İSTANBUL TEKNİK ÜNİVERSİTESİ ★ SOSYAL BİLİMLER ENSTİTÜSÜ

DOW JONES ENDÜSTRİ ORTALAMASI VE NASDAQ 100 ENDEKSLERİNİN MAKİNE ÖĞRENMESİ ALGORİTMALARI İLE TAHMİNİ

YÜKSEK LİSANS TEZİ

Öğrenci Adı SOYADI

Sezer AKTAŞ

İşletme Anabilim Dalı

İşletme Yüksek Lisans Programı

Tez Danışmanı: Prof. Dr. Oktay TAŞ

HAZİRAN 2019

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ SOSYAL BİLİMLER ENSTİTÜSÜ

DOW JONES ENDÜSTRİ ORTALAMASI VE NASDAQ 100 ENDEKSLERİNİN MAKİNE ÖĞRENMESİ ALGORİTMALARI İLE TAHMİNİ

YÜKSEK LİSANS TEZİ

Sezer AKTAŞ
(403161025)

İşletme Anabilim Dalı

İşletme Yüksek Lisans Programı

Tez Danışmanı: Prof. Dr. Oktay TAŞ

HAZİRAN 2019

İTÜ, Sosyal Bilimler Enstitüsü'nün 403161025 numaralı Yüksek Lisans Öğrencisi Adı SOYADI, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı "DOW JONES ENDÜSTRİ ORTALAMASI VE NASDAQ 100 ENDEKSLERİNİN MAKİNE ÖĞRENMESİ ALGORİTMALARI İLE TAHMİNİ" başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı :	**Prof. Dr. Oktay TAŞ** İstanbul Teknik Üniversitesi	.....
Jüri Üyeleri :	**Prof. Dr. Oktay TAŞ** İstanbul Teknik Üniversitesi	.....
	Doç. Dr. Kaya TOKMAKÇIOĞLU İstanbul Teknik Üniversitesi	.....
	Doç. Dr. Cem BERK İstanbul Arel Üniversitesi	.....

Teslim Tarihi : 06 Mayıs 2019
Savunma Tarihi : 14 Haziran 2019

Aileme,

ÖNSÖZ

Bu çalışmamda emeği geçen aileme, çalışırken yüksek lisans yapmamda bana gösterdiği anlayış için çalıştığım kurum olan Borsa İstanbul'a ve tez danışmanım Prof. Dr. Oktay TAŞ'a sonsuz teşekkürlerimi sunarım.

Mayıs 2019 Sezer Aktaş

İÇİNDEKİLER

Sayfa

ÖNSÖZ ……...vii
İÇİNDEKİLER.....ix
KISALTMALAR.....x
SEMBOLLER.....xi
TABLO LİSTESİ..... xiii
ŞEKİL LİSTESİ.....xv
ÖZET……...xvii
SUMMARY.....xviii
1- GİRİŞ.....1
2- FİNANSAL SİSTEM.....3
3- BORSALAR ve ENDEKSLER.....7
3.1 Dow Jones Endüstri Ortalaması.....8
3.2 Nasdaq 100 Endeksi.....11
4- ALGORİTMİK İŞLEMLER.....12
5- LİTERATÜR ARAŞTIRMASI.....15
5.1 Makine Öğrenmesi ile Finansal Piyasaların Tahmin Edilmesi.....15
6- METODOLOJİ.....21
6.1 Veri Madenciliği ve Makine Öğrenmesi.....21
6.1.1 Özellik Seçimi.....24
6.1.2 Özellik Çıkarımı.....25
6.2 Rassal Orman Algoritması.....26
6.3 Destek Vektör Makineleri.....29
6.4 Veri Seti.....30
6.5 Model Sonuçları.....34
7- SONUÇ VE ÖNERİLER.....46
KAYNAKLAR.....48
EKLER…….. 53
ÖZGEÇMİŞ.....66

KISALTMALAR

ARCH : The autoregressive conditional heteroskedasticity
ARMAX : Autoregressive Moving Average eXogenous
BIST : Borsa İstanbul A.Ş.
DVM : Destek Vektör Makineleri
GARCH : The generalized autoregressive conditional heteroskedasticity
HFT : High Frequency Trading (YFİ)
kNN : k-En Yakın Komşu Algoritması
ML : Machine Learning
NASDAQ : National Association of Securities Dealers Automated Quotation
NYSE : New York Stock Exchange

SEMBOLLER

DJIA **:** Dow Jones Industrial Average Index

IXIC **:** Nasdaq Composite

NDX **:** The Nasdaq 100 Index

S&P 500 **:** Standart and Poor's 500 endeksi

ÇİZELGE LİSTESİ

Sayfa

Çizelge 1 : DJIAA şirketleri ve ağırlıkları. 9
Çizelge 2 : Ekonomik girdilere ait özet tablolar. 32
Çizelge 3 : DJIA test verisi. 35
Çizelge 4 : DJIA grid araması verisi. 35
Çizelge 5 : DJIA validasyon verisi. 35
Çizelge 6 : NDX test verisi. 36
Çizelge 7 : NDX grid araması verisi. 36
Çizelge 8 : NDX validasyon verisi. 36
Çizelge 9 : RFE algoritma sonucu DJIA- DVM. 37
Çizelge 10 : RFE algoritma sonucu DJIA -RF. 39
Çizelge 11 : RFE Algoritması NDX-DVM. 41
Çizelge 12 : RFE Algoritması NDX-RF. 43

ŞEKİL LİSTESİ

Şekil 1 : CRISP-DM modeli. 22
Şekil 2 : Makine öğrenmesi adımları. 23
Şekil 3 : Yaygın kullanılan ML algoritmaları. 24
Şekil 4 : Rassal Orman Teorisi (Mennitt, Sherrill& Fristrup, 2014). 28
Şekil 5 : Destek Vektör Makineleri. 30
Şekil 6 : DJIA fiyat-zaman grafiği. 34
Şekil 7 : NDX fiyat-zaman grafiği. 34
Şekil 8: DJIA – DVM sınıflandırma raporu. 38
Şekil 9 : DJIA- DVM ROC eğrisi. 39
Şekil 10 : DJIA- Rassal Orman sınıflandırma raporu. 41
Şekil 11: DJIA- Rassal Orman ROC eğrisi. 41
Şekil 12 : NDX- DVM sınıflandırma raporu. 43
Şekil 13 : NDX DVM ROC eğrisi. 43
Şekil 14 : NDX Rassal Orman sınıflandırma raporu. 44
Şekil 15 : NDX Rassal Orman ROC eğrisi. 45

DOW JONES ENDÜSTRİ ORTALAMASI VE NASDAQ 100 ENDEKSLERİNİN MAKİNE ÖĞRENMESİ ALGORİTMALARI İLE TAHMİNİ

ÖZET

Borsalar herhangi bir ülkenin ekonomik durumu ve gelişimini gösteren önemli finansal kuruluşlardandır. Hisse fiyatları doğası gereği oynaktır ve ekonomik büyüme, enflasyon, kur, haberler gibi birçok konudan etkilenmektedir. Genel ekonomide olduğu gibi hisse fiyatları da arz ve taleple belirlenmektedir. Talebi yüksek olan hisselerin fiyatı aniden yükselecek; hızla elden çıkarılan hisselerin fiyatları aniden düşecektir. Dalganan hisse fiyatları yatırımcıyı olumsuz etkilediği için gelecek hisse fiyatını tahmin etmek önem kazanmıştır.

Finansal piyasaların sayısal verilerle tahmin edilmesi her zaman araştırmacıların ilgisini çeken bir alan olmuştur. Bu amaçla literatürde birçok araştırma yapılmış; hem istatistik ve ekonometri alanında, hem de makine öğrenmesi alanında birçok sonuç elde edilmiştir. Finansal piyasaların doğası gereği risk barındırması ve yatırımcıların getiri maksimizasyonu hedeflemesi sebebiyle son zamanlarda daha popüler bir hale gelmiştir.

Genellikle sayısal yöntemleri kullanarak işlem yapan işlemciler hisse ya da türev ürünleri ucuz fiyattan alarak yüksek fiyata satmak amacıyla işlem yapmaktadır. Hisse piyasası tahmini yeni bir konu değil; son zamanlarda popülerlik kazanmış bir konudur. Genellikle yatırımcılar temel ve teknik analiz yöntemleri kullanarak yatırım yapmak istedikleri hisseleri belirmektedir. Temel analiz endüstri performansı, ekonomi, siyasi çevre gibi kriterlere bakarken; teknik analiz işlem hacmi, kapanış fiyatı gibi piyasa aktivitesinden yola çıkmaktadır. Son zamanlarda makine öğrenmesine olan ilgideki artış birçok alanda bu tekniklerin uygulanmasına ve işe yarar sonuçlar elde edilmesini sağlamıştır.

