kaydırılarak tekrar çıkarım yapılır. Tüm parçalar bittikten sonra bulunan sonuçların ortalaması alınır. Bu şekilde heterojen olarak dağılmış veri seti homojen olarak test edilmiş olur. Bu işlem için *sklearn* kütüphanesinin *model_selection* domaininde bulunan *cross_val_score()* sınıf fonksiyonu kullanılmıştır. Veriler 10 parçaya ayrılarak tek tek her parça doğrulama işlemi için kullanılarak eğitim ve sınıflandırma işlemi gerçekleştirilmiştir.

Sınıflandırma işlemine öncelikle naive bayes ile başlandı. NB sınıflandırma yapmak için yine *sklearn* kütüphanesinin *naive_bayes* domaininden MultinomialNB sınıfı kullanıldı. NB ile sınıflandırılmış örnek veri Şekil 6.8'de gösterilmiştir.

Satır İndeksi	Etiket
0	0
1	1
2	1
3	2
4	1

Şekil 6.8. NB ile sınıflandırılmış veri

NB algoritması kullanılarak yapılan sınıflandırma sonrasında oluşan *confusion matrix* Çizelge 6.1'deki gibidir.

Çizelge 6.1. NB sınıflandırma sonucu oluşan confusion matrix

Gerçek Değer				
	0	13.777	176	0
	1	4.239	1.173	1
	2	1.830	65	4
		0	1	2
		Tahmin Edilen		

Sınıflandırma işlemi sonunda NB algoritmasının başarımı 0,703 (%70,3) olarak gerçekleştiği görüldü.

Sınıflandırma işlemine rastgele orman algoritması ile devam edildi. RF sınıflandırma yapmak için yine *sklearn* kütüphanesinin *RandomForestClassifier* metodu kullanıldı.

RF algoritması kullanılarak yapılan sınıflandırma sonrasında oluşan *confusion matrix* Çizelge 6.2'deki gibidir.

Çizelge 6.2. RF sınıflandırma sonucu oluşan confusion matrix

Gerçek Değer				
	0	13.488	380	85
	1	2.422	2.943	48
	2	1.367	152	380
		0	1	2
		Tahmin Edilen		

Sınıflandırma işlemi sonunda RF algoritmasının başarımı 0,791 (%79,1) olarak gerçekleştiği görüldü.

Sınıflandırma işlemine destek vektör makinesi algoritması ile devam edildi. SVM sınıflandırma yapmak için yine *sklearn* kütüphanesinin *svm* domaindeki *SVC()* sınıf fonksiyonu kullanıldı.

SVM algoritması kullanılarak yapılan sınıflandırma sonrasında oluşan *confusion matrix* Çizelge 6.3'teki gibidir.

Çizelge 6.3. SVM sınıflandırma sonucu oluşan confusion matrix

Gerçek Değer			
0	13.953	0	0
1	5.413	0	0
2	1.899	0	0
	0	1	2
	Tahmin Edilen		

Sınıflandırma işlemi sonunda SVM algoritmasının başarımı 0,656 (%65,6) olarak gerçekleştiği görüldü.

Sınıflandırma işlemine karar ağacı algoritması ile devam edildi. DT sınıflandırma yapmak için yine *sklearn* kütüphanesinin *tree* domainindeki *DecisionTreeClassifier()* sınıf fonksiyonu kullanıldı.

DT algoritması kullanılarak yapılan sınıflandırma sonrasında oluşan *confusion matrix* Çizelge 6.4'deki gibidir.

Çizelge 6.4. DT sınıflandırma sonucu oluşan confusion matrix

Gerçek Değer			
0	12.201	1.249	503
1	1.940	3.270	203
2	946	252	701
	0	1	2
	Tahmin Edilen		

Sınıflandırma işlemi sonunda DT algoritmasının başarımı 0,760 (%76) olarak gerçekleştiği görüldü.

Sınıflandırma işlemine k-en yakın komşu algoritması ile devam edildi. k-NN sınıflandırma yapmak için yine *sklearn* kütüphanesinin *neighbors* domainindeki *KNeighborsClassifier()* sınıf fonksiyonu kullanıldı.

k-NN algoritması kullanılarak yapılan sınıflandırma sonrasında oluşan *confusion matrix* Çizelge 6.5'deki gibidir.

Çizelge 6.5. k-NN sınıflandırma sonucu oluşan confusion matrix

Gerçek Değer				
	0	12.886	936	131
	1	3.355	1.992	66
	2	1.425	224	250
		0	1	2
		Tahmin Edilen		

Sınıflandırma işlemi sonunda k-NN algoritmasının başarımı 0,711 (%71,1) olarak gerçekleştiği görüldü.