Bu çalışmada öncelikle borsalar, dünya borsaları, endeksler hakkında bilgi verilmiş; sonrasında tahmin yapmak için seçilen DJIA ve NDX endekslerinin 2012-2018 verileri üzerinden model hazırlanmıştır. Modelde 2012-2017 veri aralığı eğitim verisi; 2018 test verisi olarak kullanılmıştır. 2019 yılı ise doğrulama amacıyla modelde yer almıştır. Girdi olarak kullanılan göstergeler çoğu araştırmada yer alan teknik göstergeler olup açıklamarına yer verilmiştir. Yöntem olarak Rassal orman ve destek vektör makineleri kullanılmıştır. Modelde DJIA için en uygun yöntem Rassal orman olarak çıkarken; NDX için DVM yöntemi daha başarılı sonuç vermiştir. Tüm kodlar Python versiyon 2.7'de yazılmış; model hesaplamasında scikit-learn kütüphanesinden faydalanılmıştır.

PREDICTION OF THE DOW JONES INDUSTRIAL AVERAGE AND THE NASDAQ 100 INDICES USING MACHINE LEARNING

SUMMARY

Stock exchanges are considered as the indispensable financial institutions of a country due to the fact that they provide insigth on the economic development and conditions. Stock prices are volatile by nature and are affected by the economic development, inflation, foreign exchange prices. As in the microeconomics, stock prices are also determined by the demand and supply rules. The prices of the highly demanded stocks are high; on the contrary they will drop suddenly once investorts sell in high quantities. Fluctuating stock prices have unfavourable impact on the investors, therefore the prediction of the stock prices is considered as important by the financial actors.

Quantitave prediction of the stock market based on the quantitative data has been an attractive area for the researches. To this objective, a great bulk of research has been conducted and both the econometrics, statistics and machine learning algorithms have provided workable results. Due to the risk embedded in the financial market and investors' striving for the profit maximization, prediction has become more and more popular in recent years.

Traders using the quantitative techniques to trade in the stock market tend to buy the equity or derivatives in lower price and sell them for the higher price. Stock price prediction is not a recent subject; however as stated before, they have become more apparent in recent years. Generally investors use fundamental and technical analysis to decide on the stocks they want to trade. Fundamental analysis focuses on the industry performence, political environment, economic conditions; whereas the technical analysis is interested in the trade volume, closing price and market activity. The increase in the machine learning field has led to the application of these techniques to various fields and they have provided promising results.

In this thesis, information upon the stock exchanges, world stock exchanges and indices have been provided, then the model has been founded on the closing price of the selected DJIA and NDX indices between the 2012-2018 years period. In the model, the dataset has been split into train and test data and 2012-2017 has been used for training the model, whereas the 2018 has been the test year. 2019 date is the validation data present in the model. As input, technical analysis indicators examined by the prior researchers have been chosen and the explanation regarding these indicators has been provided. As methodology random forest and support vector machines have been selected. For the DJIA indice, the random forest method has proved the best result; as compared to SVM; whereas for the NDX SVM model has proved the best result to the random forest. All of the codes has been written in Python 2.7 version and models have been calculated using the scikit-learn library.

1- GİRİŞ

Finansal zaman serilerinin tahmini alanında borsa tahminleri zorlu bir süreç olarak adlandırılmaktadır. Bunun sebebi piyasa hareketlerinin belirsiz seyretmesidir. Siyasi olaylar, genel ekonomik durum ve işlem yapan kurumların beklentileri gibi çoğu makroekonomik, siyasi ve psikolojik etmenler piyasayı etkilemektedir. Bu sebeple piyasa fiyat hareketlerinin tahmin edilmesi uğraş gerektiren bir alan olmuştur. Akademik araştırmalara göre piyasa fiyat hareketleri rassal değildir; aksine son derecede doğrusal olmayan, dinamik bir biçimde hareket etmektedir. Ayrıca piyasadaki fiyat hareketlerinin yönünün tahmin edilmesi ilgili piyasaya yatırım yapan yatırımcılar için kar elde etmede önemli bir hale gelmiştir. Piyasada alım ya da satım yönlü işlem yapmak isteyen yatırımcılar için fiyatların hangi yönde hareket edeceği beklentisi son derecede önemlidir. Ayrıca fiyatların hareket yönünün tahmin edilmesi fiyatların tahmininden daha fazla kazanç sağlamaktadır (Chen, Leung& Daouk, 2003).

Finansal piyasalardaki fiyat tahmini ekonomi, bilgisayar bilimi, finans, tarih, matematik alanından birçok farklı araştırmacının ilgisini çekmiş, üzerinde fazla sayıda araştırma yapılmıştır. Piyasaların oynak yapısı, basit zaman serisi ya da regresyon tekniklerinin uygulanmasını zorlaştırmıştır. Finansal kurumlar ve işlem yapanlar finansal piyasalarda yüksek kazanç elde etmek amacıyla portföy ve müşterileri için farklı farklı işlem modelleri geliştirmiş; ancak çoğu ortalamanın üzerinde kar edememiştir. Ancak, finansal tahmin alanı en küçük yüzdelik bir birim farktan dahi milyonlarca kar elde etme imkânı sağladığı için bu kurumlar için popülaritesini sürdürmektedir. Bu zamana kadar çoğu tahmin yöntemi lineer istatistik zaman serileri modellerine odaklanmıştır. Ancak hisse fiyatlarının ve diğer varlıkların yönündeki değişkenlik lineer teknikleri optimum olmaktan çıkarmaktadır ve ARCH gibi lineer olmayan modeller daha düşük tahmin hatası sağlamaktadır (Zhang, Shan& Su, 2009).

Araştırmacılar finansal tahmin amacıyla son zamanlarda büyük veri ile ilgili bilgisayar bilimleri ve makine öğrenmesine odaklanmıştır. Büyük veri ve makine öğrenimi yöntemleri sayesinde finansal kuruluşlar yüksek frekanslı ve algoritmik işlemler yapabilmektedir.

Pay piyasası tahmini finans, mühendislik ve matematik alanında finansal kazanç potansiyeli sayesinde ilgi gören bir alan olmuştur. Bununla birlikte pay piyasasının tahmin edilir olup olmadığını tartışılmıştır. Etkin piyasalar hipotezinin (EMH) geçerli olup olmadığıyla ilgili herhangi bir fikir birliği olmaması sebebiyle araştırmacılar finansal piyasaların tahmin edilebilir olduğunu kanıtlamak amacıyla birçok çalışma yapmıştır. Bilişim teknolojisindeki gelişmeler ve internet üzerinden ulaşılabilir geniş kaynaklar sayesinde pay piyasası stratejik yatırımcılar ve bireysel yatırımcılar için ulaşılabilir hale gelmiştir.

Bu tezin amacı NDX ve DJIA endekslerinden test ve eğitim verisi oluşturarak Rassal orman ve DVM modellerinin başarısını test etmektir. İki model parametresi de Gridsearch ile bulunmuş ve ilgili modeller bu parametreleri kullanarak oluşturulmuştur. Sonuçta hem test hem de doğrulama verisinde de eğitim verisinin öğrendiği bu algoritma benzer sonuçlar elde etmiştir.

2- FİNANSAL SİSTEM

Finansal sistem ödünç verenler, yatırımcılar ve ödünç alanlar arasında fonların akışını mümkün kılan sistemdir (Sarıkamış, 1980). Finansal sistemler ulusal, uluslararası ölçeklerde bulunmaktadır. Bu sistem, karmaşık, birbiriyle yakından ilişkili hizmetler, piyasalar, kurumlar sayesinde yatırımcılar ve mevduat sahipleri arasında etkin ve sürekli bir bağlantı oluşturmaktadır. Para, kredi ve finansal enstrümanlar finansal sistemlerde değişim aracı olarak kullanılmaktadır ve takas sistemine alternatif olarak mal ve hizmetlerin bilinen değeri üzerinden el değiştirmesine olanak sağlamaktadır. Modern bir finansal sistemde kamu ya da özel sektör tarafından yönetilen bankalar, finansal piyasalar, finansal enstrümanlar ve finansal hizmetler yer almaktadır. Finansal sistem fonların tahsis edilerek yatırıma dönüşmesi ya da sektörler arası akışını sağlamaktadır. Finansal sistem sayesinde bireysel ve kurumsal yatırımcılar riski paylaşmaktadır. Finansal sistem belli başlı olarak dört unsurdan oluşmaktadır: finansal kurumlar, finansal piyasalar, finansal enstrümanlar ve finansal hizmetler (Coşkun, 2010). Finansal sistem para arzı ve talebi olanları bir araya getiren birleştirici bir bağlantı oluşturarak ekonomik ve endüstriyel büyümeyi teşvik etmektedir. Buradan yola çıkarak finansal sistem varlıkların ve risklerin el değiştirmesini kolaylaştırmak amacıyla kurulan piyasalar ve kurumların bir bütünü olarak tanımlanabilmektedir.