Sınıflandırma işlemine yapay sinir ağı algoritması ile devam edildi. ANN sınıflandırma yapmak için yine *sklearn* kütüphanesinin *neural_network* domainindeki *MLPClassifier()* sınıf fonksiyonu kullanıldı

ANN algoritması kullanılarak yapılan sınıflandırma sonrasında oluşan *confusion matrix* Çizelge 6.2'deki gibidir.

Çizelge 6.6. ANN sınıflandırma sonucu oluşan confusion matrix

Gerçek Değer				
	0	11.828	1.318	807
	1	1.855	3.187	371
	2	1.155	299	445
		0	1	2
		Tahmin Edilen		

Sınıflandırma işlemi sonunda ANN algoritmasının başarımı 0,727 (%72,7) olarak gerçekleştiği görüldü.

AAPL hisse senedi için atılan tivitlerin 3'lü sınıflandırılmasında en iyi sonucu %79,1 ile rastgele orman algoritması vermiştir. RF algoritmasını sırası ile %76 ile karar ağacı, %72,7 ile yapay sinir ağı, %71,1 ile k-en yakın komşu, %70,3 ile naive bayes ve %65,6 ile destek vektör makinesi algoritmaları takip etmiştir.

6.2. Yön Tahmini

Bu başlık altında, el ile etiketlenmiş olan veri setimiz ile, borsanın yönü arasında bir ilişki olup olmadığı araştırılacaktır.

Çalışmamızda kullanılan corpus oluşturulurken, el ile verilmiş olan etiketler kullanılmıştır. Bu etiketlerin verilmesinde de, veri setinin büyüklüğü göz önünde bulundurularak, yaklaşık 120.000 adet tivit, yaklaşık 1.000'er tane tivit içerecek şekilde 120 adet dosyaya bölünmüş, bu dosyalar da iyi seviyede İngilizce bilgisi olan, ancak borsa bilgisi faklılık gösteren 75 katılımcı arasında paylaştırılmıştır. Bazı katılımcılar 4, bazıları 3, bazıları 2 ve geneli 1'er dosyayı etiketleyerek corupusun oluşturulmasına katkıda bulunmuşlardır.

Katılımcılardan gelen etiketlenmiş dosyaların içeriği kontrol edildiğinde, farklı katılımcılar tarafından yapılan etiketlemelerin aynı standartlarda olmadığı hemen göze çarpmaktadır.

Corpus oluşturulurken başa çıkılan bir başka problem ise, düzensiz cümlelerden, linklerden, reklamlardan v.b. oluşan tivitlerin çokluğu oldu. Tüm katılımcılara, el ile etiketleme sırasında bu tür tivitlerin, ayrıca herhangi bir yorum, tahmin v.b. belirtmemiş olan tivitlerin ve anlaşılmayan cümlelerden oluşan tivitlerin tamamının nötr olarak etiketlenmesi istenmiştir.

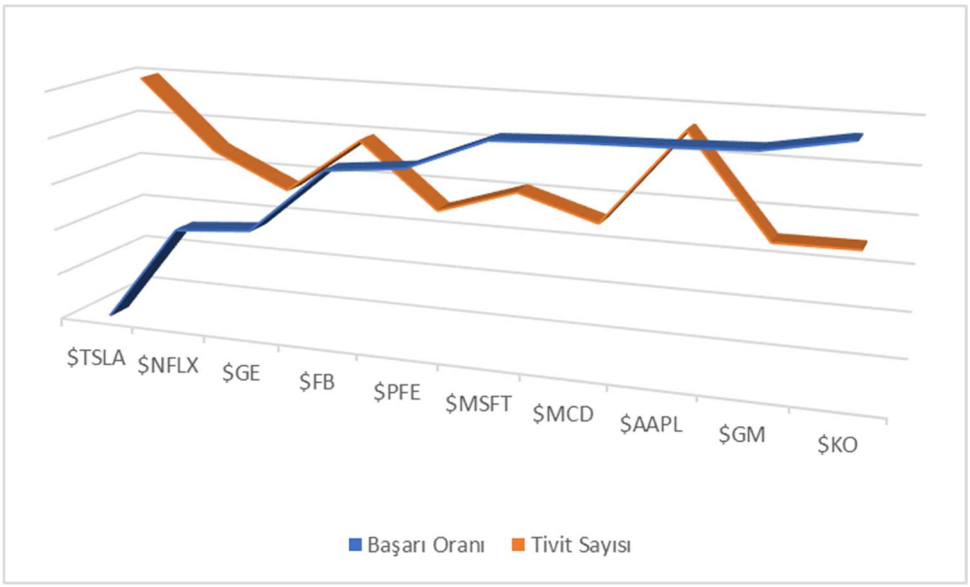
Son olarak da tahmin edilen değer ile hisse senedinin hareket yönü birbiri ile karşılaştırarak hisse hisse tahmin doğruluğu oranları çıkarılmıştır. Bunu yaparken, yukarıda bahsedilen zorluklardan dolayı, nötr sınıfını aynı zamanda bir çöplük gibi kullanmamız sebebi ile, sadece pozitif ve negatif sınıfları dikkate aldık. Tahmin doğruluğunu hesaplarken de ters yönlü hareketler başarısız, diğer veriler ise başarılı tahmin olarak alınmıştır. Örneğin Facebook hisse senedi için pozitif olarak etiketlenmiş bir tivit, hisse senedi ertesi gün %1'den daha fazla düşmemiş ise başarılı bir tahmindir. Nötr olarak kalmış olsa da pozitif yönde hareket etmiş olsa da bu tahmini doğru olarak aldık. Aynı şekilde, Tesla için negatif olarak etiketlenmiş bir tivit, hisse senedi ertesi gün %1'den daha fazla yükselmemiş ise başarılı bir tahmindir. Nötr olarak kalmış olsa da negatif yönde hareket etmiş olsa da bu tahmini doğru olarak aldık. Sadece, pozitif olarak etiketlediğimiz bir tivit, ertesi gün ilgili hisse senedi negatif yönlü bir hareket yapmış ise başarısızlık, aynı şekilde negatif olarak etiketlediğimiz bir tivit, ertesi gün ilgili hisse senedi pozitif yönlü bir hareket yapmış ise başarısızlık olarak alınmıştır.

Bu kriterlere göre her bir hisse senedi için yön tahmin başarı oranları Çizelge 6.7'de verilmiştir.

Çizelge 6.7. Hisse senetlerinin yön tahmin başarı oranları

Hisse Senedi	Başarı Oranı
$TSLA	66,67%
$NFLX	79,12%
$GE	79,96%
$FB	88,42%
$PFE	89,47%
$MSFT	93,56%
$MCD	94,02%
$AAPL	94,25%
$GM	94,61%
$KO	96,46%

Çizelge 6.7'yi incelediğimizde bir şey hemen dikkatimizi çekiyor: Hisse senetlerinin yön tahmin başarıları, Çizelge 4.1'de verilen hisse senetleri için atılan tivit sayıları ile ters orantılı olarak seyretmektedir. Bu iki çizelge birleştirilerek grafik haline dönüştürülmüş ve Şekil 6.9'da gösterilmiştir.



Şekil 6.9. Hisse başına atılan tivit ve yön tahmin başarıları

Bu grafiğin oluşmasında özellikle CocaCola ($KO) gibi az sayıda tivit atılmış olan hisse senetlerinin, genellikle tek bir katılımcı tarafından etiketlenerek bir etiket

standardına sahip olması, Tesla ($TSLA) gibi çok tivit sayısı çok olan hisse senetlerinin çok sayıda katılımcı tarafından ortaklaşa etiketlemesi ve herhangi bir etiketleme standardının olmamasının rolü olduğu tahmin edilmektedir.

7. SONUÇLAR VE ÖNERİLER

Tez çalışmasının kısa bir özeti ve katkısı, yapılan ve ileride yapılması düşünülen çalışmalar bu bölümde verilmiştir.

Bu tez çalışmasında, Twitter web arayüzü kullanılarak Twitter'dan toplam 21.031 KB boyutunda 119.343 adet tivit alınmıştır. Alınan tivitler, içeriklerine göre üç farklı sınıfa (pozitif, negatif ve nötr) ayrılmıştır. Bu işlem, tivitler borsa ve İngilizce bilgilieri değişkenlik gösteren 75 farklı katılımcı tarafından yapılmıştır.

Tivitler, öncelikle temizleme işlemine tabi tutulmuştur. Bunu sağlamak için öncelikle birebir aynı olan tivitler silinmiştir. Sonrasında elde kalan bu tekrarsız tivitlerin içersinde geçen, alfanumerik olmayan tüm karakterler silinmiştir. Daha sonra *Tab* karakteri de silinip, "[/(){}\[\]\|@,;]" karakterleri, *boşluk* karakteri ile değiştirilmiştir. Metin sınıflandırma için gerekli olmayan (bağlaçlar, ikilemeler v.b.) kelimeler de daha önceden hazırlanmış olan stopwords listesinde olup olmadığına bakılarak silinmiştir. "*RT*" ile başlayan tivitler, daha önce başka bir tivit olarak ele alındığı için, bu tivitler de silinerek göz ardı edilmiştir. *Hashtag (#)* karakteri de bizim için anlamsız olduğu için silinmiştir. "@" karakteri ile başlayan kelimeler, *varlığı* ifade ettiği için silinmiştir. Ayrıca tivit içerisinde geçen web adresleri ve linkler de silinmiştir. Böylece temizleme işlemi tamamlanmıştır. Standardı sağlamak amacıyla, tüm tivitler, tamamı küçük harf olacak şekilde değiştirilmiştir.