Finansal piyasalar finansal enstrümanların, hizmetlerin alım satımının yapıldığı merkezlerdir. Bireylerin, şirketlerin ve kurumların kredi ihtiyaçlarını karşılamakta, mevduat, çek, bono, tahvil gibi birçok çeşitli enstrümanlarla yatırımcıların çeşitli taleplerini karşılamaktadır. Yatırımcılar ve ödünç alanlar arasında fonların akışını sağlayarak iletim mekanizması rolü oynamaktadır. Piyasalar, finansal varlık ve kredi enstrümanlarına yönelik işlem yapılmasını kolaylaştırmak amacıyla oluşturulan resmi işlem kurallarıyla bağlı bireysel yatırımcılar, finansal kurumlar ve diğer aracı kurumlardan oluşmaktadır (Afşar, 2006).

Finansal piyasalar hisselerin, borçlanma araçlarının, dövizlerin ve türev ürünlerin işlem gördüğü piyasalar anlamına gelmektedir. Bazı finansal piyasalarda hacim düşükken New York Borsası gibi piyasalarda günlük trilyon dolarlık işlem hacmi

gerçekleşmektedir. Finansal piyasa fiyatları vergilendirme dolayısıyla hissenin gerçek değerini yansıtmayabilir. Ayrıca kıymetler fiyatları, fiyatların piyasa tarafından etkin ve doğru bir biçimde belirlenmesine ve şeffaflığa dayanmaktadır.

Finansal piyasalar sermayenin etkin bir biçimde tahsis edilmesini sağlamak amacıyla fonların herhangi bir yatırım fırsatı olmayanlardan olanlara hareket etmesini sağlayarak sermayenin etkin bir biçimde tahsis etmesine olanak sağlaması açısından önem taşımaktadır. Finansal piyasadaki belli başlı aktörler finansal kuruluşlar, aracı kurumlar, ödünç verenler, ödünç alanlar ve mevduat sahipleridir (Aras, 2003).

Para piyasası paranın yakın ikameleri ya da paraya benzer bono, hazine bonoları, ödeme senetleri gibi vadesi bir seneyi aşmayan enstrümanların konu edildiği piyasadır. Bu mekanizma sayesinde ödünç alanlar ve verenler kısa vadeli fonlara yatırım yapmak amacıyla bir araya gelmektedir. Kısa vadeli işlem yapan tüm bireyler, kurumlar ve aracılar bu piyasanın aktörlerindedir. Ödünç alanların kısa vadeli ihtiyaçlarını karşılarken ödünç verenlerin likidite ve nakit ihtiyaçlarını karşılamaktadır. Para piyasası yüksek likiditeye sahip kısa vadeli finansal enstrümanların işlem gördüğü piyasalardır. Piyasa katılımcıları tarafından vadesi gecelikten bir yıllığa kadar borç alıp verme amacıyla kullanılmaktadır. Para piyasası işlemleri genellikle perakende ödemeler olup yüksek hacimlidir ve bireysel yatırımcılardan ziyade şirketler ve finansal kuruluşlar arasında gerçekleşmektedir. Para piyasası fonları bireysel yatırımcılara para piyasası varlıklarına küçük hacimlerde yatırım yapmasına olanak sağlamıştır (Levine, 2004).

Para piyasalarının bireyler, şirketler ve hükümet kuruluşlarına yönelik birçok fonksiyonu bulunmaktadır. Para piyasasına başvurmak için en önemli amaç likidite sağlamaktır. Kısa vadeli borçlanma yapıldığında bu borç genellikle operasyonel giderleri karşılamak ya da çalışma sermayesine katkıda bulunması için kullanılmaktadır. Semaye iyileştirmeleri ya da büyük çaplı projelerde kullanılmamaktadır.

Para piyasaları şirketler ve hükümetlere günlük olarak uygun likidite seviyesini korumalarında önemli rol oynamaktadır. Böylelikle açığa düşerek daha yüksek maliyetli kredilere yönelmek durumunda kalmazlar. Fon arzında bulunan taraf ise ilgili tutarı elinde tutmak yerine faiz getirisi elde etme fırsatını yakalamış olmaktadır. Şirketler bu amaç için gecelik olarak para piyasasına yönelebilirler.

Diğer bir yandan yatırımcılar para piyasalarında daha güvenilir bir biçimde yatırım yapmak amacıyla işlem yapmaktadır. Sermaye piyasalarına kıyasla para piyasalarında risk azdır ve riskten kaçınan yatırımcılar para piyasasına likidite sağlamak amacıyla başvurmaktadır.

Sermaye piyasaları tasarruf ve yatırımların sermaye sahibi (arz edenlerle) sermayeye ihtiyacı olanlar arasında el değiştirmesine olanak sağlayan piyasalardır. Sermaye sahibi aktörler kurumsal yatırımcılar olup sermaye talebi olanlar hükümetler, insanlar ya da iş çevreleri olabilmektedir.

Sermaye piyasaları birincil ve ikincil piyasadan oluşmaktadır. En çok bilinen sermaye piyasaları hisse ve borçlanma araçları piyasalarıdır. Sermaye piyasaları işlemlerin etkin bir biçimde gerçekleşmesini amaçlamaktadır. Bu piyasalar sermaye sahipleri ve sermayeye ihtiyacı olanları bir araya getirerek menkul kıymetlerin el değiştirdiği güvenilir bir ortam yaratmaktadır (Tümer, 1979).

Sermaye piyasaları en geniş anlamda çeşitli kurumların farklı finansal enstrümanlara yönelik işlem yaptıkları piyasalar anlamına gelmektedir. Bu platformlar hisse, borçlanma araçları, vadeli işlemler gibi piyasalardan oluşmaktadır. En önemli piyasalar New York, Londra, Singapur ve Hong Kong'da yer almaktadır.

Sermaye piyasaları sermaye arz edenler ve bu sermayeyi kullanan aktörlerden oluşmaktadır. Sermaye arz edenler hane halkları, emeklilik fonları, hayat sigorta şirketleri ve elinde ihtiyacından fazla nakit bulunduran finansal olmayan kuruluşlardır. Bu fonları kullananlar ise alt yapı yatırımları ve operasyonel harcamalarını gidermek amacıyla hükümetler ve finansal olmayan kuruluşlardır (Afşar, 2006).

Bu piyasalar menkul kıymetlerin ihraç edilerek yatırımcılara ilk satıldığı birincil piyasalar ve ilk ihracın gerçekleştiği birincil piyasanın ardından menkul kıymetin sahibi tarafından farklı kişilere satıldığı ikincil piyasalardan oluşmaktadır.

Semaye piyasalarında faaliyet gösteren kurumlara birleşme yada satın alma, yeni bir iş kolu kurma ya da diğer sermaye projeleri için kullanılmak üzere uzun vadeli sermaye oluşturmak amacıyla başvurulmaktadır. Bu gibi uzun vadeli amaçlara hizmet etmek için kurumlar sermaye oluşturmaktadır. Borçlanma araçları piyasasında şirketler özel sektör tahvili; hükümetler ise devlet tahvili şeklinde borçlanma araçları ihraç edebilmektedir.

Benzer şekilde şirketler hisse ihraç ederek hisse piyasasında da sermaye oluşturabilir. Hükümet kuruluşları halka açık olmadığı için hisse senedi ihraç etmezler. Borçlanma aracı ya da hisse ihracı yapan şirket ve hükümet kuruluşları bu piyasalarda satıcı olarak adlandırılmaktadır.

Alıcı yani yatırımcılar satıcılar tarafından ihraç edilen hisse ya da borçlanma araçlarını alarak işlem yapanlardır. Eğer ihraççı tarafından ihraç edilen menkul kıymetler piyasada ilk kez işlem görüyorsa bu piyasa birincil piyasa olarak adlandırılmaktadır.

Tam tersine ihraç edilmiş kıymetler alıcılar arasında işlem görüyorsa bu da ikincil piyasada gerçekleşmektedir. Satıcılar yani ihraççılar menkul kıymetlerin birincil piyasadaki satışından kazanç elde etmekte olup ikincil piyasadan herhangi bir getiri elde etmemektedir. Ikincil piyasadaki menkul kıymet fiyatlandırmasında da ihraççıların herhangi bir etkisi bulunmamaktadır.

Menkul kıymet alıcıları bu fonları uzun vadeli yatırım amacıyla kullanmaktadır. Sermaye piyasaları riskli piyasalar olup kısa vadeli fonların kullanıldığı bir piyasa değildir. Birçok yatırımcı sermaye piyasalarına eğitim ya da emeklilik için tasarrufta bulunmak amacıyla başvurmaktadır.

3- BORSALAR ve ENDEKSLER

Borsalar her türlü değerli menkul kıymetin, payın, tahvilin ya da emtianın kote olarak işlem gördüğü ve devletin ilgili gözetim ve denetim biriminin denetiminde olarak işleyen organize piyasalardır. Kotasyon öncelikli birinci piyasada gerçekleşmektedir. Bu piyasa şirketlerin paylarının bir kısmının ya da tamamının halka arz edildiği (IPO) piyasadadır. Şirketler genellikle yeni pay çıkararak kaynak tesis etmek amacıyla halka arz edilmektedir. Yatırımcılar bu halka arz edilmiş kıymetleri ikincil piyasada kendi aralarında alım satım yapmaktadır. Birincil piyasada kotasyon yatırımcılardan şirkete doğru bir kaynak hareketi gerçekleştirirken; yatırımcılar arasındaki işlemler böyle bir harekete yol açmaz.

Hem birincil piyasa hem de ikincil piyasadaki faaliyetler adil, saydam ve kurallara uygun bir şekilde işleyen piyasaların devamlılığı açısından belirli kural, kanun ve yönetmelikler çerçevesinde gerçekleşmektedir. Bu amaçların elde edilmesi yatırımcının varlıklarını korumak, payların sahipliğinin el değiştirmesini sağlamak ve işlem görmek isteyen şirketlerin taşıması gereken bazı kriterler sayesinde sağlanmaktadır.

Borsalar aynı zamanda işlemlerin önceden belirlenmiş, kamuya açık kurallar kapsamında piyasada alım ve satım yapılmak istenen fiyatların açık bir şekilde yayınlandığı, işlem öncesi ve sonrası saydamlık ilkelerine uyumlu bir ortam sağlamaktadır. Borsalar sermaye piyasalarını denetleyen düzenleyici ve denetleyici kurumlar tarafından denetlenmektedir. Piyasa düzenlemelerinden kimin sorumlu olduğu bölgeden bölgeye değişkenlik gösterebilir. Ancak borsadaki tüm piyasaların kotasyon şartları (borsada işlem görmek isteyen şirketlerin ilk başta ve sonrasında devam edecek şekilde uyması gereken kurallar ve kamuya duyurması gereken bilgileri); üyelik şartları (finansal, eğitim açısından müşterileri adına borsada işlem yapacak aracı kurumların uyması gereken kurallar); işlem kuralları (piyasada alım satım emirlerinin eşleşme kriterleri, fiyat belirleme süreçleri, hata olması durumunda ne yapıldığı, işlemin gerçekleştiğinde ne olacağına ilişkin kurallar); borsada

gerçekleşen işlemlerin takas ve mutabakatının nasıl yapılacağı (pay sahipliğinin ve nakdin el değiştirmesi, bunların hangi sürede olacağı ve temerrüt olması durumundaki kurallar) gibi süreçlerin önceden belirli olması gerekmektedir. Düzenleme yapıları ne olursa olsun tüm borsalar piyasa düzenlemesinin yukarıda bahsedilen noktalarla uyumlu olmasını sağlamak ve piyasa gözetim faaliyetlerini gerçekleştirmek için sorumludur.

Borsalarda menkul kıymetler çoğu zaman farklı fiyatlar üzerinden alınıp satılmaktadır. Bu sebeple menkul kıymet fiyatları dalgalanabilmektedir. Kıymetlere yönelik genellikle açılış, kapanış, en yüksek ve en düşük fiyat bilgileri borsalar tarafından sunulmaktadır. Borsa endeksi tüm piyasayı, belirli bir sektörü ya da piyasanın herhangi bir kesimini kapsama özelliğine sahip olan hisselerin seçilmesiyle oluşturulmaktadır. Bu sepetteki hisselerin fiyat değişimleri baz dönemle kıyaslanarak hesaplanmaktadır. Genellikle ekonomideki önemlerine göre hisseler farklı şekilde ağırlıklandırılmaktadır. Borsa endeksleri tüm ekonomonin genel performansını yansıtması açısından önem taşımaktadır.

Endeks oluştururken önemli olan faktörler; endekste yer alan kıymetler, ilgili kıymetlerin ağırlığı ve hesaplama yöntemidir. Örneğin dünyaca ünlü Dow Jones Sanayi Endeksi (DJIAA) 30 en kurumsal, en karlı ve en büyük şirketlerin hisselerinden oluşmaktadır. DJIAA endeksi 30 hissenin fiyat ağırlıklı ortalaması yöntemiyle hesaplanmaktadır. Bir diğer önemli endeks ise S&P 500 endeksidir. Bu endekste 500 adet hisse senedi bulunmaktadır ve hesaplama yöntemi olarak piyasa değeri ağırlıklı yöntemi kullanılmaktadır.

Genel ve endüstri bazında endekslere ek olarak küresel hisse senedi endeksleri de bulunmaktadır. Bunlar arasında 30 ülkenin yer aldığı FT/S&P Actruaries Dünya endeksleri, Morgan Stanley Capital International Indexes (MSCI) ve 33 ülkenin yer aldığı Down Jones World Stock Index endeksler bulunmaktadır.

3.1 Dow Jones Endüstri Ortalaması

Dow Jones Endüstri ortalaması endeksi dünyada en çok takip edilen endekslerden birisidir. Endeks değişimlerinin pay piyasasının genel durumunu yansıttığı belirtilmektedir. Dow Jones & Co. Şirketi kurucularından Charles Dow ilk kez endeksi oluşturmuştur. 1884 yılında 11 adet likit ve piyasa değeri yüksek hisseyi kullanmıştır. 26 Mayıs 1896'da DJIAA endeksi ilk kez yayınlanmıştır.1920 yılında

endekse dahil pay sayısı 20'ye yükseltilmiş Kasım 1928'de bugünkü hali olan 30'a ulaşmıştır.

Dow Jones Endüstri Ortalaması Endeksi NYSE ve NASDAQ'da işlem gören halka açık 30 şirketin hissesinden oluşmaktadır. Endeks adı 1896 yılında endeksi ilk oluşturan Charles Dow ve Edward Jones tarafından verilmiştir. Dow endeksi olarak da belirtilen endeks Walt Disney, Exxon, Microsoft gibi çok büyük şirket hisselerini de içerisinde bulunduran dünyanın en eski ve en çok takip edilen endekslerinden biridir. Endeks hesaplamasında hisse fiyatları yüksek olan hisselerin endeksteki ağırlıkları daha yüksektir. Bu sebeple yüksek fiyatlı bileşen kaynaklı bir yüzdelik değişim hesaplanan değere daha fazla etki edecektir. Charles Dow tarafından endeks ilk olarak 12 adet hissenin fiyatlarının toplanarak 12'ye bölünmesi şeklinde ortalama olarak hesaplanmaktaydı. Zaman içerisinde şirket birleşimi, hisse bölünmesi gibi endekse eklemeler ve çıkarmalar hesaplama yönteminde basit ortalamanın yetersiz olduğunu ortaya çıkarmıştır. Bu sayede önceden belirlenen Dow Divisor endeksi oluşturan 30 şirketten herhangi birindeki bir birimlik değişimin etkisini ortaya çıkarmak amacıyla oluşturulmuştur. DJIAA değerinin tutarlı olması adına hisse bölünmesi, hisselerin eklenip çıkarılması gibi durumlarda Dow divisor değerinin de değiştirilmesi gerekmektedir.