Metin sınıflandırma için, makine öğrenmesi teknikleri kullanılarak sınıflandırma yapılmıştır. TF-IDF yöntemi kullanılarak her bir veri setinde her bir tivit için geçen tüm kelimelerin frekans ağırlıkları hesaplanarak vektör haline dönüştürüp sayısallaştırılmıştır. Sayısallaştırılmış olan bu veriler üzerinde 6 farklı sınıflandırma algoritması kullanılarak başarım sonuçları elde edilmiştir.

AAPL, PFE, NFLX, MSFT, FB, MCS, GM, KO, TSLA ve GE, twitter verilerini sınıflandırarak yönünü tahmin etmeye çalıştığımız hisse senetleri olarak belirlenmiştir. Bu hisse senetleri için atılmış olan tivitlerin sınıflarını tahmin etmek için yaptığımız makine öğrenmesi yöntemi ile sınıflandırmada da NB, RF, SVM, DT, k-NN ve ANN sınıflandırıcıları kullanılmıştır. Bu sınıflandırma algoritmalarını kullanırken, homojenliği sağlamak adına corss-validation yöntemi ile veri setleri 10 parçaya ayrılmıştır. Bu parçalardan her birisi sıra ile doğrulama maksadı için kullanılmış, diğer parçalar ise sistemin eğitimde kullanılmıştır. En sonunda da tüm parçalar bittiği zaman, 10 parçanın ortalaması alınarak genel sınıflandırma başarısı elde edilmiştir.

7.1 Sonuçlar

Çalışmamız sonucunda elde ettiğimiz değerleri iki kategoride inceleyeceğiz. Bunlardan birincisi, ortaya koymuş olduğumuz sistemde, 6 farklı sınıflandırma algoritmasının eğitilmesi sonucu ortaya koydukları sınıflandırma başarı sonuçları, diğeri ise, bir gün öncesinin tivitlerinin sınıflarına bakarak, bir gün sonrası için yapılan tahminin başarısı.

Bu tezde, üç farklı sınıfın; (pozitif, negatif ve nötr) makine öğrenmesi yöntemi altında altı farklı sınıflandırma algoritması kullanarak analizleri yapılmış ve bir algoritma on farklı veri seti üzerinde eğitilmiş ve test edilmiştir. Kullanılan veri setleri içerisinde TSLA 33.348 adet tivit ile en büyük veri setine sahipken, AAPL 28.649 adet tivit ile ikinci, FB 19.969 adet tivit ile üçüncü, NFLX 14.187 adet tivit ile dördüncü, MSFT 9.344 adet tivit ile beşinci, GE 4.619 adet tivit ile altıncı, MCD 2.868 adet tivit ile yedinci, PFE 2.508 adet tivit ile sekizinci, KO 2.005 ile dokuzucu ve son olarak da GM 1.846 adet tivit ile en küçük veri setine sahiptir.

Uygulanan yöntem ve algoritmalar sonucu elde edilen sınıflandırma başarım oranları Çizelge 7.1'de gösterilmektedir.

Çizelge 7.1. Başarım sonuçları

Hisse Senetleri	Sınıflandırıcılar					
	NB	RF	SVM	DT	k-NN	ANN
AAPL	70,3	79,1	65,6	76,0	71,1	72,7
FB	71,2	82,9	64,3	82,5	69,0	64,3
GE	66,2	69,2	43,0	62,8	64,9	66,8
GM	82,7	87,5	75,2	95,5	83,6	75,2
KO	71,1	81,0	57,9	84,6	72,5	68,7
MCD	68,6	68,4	62,6	64,2	66,0	62,6
MSFT	64,7	71,0	54,8	66,0	63,3	65,6
NFLX	64,0	69,8	61,8	64,5	64,0	61,0
PFE	73,1	84,3	65,2	83,4	75,3	79,4
TSLA	65,7	80,5	63,7	74,7	67,6	63,7

Sınıflandırma yaparken üç farklı sınıfın kullanılmış olması, genel olarak sınıflandırma başarılarının oranlarını negatif yönde etkilediği söylenebilir.

Sınıflandırma algoritmalarının kıyasladığımız zaman ortalama olarak NB'nin %69,76, RF'nin %77,37, SVM'nin 61,41 DT'nin 75,42, k-NN'nin %69,73 ve ANN'nin 68,00 olduğu görülmektedir. Bu değerlere göre, sınıflandırma algoritmaları içerisinde, kullandığımız veri setleri için en iyi sınıflandırma başarı sonucunu %77,37 ile rastgele orman algoritması vermiştir. En kötü sınıflandırıcı olarak da bu veri seti için %61,41 ile destek vektör makineleri gösterilebilir.