DJIAA endeksi ağırlıklı aritmetik ortalama ya da S&P500 gibi endeksi oluşturan şirketlerin piyasa değerini belirten bir endeks değildir. Endeks her bir bileşenin hisse başı fiyatlarının toplamının dow divisor'a bölünmesi ile hesaplanmaktadır. Böylelikle endeksi oluşturan her bir bileşen endeksi benzer puanda kaydıracaktır. Çizelge 1'de DJIAA endeksini oluşturan hisselere yer verilmiştir.

Çizelge 1 : DJIAA şirketleri ve ağırlıkları.

Sıra	Şirket	Endeks Ağırlık %
1	Boeing	9.03
2	UnitedHealth Group	6.42
3	McDonalds	5.33
4	Goldman Sachs	5.22
5	Home Depot	5.1

6	Apple	4.97
7	3M	4.66
8	Visa A	4.21
9	Travelers	3.84
10	Johnson & Johnson	3.67
11	United Technologies	3.53
12	IBM	3.52
13	Disney	3.52
14	Caterpillar	3.36
15	Microsoft	3.3
16	Chevron	3.23
17	American Express	3.1
18	JPMorgan Chase	2.93
19	Procter & Gamble	2.84
20	Walmart	2.68
21	NIKE B	2.21
22	Merck	2.07
23	ExxonMobil	2.03
24	Verizon Communications	1.52
25	Walgreens Boots Alliance	1.4
26	Cisco Systems	1.37
27	Dow	1.36
28	Coca-Cola	1.29

29	Intel	1.2
30	Pfizer	1.09

3.2 Nasdaq 100 Endeksi

Nasdaq 100 endeksi Nasdaq'ta işlem gören ve piyasa değerine göre en büyük 100 finansal olmayan yerel ya da yabancı şirketten oluşmaktadır. Endeks biyoteknoloji, bilgisayar donanımı, telekomünikasyon, yazılım gibi çok büyük ve çeşitli endüstri alt dallarını içermektedir. Yatırım şirketlerinin hisseleri endekste yer almamaktadır.

Nasdaq 100 endeksinde yer alabilmek için şirketlerin minimum aşağıdaki kriterleri sağlaması gerekmektedir ("Nasdaq" t.y.).

- Ilgili hissenin öncelikli olarak Nasdaq Global Market ya da Nasdaq Global Select Market'te işlem görmesi gerekmektedir.
- Hisse finansal olmayan bir şirket tarafından ihraç edilmelidir.
- Hissenin iflas sürecinde olan bir ihraççı tarafından ihraç edilmemesi gerekmektedir.
- Şirketin günlük ortalama işlem hacmi en az 200,000 nominal olmalıdır.
- Şirketin ilgili hissenin endekste kalmasını engelleyecek herhangi bir sözleşmeye taraf olmaması gerekmektedir.
- Şirketin Nasdaq, NYSE, NYSE AMEX'te ilk işlem gördüğü ay hariç en az 3 ay kote olması gerekmektedir.

4- ALGORİTMİK İŞLEMLER

Algoritma bir işi yapmak ya da bir problemi çözmek için tasarlanan yol anlamına gelmektedir. Algoritmik işlemler (automated trading ya da black-box trading), bilgisayarların insanlar için imkânsız olan hız ve frekansta kâr elde edecek şekilde emir girmeleri için önceden belirlenen talimatlar doğrultusunda programlanmasıdır ("Nasdaq" t.y.). Bu kurallar zamanlama, fiyat, miktar ya da herhangi bir matematiksel modele göre oluşturulabilir. İşlemci için kâr fırsatlarına ek olarak algoritmik işlemler piyasaları daha likit hale getirir ve işlem aktivitelerindeki duygusal insan etkilerini bir kenara koyarak işlemlerin daha sistematik hale gelmesini sağlamaktadır. İşlemcinin, hissenin 50 günlük hareketli ortalamasının 200 günlük hareketli ortalamasından yüksek olduğu durumda 50 adet hisse aldığı; 50 günlük hareketli ortalamasının 200 günlük hareketli ortalamasından düşük olduğu durumda ise 50 adet hisse sattığını varsayarsak bu iki basit talimatı kullanarak bilgisayar programına hisse fiyatını ve hareketli ortalama göstergelerini otomatik olarak izlemesini ve belirlenen şartlar gerçekleştiğinde alım/satım emirlerini iletmesine yönelik bir algoritma yazılabilir. Bu sayede işlemci güncel fiyatlar ve grafikleri takip etme ve manuel olarak emir iletme zorunluluğundan kurtulacaktır. Algoritmik işlem sistemleri işlemci için otomatik olarak bu süreçleri gerçekleştirmektedir.

Algoritmik işlemlerin avantajları ("Nasdaq" t.y.).

- İşlemler mümkün olan en iyi fiyattan gerçekleşir.
- Anlık ve doğru emir iletimi, bu sayede istenen düzeyde işlem gerçekleştirme ihtimali artar.
- Emirler doğru zamanda ve anlık olarak iletilir ve bu sayede önemli fiyat değişikliklerinin önüne geçilmiş olur.
- İşlem ücretleri azalır.
- Çoklu piyasa şartları üzerinde eşzamanlı otomatik kontroller gerçekleştirilir.
- Emir girişi esnasında olabilecek manual hataların önüne geçer.

- Mevcut geçmiş tarihli ve gerçek zamanlı veriler üzerinden algoritmanın geriye dönük testleri yapılabilir.
- Duygusal ve psikolojik faktörlerden kaynaklanan insan hataları ihtimallerini azaltır.

Günümüzde algoritmik işlemlerin büyük çoğunluğu yüksek frekanslı işlemler (HFT) şeklindedir. Algoritmik işlemler günümüzde aşağıda yer verilen birçok işlem ve yatırım aktivitelerinde sıklıkla kullanılmaktadır: ("Investopedia" t.y.)

- Orta- uzun vadeli yatırım yapan yatırımcılar ya da emeklilik, yatırım fonları ve sigorta şirketleri gibi yüksek miktarda hisse alan; ancak hisse fiyatlarını kesikli yüksek hacimli alımlarla etkilemek istemeyen alış tarafı şirketleri
- Kısa vadeli satım yapan işlemciler ve piyasa yapıcılar, spekülatörler ve arbitraj yapan satış tarafındaki piyasa katılımcıları otomatik işlem gerçekleştirmeden faydalanmaktadır. Ayrıca algoritmik işlemler piyasadaki satıcılara likidite sağlamaktadır.
- Piyasa trendini takip eden yatırımcılar ya da serbest fonlar gibi sistematik işlemcilerişlem kurallarını programlayarak işlemlerini otomatik olarak iletmeyi daha etkin bulmuşlardır.

Algoritmik işlemlerde izlenecek herhangi bir strateji kâr elde etmek ya da maliyetleri minimize etmek amaçlı kullanılabilir. Algoritmik işlemlerde sıklıkla kullanılan stratejiler aşağıda yer almaktadır (Mcgroarty, Booth, Gerding& Chintalapati, 2018).

1- **Trendi takip eden stratejiler**: Algoritmik işlemlerde kullanılan çoğu strateji fiyat seviyeleri, ağırlıklı ortalamalar, teknik göstergelerdeki trendi takip etmektedir. Bu algoritmik işlemlere uygulanabilen en kolay ve basit stratejilerdendir çünkü herhangi bir fiyat tahmini içermemektedir. İşlemler beklenen trendlerin gerçekleşmesiyle gerçekleştirilir ve tahmin analizine yer verilmemektedir.

2- **Arbitraj fırsatları**: Çifte kote olan bir hisseyi bir piyasadan ucuza alıp diğer piyasada eş zamanlı olarak daha yüksek fiyata satarken arbitraj imkânı doğmaktadır.

3- **Endeks fon yeniden dengeleme:** Endeks fonlar portföylerini belirli dönemlerde ilgili karşılaştırma ölçütlerinin oranına getirmek için yeniden dengeleme yapmaktadır. Bu durum algoritmik işlemciler için endeks fondaki hisselerinin sayısına bağlı olarak yeniden dengelemenin hemen öncesinde 20-80 baz puan kâr sağlamaktadır. Bu tarz işlemlerin algoritmik işlemlerle yapılmasının amacı zamanında en iyi fiyattan işlem gerçekleştirmektir.

5- LİTERATÜR ARAŞTIRMASI

Finansal piyasalar lineer olmayan dinamik doğasından dolayı öngörülmesi zor piyasalardır. Özellikle günümüzde gelişmiş iletişim teknolojileri ve kişilerin iletişim teknolojilerini yoğun olarak kullanması ile piyasalara olan etkiler gerçek-zamanlı olarak gözlenebilmektedir. Aynı zamanda bilgisayarların hızlanması ve internetin yayılması ile birlikte oluşan veri yığınları "veri madenciliği" adı altında işlenip tutarlı veriler üretilmektedir. Bu verilerin analiz edilmesi için gelişmiş istatistiki ve teknik analiz yöntemleri geliştirilmiştir.

Bu bölümde makine öğrenmesi ve gelişmiş istatistiki yöntemler ile finansal piyasaların tahminini sağlayan birtakım yöntemlerin literatürdeki yeri açıklanmıştır.

5.1 Makine Öğrenmesi ile Finansal Piyasaların Tahmin Edilmesi

Makine öğrenmesi büyük bir veri yığını içerisindeki örüntüleri algılamak için kullanılmaktadır. Bu bölümde çok sayıdaki makine öğrenme algoritmasından en sık kullanılan dördü açıklanmıştır. Bu algoritmalar sayesinde finansal piyasalardaki değişimleri, geçmişteki ve şu anda var olan veriler analiz edilerek gelecekteki değerleri, yükseliş veya düşüşte oldukları hesaplanmaktadır.

DVM (Vapnik ve Cortes,1995) sınıflandırma ve regresyon analizinde kullanılan istatistiksel bir denetimli öğrenme modelidir. DVM finansal zaman serilerinin tahmini için ilk olarak Tay ve Cao tarafından kullanılmıştır (Cao ve Tay, 2001). L. J. Cao ve Francis E. H. Tay tarafından Standard & Poor 500 endeks vadeli işlem sözleşmeleri (CME-SP), ABD 30-yıllık DİBS (CBOT-US), ABD10-yıllık DİBS (CBOT-BO), Almanya 10-yıllık DİBS (EUREX-BUND), ve Fransız hisse endeks vadeli sözleşmeleri (MATIF-CAC40) 1988-1999 yılları arasında ilgili tarih aralığındaki kapanış değerleri kullanılarak DVM ve YSA yöntemleri ile tahmin edilmeye çalışılmıştır (Cao ve diğerleri, 2001). Girdi olarak EMA-100 ve 5 günlük fiyat değişim yüzdesi kullanılmıştır. Performans ölçütü olarak ortalama mutlak hata ve ortalama kare hata ve yönsel simetri kullanılmıştır. Sonuç olarak DVM algoritması YSA'na

kıyasla daha yüksek bir tahmin performansı göstermiştir. Kim (2013) ve Sewell (2010) teknik analiz göstergelerini kullanan DVM modeli ile hisse fiyatlarının yönünü tahmin eden bir algoritma geliştirmiştir. Kyoung-jae Kim tarafından KOPSI endeksinin günlük fiyat değişikliğinin yönünü tahmin etmede DVM algoritması uygulanmış, sonuçları geriye yayılımlı YSA ve CBR (case based reasoning) yöntemleriyle kıyaslanmıştır. Modelde girdi olarak teknik göstergeler kullanılmış olup sonuç olarak hem test verilerinde hem de eğitim verilerinde DVM algoritması daha yüksek doğruluk göstermiş olup Cao ve Tay tarafından yapılan araştırmaya uygun sonuçlar elde edilmiştir.

DVM uygulamaları finansal verilerin indeks tahmini üzerinde başarılı sonuçlar vermektedir. Van Gestel v.d.(2001) ikincil piyasadaki 90 günlük hazine bonosu faizlerini ve DAX 30 endeksini DVM kullanarak tahmin etmiştir. Huang v.d.(2005) tarafından DVM yöntemi kullanılarak pay piyasası tahmin modeli önerilmiştir. Kendi tahmin modellerini Doğrusal Diskriminant Analizi, Karesel Diskriminant Analizi ve Elman Geri Yayılımlı Yapay Sinir Ağı metotları ile karşılaştırmıştır. Ayrıca DVM ve diğer sınıflandırma teknikleri ile birlikte yeni bir tahmin modeli oluşturmuşlardır. Huang, W., Nakamori, Y., Wang, S.-Y tarafından yapılan çalışmada NIKKEI 225 Endeksi USD/JPY ve S&P 500 değerleri girdi olarak kullanılarak haftalık olarak tahmin edilmiştir. 1990-2002 yılları arasında yapılan 676 gözlemden 640'ı öğrenme; 36'sı test için kullanılmıştır. DVM modelinin sınıflandırma başarısını ölçmek amacıyla Rassal Yürüyüş, LDA, QDA, EBNN modelleri kullanılmış doğruluk oranı en fazla DVM modelinde elde edilmiştir. Sewell (2010) makalesinde DVM ve türevlerinin diğer yöntemlerden gözle görülür derecede başarılı olduğunu belirtmiştir.

Lin v.d.(2013) DVM modelini kullanarak pay piyasası tahmin modeli önermiştir [19]. Bu model kaliteli bir özellik alt kümesi seçer ve bu veri kümesi üzerinden hisse senedi göstergesini ve borsa eğim beklentisine uyumunu değerlendirir. Yuling Lin, Haixiang Guo, Jinglu Hu tarafından yapılan çalışmaya istinaden Taiwan hisse senedi tahmini DVM modeli kullanılarak yapılmış, modelde kullanılan girdiler özellik seçimi parametreleri ve korelasyon tabanlı DVM filtresi uygulanarak doğru bir şekilde kullanılmıştır. Sonuca göre DVM modeli daha yüksek bir tahmin performansı elde etmiştir. Hu v.d.(2009) ve Ding v.d.(2008) DVM kullanarak Çin piyasasındaki firmaların finansal durumlarını ve finansal krizi tahmin modeli geliştirmişlerdir. Bu çalışmaya benzer başka çalışmalar da mevcuttur. Emir, Dinçer ve Timor ve Ruei-

Shan(2008) Avustralya ve Tayvan piyasasındaki en karlı hisseleri sıralamak için DVM sınıflandırıcıları kullanmışlardır. Eğitim verisi olarak tüm piyasa verilerini, pozitif sınıf en değerli %25 hisse olmak ve negatif sınıf geri kalan %75 olarak sınıflandırarak, çalışmalarında kullanmışlardır. Yazarlar sadece alım içeren eşit ağırlıklandırılmış portföylerde fazla getiri elde etmekte, ancak portföylerde likidite ve büyüklüğü kontrol etmemektedirler. Mali verilerde SVM modeli uygulaması konusunda yazılmış makalelerin birçoğunda, verilerin düzenlenmesi için bir ön parametre seçilmesinin sonuçları etkilediğini gösterilmektedir. Bu parametreler seçilirken ileriye dönük olarak düşünülmemesi gerekmektedir; yoksa bu değerler ileriye dönük hatalı veriler üretmeye neden olur. Bu sorunu önlemek için, parametreler geçmişteki verilerden yola çıkılarak seçilmelidir. Fan, A., Palaniswami, M. tarafından yapılan çalışmada 2002-2010 yılları arasındaki BIST 30 performansı temel ve teknik analiz parametreleri kullanılarak tahmin edilmiş, bu amaçla yıl sonu itibariyle en iyi getiri elde edilen ilk 10 hisse 1, diğerleri 0 olarak etiketlendirilmiştir. Modelde DVM ve YSA algoritmaları kullanılmış, en iyi performansı DVM algoritması ortaya koymuştur.

DVM ile diğer modeller birlikte kullanılarak hibrit çözümler oluşturulabilmektedir. Shom Prasad Das ve Sudarsan Padhy(2012) Geri Yayılımlı Yapay Sinir Ağı modeli ve DVM modelini entegre ederek Hindistan piyasasındaki opsiyon fiyatlamasını tahmin etmişlerdir. Shom Prasad Das and Sudarsan Padhy tarafından yapılan çalışmada S&P CNX NIFTY, BANK NIFTY, S&P CNX 500, CNX INFRA ve CNX 100 endeks vadeli sözleşmeleri 2007-2010 yılları arasında açılış, kapanış, yüksek ve düşük değerleri ile alınmıştır. Girdi olarak teknik göstergelerden olan günlük fiyat değişim yüzdesi ve EMA15 kullanılmıştır. Modelde hem DVM hem de kıyaslama ölçütü olarak YSA kullanılmış olup sonuç olarak DVM algoritması yapay sinir ağlarına göre daha iyi bir tahmin elde etmiştir.

DVM sadece sınıflandırma için değil regresyon analizi için de kullanılmaktadır. Huerta v.d.(2012) olası yüksek getirileri olan hisse senetlerini belirlemek için bir model eğitmeye çalışmışlardır. DVM, değişen piyasa koşullarına uyum sağlamak için ayda bir kez eğitilmektedir.