Hisse senetlerinin başarı oranlarına bakıldığı zaman, AAPL %72,47, FB %72,37, GE %62,15, GM %83,28, KO %72,63, MCD %65,4, MSFT %64,23, NFLX %64,18, PFE %76,78 ve TSLA ise %69,32 olarak gerçekleşmiştir. Bu değerlere göre, hisse senetleri içerisinde en iyi sınıf tahmini %83,28 ile GM için, en kötü tahmin ise %62,15 ile GE için yapılabilmiştir.

Diğer taraftan, atılan pozitif ve negatif yorumlu tivitlerin, hisse senedinin ertesi gün borsada yapacağı hareketin yönünü tahmin etmede kullanılıp kullanılmayacağı araştırılmıştır. Bunu yaparken de, çok fazla anlamsız veri içerdiği için nötr veri sınıfı gözardı edilmiştir. Yön tahmin başarı hesabı, sadece pozitif ve negatif yorumlu tivitler üzerinden yapılmıştır. Nötr olarak etiketlenmiş olan tivitler yok sayılmıştır.

Ayrıca yön tahmin başarısı hesaplanırken, yine hisse senedinin yaptığı yatay hareketler göz ardı edilmiş, pozitif olarak etiketlenmiş bir tivit, ertesi gün hisse senedi %1'den daha fazla düşmemişse başarılı olarak alınmıştır. Yine negatif olarak etiketlenmiş bir tivit, ertesi gün hisse senedi %1'den daha fazla yükselmemişse başarılı olarak alınmıştır.

Bu kriterleri göre hesaplanan yön tahmin başarıları Çizelge 6.7'de gösterilmiş ve Şekil 6.9'da gösterildiği gibi tivit sayıları ile ters orantılı bir seyir izlediği fark edilmiştir.

7.2 Öneriler

Bu araştırmadan elde edilen sonuçlar incelendiği zaman, birtakım çıkarımlar ve ileriye dönük soru işaretleri ve araştırma konuları ortaya konulabilir.

Öncelikli olarak, yapılan çalışmada, corpus oluşturulurken, veri setlerinin büyüklükleri göz önünde bulundurularak İngilizce ve borsa bilgileri farklık gösteren 75 farklı katılımcıya el ile etiketleme yaptırılmıştır. Bu da etiketlemeler arasında bir standart bulunmamasına neden olmuştur. Tahmin başarıları incelendiği zaman, tamamını tek bir

kişinin etiketlediği daha küçük veri setine ait hisse senetlerindeki tahmin başarı oranlarının %95'ler seviyesine kadar çıktığı görülmektedir. Bu durum göz önünde bulundurulduğu zaman, daha az tivitten oluşsa bile, el ile düzgün şekilde etiketlenmiş daha tutarlı bir corpus üzerinde çalışılırsa, çok daha başarılı sonuçların alınabileceği çıkarımını yapmak yanlış olmaz. Mevcut çalışmada diğer sınıflandırma algoritmalardan daha iyi sonuç vermiş olan rastgele orman algoritması veya en düşük sonucu vermiş olan destek vektör makinesi, değiştirilmiş ve daha tutarlı hale getirilmiş veri seti üzerinde daha farklı sonuçlar verebilir.

Bu çalışmamızda metin sınıflandırma işlemini makine öğrenmesi yöntemi ile gerçekleştirdik. Aynı konu üzerinde sözlük tabanlı yöntemler kullanılarak, semantik-sentaktik yaklaşımlar ile duygu analizi ve doğal dil işleme yöntemleri ile daha farklı sonuçlar elde etmek de mümkün olabilir.

Sonuçları sınıflandırma algoritmaları penceresinden değerlendirdiğimiz zaman, genel olarak tüm algoritmaların tatmin edici başarı oranları elde ettiğini görüyoruz. Muhtemeldir ki daha farklı sınıflandırma algoritmaları kullanılarak da %70 - %75 bandında sınıflandırma başarı sonuçları alınabilir ki bu da üç sınıflı bir sınıflandırmada tatmin edici bir orandır. Ancak hisse senedi tahmin başarıları penceresinden bakıldığı zaman, çok daha farklı araştırmalar yapılabilir. Biz, bu çalışmamızda, dünyanın en stabil borsalarından biri olan Dow Jones borsasını ve burada işlem gören dünyanın en büyük şirketlerinin hisse senetlerinin yön tahmini yapmaya çalıştık. Bu da yaptığımız çalışmanın spekülasyona çok fazla açık olmayan bir platformda olduğu anlamına gelmektedir. Halbuki yaptığımız araştırmanın temelinde atılan tivitlerin hisse senetlerini ne kadar manipüle edebileceği araştırılmaktadır. Aynı şekilde düşünüldüğü zaman, başka ülkelerin daha spekülatif menkul kıymetler borsasında işlem gören, işlem hacmi çok daha küçük hisse senetleri üzerinde çalışılırsa daha başarılı sonuçlar elde edilebileceği çıkarımı da yapılabilir. İleride benzer çalışmalar daha spekülatif borsalar ve daha spekülatif hisse senetleri üzerinde yapılabilir.

Bu çalışmamızın benzerlerinin, farklı ülkelerin menkul kıymetler borsaları üzerinde yapılması durumunda, veri setleri, farklı diller ile atılan tivitlerden oluşacaktır. Bu durumda da farklı diller için duygu analizi gündeme gelecektir. Farklı diller için, örneğin Türkçe için yapılmış bir duygu analizi ile İstanbul Menkul Kıymetler Borsasında işlem gören hisseler için yön tahmini gibi bir çalışma, çok değişik sonuçlar verebilir.

Bu çalışmamızda işlem hacminin çok daha büyük olması sebebi ile menkul kıymetler borsasındaki hisse senedi için tahminler yapılmıştır. Ancak benzer çalışmalar emtia piyasaları için de gerçekleştirilebilir ve ilgi çekici sonuçlar elde edilebilir.

KAYNAKLAR

Agarwal, A., Xie, B., Vovsha, I., Rambow, O., Passonneau, R., 2011, Sentiment analysis of twitter data, *LSM '11 Proceedings of the Workshop on Languages in Social Media*, 30-38.

Akın, Ç., E., 2017, Makine öğrenmesi nedir? - ML#1 [online], Kocaeli, http://cagriemreakin.com/veri-bilimi/makine-ogrenmesi/maline-ogrenmesi-nedir-ml-1.html, [Ziyaret tarihi: 9 Haziran 2019].

Albayrak, A., S., Yılmaz, Ş., K., 2009, Veri madenciliği: Karar ağacı algoritmaları ve İMKB verileri üzerine bir uygulama, *Süleyman Demirel Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi,* 14 (1), 31-52.

Albeni M., Demir Y., 2005, Makro ekonomik göstergelerin mali sektör hisse senedi fiyatlarına etkisi (İMKB uygulamalı), *Muğla Üniversitesi SBE Dergisi,* Sayı 14, 4–10.

Anonim, 2018, Karar ağaçları (Decision trees) [online], California – San Francisco – A.B.D., https://veribilimcisi.com/2018/02/23/karar-agaclari-decision-trees, [Ziyaret tarihi: 12 Haziran 2019].

Ayhan, S., Erdoğmuş, Ş., 2014, Destek vektör makineleriyle sınıflandırma problemlerinin çözümü için çekirdek fonksiyonn seçimi, *Eskişehir Osmangazi Üniversitesi İİBF Dergisi,* 9 (1), 175-198.

Bermingham, A., Smeaton A., F., 2010, Classifying sentiment in microblogs: Is breavity an advantage?, *Proceedings of the 19th ACM International Conference on Information and Knowledge Management,* 1833-1836.

Blei, D., M., Ng, A., Y., Jordan, M., I., 2003, Latent Dirichlet Allocation, *The Journal of Machine Learning Research*, 3, 993-1022.

Bollen, J., Mao, H., and Zeng, X., 2011, Twitter moods predicts the stock market, *Journal of Computational Science*, 2 (1), 1-8.

Can, U., Alatas, B., 2017, Duygu analizi ve fikir madenciliği algoritmalarının incelenmesi, *International Journal of Pure and Applied Sciences,* 75-111.

Dayıbaşı, O., 2017, Makine öğrenmesi nedir?, [online], A.B.D., https://medium.com/@odayibasi/makine-%C3%B6%C4%9Frenmesi-nedir-eac85d27d438, [Ziyaret tarihi: 8 Haziran 2019].

Demir, Y., 2001, Hisse senedi fiyatlarını etkileyen işletme düzeyindeki faktörler ve mali sektör üzerine İMKB'de bir uygulama, *Süleyman Demirel Üniversitesi İktisadi ve İdari Bilimler Fakültesi*, Isparta, 117 – 119.

Demiriz, A., 2006, Karar ağaçları ile sınıflandırma [online], Berlin – Almanya, https://slideplayer.biz.tr/slide/2745480, [Ziyaret tarihi: 12 Haziran 2019].

Dolgun, O., 2013, Destek Vektör Makineleri [online], Dublin – İrlanda, https://www.slideshare.net/ozgur_dolgun/destek-vektr-makineleri, [Ziyaret tarihi: 9 Haziran 2019].

Engin, Ö., Fığlalı, A., 2002, Akış tipi çizgeleme problemlerinin genetik algoritma yardımı ile çözümünde uygun çaprazlama operatörünün belirlenmezi, *Doğuş Üniversitesi Dergisi*, Sayı: 6: 27-35.