DVM yönteminin en büyük eksiklerinden biri, girdi olarak aldığı verilerin çok boyutlu ve çok sayıda olmasından dolayı büyük veriler olmasıdır. Veri büyüdükçe DVM modelinin uygulanıp sonuç elde edilmesi uzun sürmektedir. Bu yüzden verinin boyutu, ilgili veri analiz edilip anlamlı veri elde edilerek küçültülür ve normalize edilirse hem

elimizde nitelikli bir veri kümesi olur hem de DVM modelinin uygulanma süresi azalır.

Kara, Boyacioglu, & Baykan tarafından 1997-2007 yılları arasında BIST 100 endeksi DVM ve YSA algoritmasıyla teknik göstergeler girdi olarak kullanılarak yıllık olarak tahmin edilmiş; sonuç olarak YSA daha yüksek bir doğruluk elde etmiştir (Kara, Boyacıoğlu& Baykan, 2011).

Shen, Shunrong, Jiang, Haomiao, & Zhang, Tongda tarafından NASDAQ, S&P ve DJIAA endeksleri SVM kullanılarak tahmin edilmiş, girdi olarak EUR, AUD, JPY, USD, gümüş, platin, altın, DAX, ASX, FTSE100 kullanılmıştır. Modelde doğruluk NASDAQ için %76, S&P 500 için 76 ve DJIAA için 77.6 olarak ölçülmüştür. (Shen vd., 2009).

Tsai, C.-F. and Wang, S.-P. tarafından yapılan çalışmada hisse fiyatları YSA ve karar ağaçları algoritmaları kullanılarak tahmin edilmeye çalışılmış; karar ağaçları ve YSA algoritmasının birlikte kullanıldığı karma modelin doğruluğu %76 olarak ölçülmüş olup her iki modelin tekli performansından yüksektir. (Tsai ve Wang, 2009).

De Oliviera tarafından yapılan analizde hem teknik analiz hem de ekonomik değişkenler modele dahil edilerek PETR4 hissesi tahmin edilmiş ve ANN modelinin başarısı 87.50 olarak ölçülmüştür (De Oliviera, Nobre, Zarate, 2013).

Lu tarafından yapılan çalışmada Nikkei 225 ve TAIEX endeksleri tahmin edilmiş girdi olarak teknik göstergeler kullanılmıştır. Nikkei için SVR modeli TAIEX için ise ICA-SVR modeli en başarılı şekilde tahmin etmiştir (Yu, Wang, Lai, 2009).

Son zamanlarda finansal hizmet sunan şirketler ürünlerini geleceği tahminlemek amacıyla sunmaktadır. Bilgiye ulaşmanın kolaylaşması ve çok sayıdaki finansal bilgi kaynakları yeni araştırma alanların ortaya çıkmasına sebep olmuştur. Aralarında hisse fiyatı tahminleme ve finansal piyasalar için veri madenciliğinin de bulunduğu bu alanda doğru kuralların oluşturulması farklı piyasalarda daha karlı yatırım yapma olanağı sağlayacaktır.

KNN algoritması veri dağılımı hakkında az ya da hiçbir ön bilginin olmadığı durumlarda kullanılan bir sınıflandırma tekniğidir (Devroye, 1978). Fuzzy genetik algoritması kullanıcı tercihlerine dayanarak hisse verisindeki ilişkileri ortaya çıkarmak amacıyla Cao tarafından 1977 yılında kullanılmıştır (Cao, 1997).

KNN algoritmasının basitlik, verimlilik ve yüksek sınıflandırma performansı bulunmaktadır. Gürültülü eğitim datası ve veri boyutunun büyük olduğu durumlarda etkilidir. Belirtilen avantajlarına rağmen KNN algoritmasının büyük verilerde çalışma zamanı performansı düşüktür ve algoritma alakasız ya da gereksiz özellikler seçimlerinde, tüm özelliklerin sınıflandırma sürecine dahil olması sebebiyle etkili değildir. Bu sebeple algoritmaya girecek olan özelliklerin seçilim ve ağırlıklandırması önem kazanmaktadır. Son olarak KNN algoritmasının hesaplama maliyeti oldukça yüksektir.

Kantitatif analiz hisseleri rastsal değişkenler olarak görmekte, matematik ve istatistik kantitatif analiz kullanarak bu rastsal değişkenlerin nasıl hareket edeceğini gözlemlemektedir. Temel analiz etkin piyasalar hipotezinin yanlış olduğu varsayımı üzerine kurgunlanmıştır. Temel analize göre olması gerekenden daha fazla ya da daha az değerlenmiş hisseler bulunmaktadır. Hisse düşük değerlenmişse hissenin fiyatı, şirketin fiyatını yansıtmayacaktır.

Refenes v.d. Londra borsası hisselerinin tahmini için sinir ağları test etmiş; sonuçlarını klasik istatistik tekniklerle kıyaslamıştır (Refenes, Zapranis& Francis, 1994). YPA altı aylık verilerle 143 hisse üzerinden test edilmiş ve sonraki 6 ay değerlendirilmiştir. Bu döngü Mayıs 1985'ten Aralık 1991'e kadar tekrar edilmiştir. Sonuç olarak gizli katman miktarı ve nöronların karar verici faktörlerden olmadığı ortaya çıkmıştır. karekök ortalama hatası farklı değerler için 0.044 ile 0.072 arasında değilşkenlik göstermiştir. Çalışma sonucunda istatistik yaklaşımların YPA yaklaşımının gerisinde kaldığı belirtilmiştir.

Molin ve Suljkic Stokholm, Kore ve Barselona gibi farklı piyasaların tahmin edilebilirliğini YPA kullanarak karşılaşırmıştır. Farklı piyasalar için farklı sonuçlar elde ederek Stokholm'ün diğer piyasalara nazaran daha tahmin edilebilir olduğunu ortaya çıkarmışlardır. Otoregresif Hareketli Ortalama (ARMAX) metodu karşılaştırma amacıyla kullanılmıştır. (Suljkic, 2014).

Niaki ve Hoseinzade 27 finansal ve ekonomik faktörü girdi olarak kullanarak S&p 500 indeksini YPA kullanarak tahmin etmeye çalışmıştır. Piyasa yönüne göre alım satım stratejisi geliştirmişlerdir. Stratejilerine göre ANN tarafından tahmin yapılmış,

portföyleri bu tahminlerine göre yeniden oluşturulmuştur. YPA'nın logit modelinden daha iyi sonuçlar elde ettiğini göstermişlerdir (Niaki ve Hoseinzade, 2013).

O"Connor ve Madden tarafından farklı YPA modelleri farklı kurgularla Dow Jones endüstri endeks hareketlerini tahmin etmek amacıyla kullanılmıştır. 6 adet kurgu tasarlanmıştır. Her bir kurguda farklı girdiler test edilmiştir. Bazı kurgularda kur ve ham petrol gibi dış faktörler girdi olarak modele sokulmamış; onların yerine Dow Jones zaman serisi verileri ve ilgili teknik göstergeler girdi olarak kullanılmıştır. Sonuçlara göre dış faktörler girdi olarak modele sokulduğunda karlılık gibi modelin genel performansı daha başarılı sonuç vermiştir (O'Connor ve Madden, 2006).

Huang, Nakamori ve Wang NIKKEI 225 endeksinin haftalık hareket yönünü . Elman tipi geri beslemeli YPA, rastsal yürüyüş modeli, lineer diskriminant analizi, karesel diskriminant analizi ve diğer sınıflandırma metodları kullanarak tahmin etmiş, sonuçlarını karşılaştırmıştır (Huang vd., 2004). Kumar ve Thenmozhi S&P CNX NIFTY endeksini farklı yöntemlerle tahmin etmiştir. Lineer diskriminant analizi, YPA, logit modeli, rastsal orman teorisi ve DVM yöntemi kıyaslanmıştır (Kumar ve Thenmozhi, 2006).