Fidancı, A., S., 2017, Random forest [online], Dublin – İrlanda, https://www.slideshare.net/SezerFidanc/random-forest-algoritmas?qid=3aaadeab-c590-46d1-a6ae-858ef14f382b&v=&b=&from_search=2, [Ziyaret tarihi: 13 Haziran 2019].

Gamon, M., 2004, Sentiment classification on customer feedback data: Noisy data, large feature vectors, and the role of linguistic analysis, *Proceedings of the 20th International Conference on Computational Linguistics.*

Go, A., Bhayani, R., Huang, L., 2009, Twitter sentiment classification using distant supervision, Technical support, *Stanford Digital Library Technologies Project,* 1-6.

Gülcan, 2019, Yeni başlayanlar için A'dan Z'ye makine öğrenmesi [online], San Francisco, California, A.B.D., https://www.kizgibikodla.com/news/yeni-baslayanlar-icin-adan-zye-makine-ogrenmesi/, [Ziyaret Tarihi: 8 Haziran 2019].

Hao, M., Rohrdantz, C., Janetzko, H., Dayal, U., Keim, D., A., Haug, L., E., 2011, Visual sentiment analysis on twitter data streams, *IEEE Conference on Visual Analytics Science and Technology (VAST).*

Hatipoğlu, E., 2018, Machine learning – Classification – Naive Bayes – Part 11 [online], A.B.D., https://medium.com/@ekrem.hatipoglu/machine-learning-classification-naive-bayes-part-11-4a10cd3452b4, [Ziyaret tarihi: 9 Haziran 2019].

Hu, M., Liu, B., 2004, Mining and summarizing costumer reviews, *KDD '04 Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining,*

Kang, D., Park, Y., 2014, Review-based measurement of customer satisfaction in mobile service: Sentiment analysis and VIKOR approach, *Expert Systems With Applications,* 41, 1041-1050.

Kaya, M., Fidan, G., and Toroslu, I., H., 2012, Sentiment analysis of Turkish political news, IEEE/WIC/ACM *International joint conferences on web intelligence and intelligent agent technology,* 174-175.

Kaya, M., Fidan, G., and Toroslu, I., H., 2013, *Transfer* learning using twitter data for improving sentiment classification of turkish political news, *Information Sciences and Systems,* 139-148.

Kim, M., Hovy, E., 2004, Determining the sentiment of opinions, *COLING '04 Proceedings of the 20th international conference on Computational Linguistics*, 1367.

Koç, R., 2016, Karar ağaçları (decision tree) [online], Dublin - İrlanda, https://www.slideshare.net/reyhanko/karar-aalari-63285758?qid=fc6c4725-e964-4cbb-8307-fd6da630f762&v=&b=&from_search=1, [Ziyaret tarihi: 12 Haziran 2019].

Kushal, D., Steve, L., Pennock, D.M., 2003, Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews, *In Proceedings of WWW'03, 12th International Conference on World Wide Web*, Budapest Congress Centre, Macaristan, 519-528.

Küçüksille, E., U., Ateş, N., 2013, Destek vektör makineleri ile yaramaz elektronik postaların filtrelenmesi, *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi,* Isparta, 6 (1).

Lafferty, J., McCallum, A., Pereira F., 2001, Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data, *In Proceedings of ICML '01, 18th International Conference on Machine Learning*, Williams College, Williamstown, MA, ABD, 282–289.

Liu, B., 2012, Sentiment Analysis and Opinion Mining, *Morgan & Claypool Publishers.*

Medhat, W., Hassan, A., Korashy, H., 2014, Sentiment analysis algorithm andapplications: A survey, *Ain Shams Engineering Journal*, 5(4), 1093-1113.

Mittal, A., Goel, A., 2012, Stock prediction using twitter sentiment analysis, *Stanford Univeresity Working Paper.*

Ohana, B., Tierney, B., 2009, Sentiment classification of reviews using SantiWordNet, *9th. IT & T Conference | Scool of Computing – Dublin Institute of Technology.*

Orçun, Ç., 2010, Finansal piyasalarda alım satım kararlarında teknik analiz ve İMKB uygulaması, Yüksek lisans tezi, *Dokuz Eylül Üniversitesi Sosyal Bilimler enstitüsü,* İzmir, 39-83.

Özçam, M., 1990, Hisse senetleri fiyatlarını belirleyen unsurlar ve Türkiye, *SPK*, Ankara.

Öztemel, E., 1992, Integrating expert systems and neural networks for intelligent online statistical process control, *PhD Thesis, School of Electrical, Electronic and Systems Engineering*, Cardiff, 67-98.

Özyurt, B., Akcayol, M., A., 2018, Fikir madenciliği ve duygu analizi, yaklaşımlar, yöntemler üzerine bir araştırma, *Selçuk Üniversitesi Mühendislik Bilim ve Teknoloji Dergisi,* 6 (4), 668-693.

Pang, B., Lee, L., Vaithyanathan, S., 2002, Thumbs up?: Sentiment Classification Using Machine Learning Techniques, *In Proceedings of EMNLP-2002, Conference on Empirical Methods in Natural Language Processing*, University of Pennsylvania, PA, ABD, 79-86.

Pang, B., Lee, L., 2004, A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts, *ACL '04 Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics,* 271.

Papadrakakis, M., Lagaros, N. D., 1994. Advances in Structural Optimization, *İnstitute of Structural Analysis and Seismic Research, National Technical University*, Zografou Campus, Civil-Comp Press, Athens, Greece.

Park, A., Paroubek, P., 2010, Twitter as a corpus for sentiment analysis and opinion mining, *Proceedings of the International Conference on Language Resources and Evaluation.*

Rabiner, L.R., 1989, A tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, *Proceedings of the IEEE*, 77, 257-286.

Saif, H., He, M., and Alani, H., 2012, Semantic sentiment analysis of twitter, *International Semantic Web Conference,* 508-524.

Shehu, H., A., 2019, Kutupsallık sözlüğü ve yapay zekâ yardımı ile Türkçe twitter verileri üzerinde duygu analizi, Yüksek lisans tezi, *Pamukkale Üniversitesi Fen Bilimleri Enstitüsü*, Denizli, 11-61.

Soman, K., P., Loganathan, R., Ajay, V., 2011, Machine learning with SVM and other kernel methods, *PHI Learning Pvt. Ltd.,* 486.

Şahin, İ., 2008, Uzman Sistem Kullanarak İki Boyutlu İzdüşümlerden Katı Modeller Oluşturma, Doktora Tezi, *Gazi Üniversitesi Fen Bilimleri Enstitüsü*, Ankara, 36-59.

Tetsuya, N., Jeonghee, Y., 2003, Sentiment Analysis: Capturing Favorability Using Natural Language Processing, *In Proceedings of KCAP-03, 2nd International Conference on KnowledgeCapture, Sanibel Island*, FL, ABD, 70-77.

Tong, R., M., 2001, An Operational System for Detecting and Tracking Opinions in On-Line Discussion, *In Proceedings of SIGIR 2001 Workshop on Operational Text Classification*, New Orleans, Louisiana, ABD.

Turney, P., 2002, Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews, *ACL '02 Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics.* 417-424.

Vasileios, H., Janyce, M., W., 2000, Effects of Adjective Orientation and Gradability on Sentence Subjectivity, *In Proceedings of COLING-2000, 18th Interntional Conference on Computational Linguistics*, Saarbrücken, Almanya, 299-305.

Wikimedia Foundation Inc., 2016, Genetik Algoritma [Online], USA, [Ziyaret Tarihi: 9 Haziran 2019].

Wikipedia, 2019, Naive Bayes classifier [online], A.B.D., https://en.wikipedia.org/wiki/Naive_Bayes_classifier, [Ziharet tarihi: 9 Haziran 2019].

Wilson, T., Wiebe, J., Hoffmann, P., 2005, Recognizing contextual polarity in phrase-level sentiment analysis, *HLT '05 Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, 347-354.

Yalçın, N., 2019, Bayes öğrenmesi, Dublin – İrlanda, https://www.slideshare.net/YazhanKerimDeer/bayes-renmesi?qid=a6919542-0997-42c1-8754-5475d214dbff&v=&b=&from_search=1, [Ziyaret tarihi: 12 Haziran 2019].

Yurtcu, Ş., İçağa, Y., 2006, Evrimsel Algoritmaların İnşaat Mühendisliği Sistemlerinde Kullanımı, *Yapı Teknolojileri Elektronik Dergisi,* Sayı: 1, 51 – 59.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Mustafa Vehbi TÜRKALP
Uyruğu : T.C.
Doğum Yeri ve Tarihi : Konya – 08.03.1981
Telefon : 0 506 510 12 42
Faks : -
E-Posta : alpkocaturk@yahoo.com

EĞİTİM

Derece	Adı	İlçe	İl	Bitirme Yılı
Lise	: Özel tür-mak Fen Lisesi	Selçuklu	Konya	1999
Üniversite	: Selçuk Üniversitesi	Selçuklu	Konya	2003
Yüksek Lisans	: Konya Teknik Üniversitesi	Selçuklu	Konya	2019
Doktora	:			

İŞ DENEYİMLERİ

Yıl	Kurum	Görevi
2004-2008	Mega Information Technology Ltd. Şti.	Yazılım Mühendisi
2008-2014	Turquaz Yazılım Ltd. Şti.	Yazılım Mühendisi
2014-	Havelsan A.Ş.	Yazılım Mühendisi

UZMANLIK ALANI

Java, C++, MSSQL, PostgreSQL, HTML.

YABANCI DİLLER

İngilizce, Arapça (orta)