6- METODOLOJİ

6.1 Veri Madenciliği ve Makine Öğrenmesi

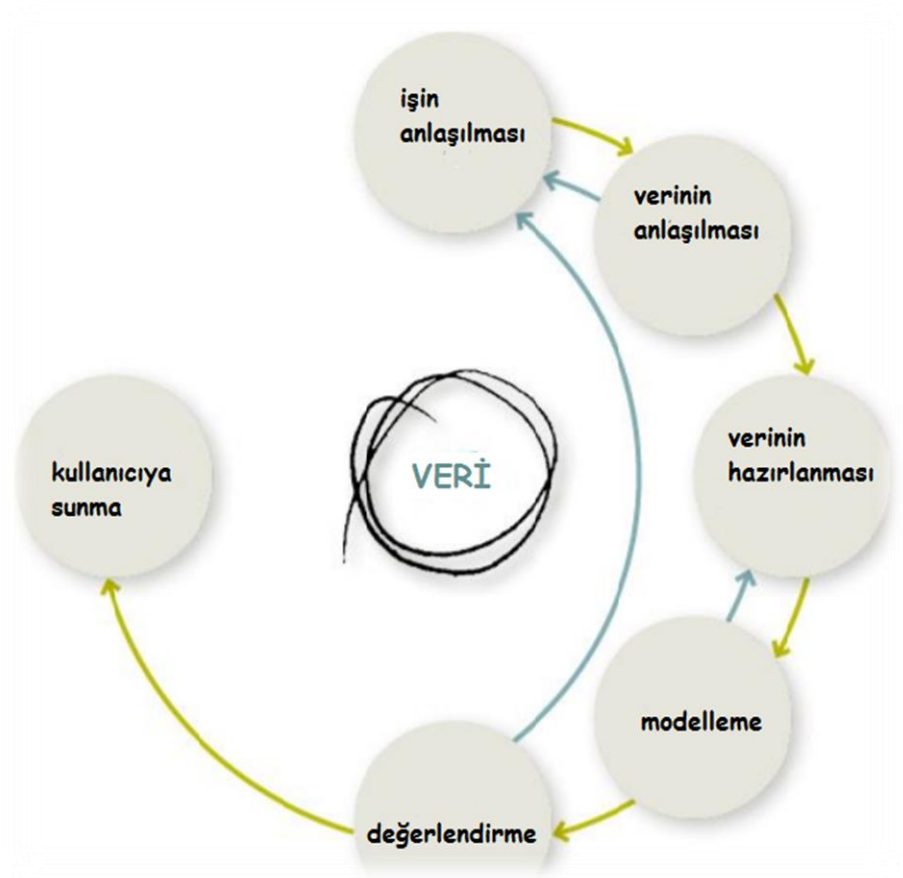
Veri madenciliği yapılandırılmış veri tabanlarında bulunan verilerden geçerli, yeni, kullanılabilir ve anlaşılabilir örüntülerin çıkarılmasını sağlayan zorlu bir süreçtir (Öğücü, 2006).

Veri madenciliği sayesinde şirketler ham veri işleyerek işe yarayan bir bilgiye dönüştürebilmektedir. Büyük veri kümelerinde örüntüler elde etmek amacıyla yazılım kullanarak şirketler müşteri memnuniyetini ve satışlarını artırabilir, maliyetlerini azaltabilmektedir. Veri madenciliği için veri toplama, veri ambarı ve bilgisayarın verileri işlemesi önem taşımaktadır. Veri madenciliği sürecinde veri bilimciler tarafından kullanılan iki standart süreç bulunmaktadır.

SAS Institute tarafından geliştirilen SEMMA (Sample, *Explore, Modify, Model, Assess*) modeli veri madenciliği projelerinde kullanılmak üzere tasarlanmıştır. Sürecin ilk aşaması olan örnekleme aşamasında seçilen veri önemli bilgileri içerecek kadar büyük; uygulaması kolay olacak kadar küçük olmalıdır. Araştırma aşamasında veriye bakarak fikir elde edilmesi amacıyla beklenmedik ilişkilerin olup ve anormalliklerin olup olmadığı kontrol edilmektedir. Dönüştürme aşamasında model seçme sürecine uyması açısından değişkenlerin belirlenerek seçilmesi ve dönüştürülmesi sürecidir. Modelleme sürecinde istenen sonucun güvenilir bir şekilde tahmin edilmesi amacıyla yazılımın ilgili veri kombinasyonlarını otomatik olarak bulması sürecidir. Son süreç olan değerlendirme aşamasında VM süreci kullanılarak elde edilen bulguların kullanılabilirlik ve güvenilirliğini değerlendirilmektedir (Palacios, Pantoja, Navarro, Puetaman ve Toledo, 2016).

Diğer model ise 6 aşamadan oluşan CRISP-DM *(Cross Industry Standard Process for Data Mining)* modelidir. Sürecin ilk aşaması işin anlaşılması sürecidir. Bu aşamada projenin amaçlarının ve gerekliliklerinin anlaşılması ve bu bilginin bir VM problem tanımına dönüştürerek ilgili hedeflerin gerçekleştirilmesi amacıyla planı oluşturmaktır. İkinci aşama verinin anlaşılmasında veri toplama, veri kalitesini bozan

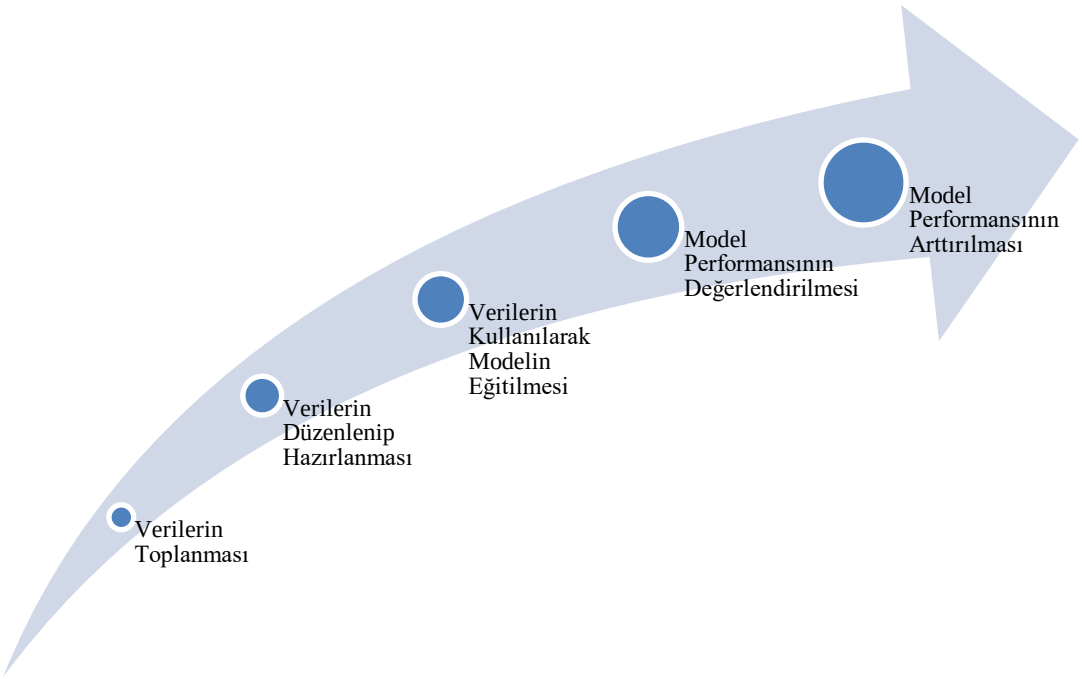
sorunlara işaret etmek ve veriyle ilgili ilk izlenimleri ortaya çıkarmaktır. Veri hazırlama sürecinde ilk olarak elde edilen ham veriden nihai veri setinin oluşturulmasıdır. Modelleme sürecinde çeşitli modelleme teknikleri seçilerek veri setine uygulanır ve parametreler optimum değerlerine dönüştürülür. Değerleme aşamasında kullanılan modellerin ve modelin kurulması aşamasında izlenen adımların detaylı bir şekilde değerlendirilmesi ve sonuçların amaçlara yönelik olup olmadığının izlenmesidir. Son aşama olan sonucun kullanıcılarla paylaşılmasında süreç sonunda elde edilen bilginin son kullanıcıya anlayabileceği şekilde sunulmasıdır. Şekil 1'de CRISP-DM modelinin aşamalarına yer verilmiştir.



Şekil 1 : CRISP-DM modeli.

Makine öğrenmesi en genel tanımıyla geçmiş deneyimlerden yola çıkılarak elde edilen öğrenmelerden bir model oluşturulup geleceğe yönelik tahminlerde bulunmaktır. Bilgisayarlar elde edilen verileri kullanarak bir model oluşturur ve bu modeli yeni verilerde kullanarak karar almaktadır (Öğücü, 2006).

Makine öğrenmesi modellerinin oluşturulması için izlenmesi gereken beş adım Şekil 2'de bulunmaktadır.

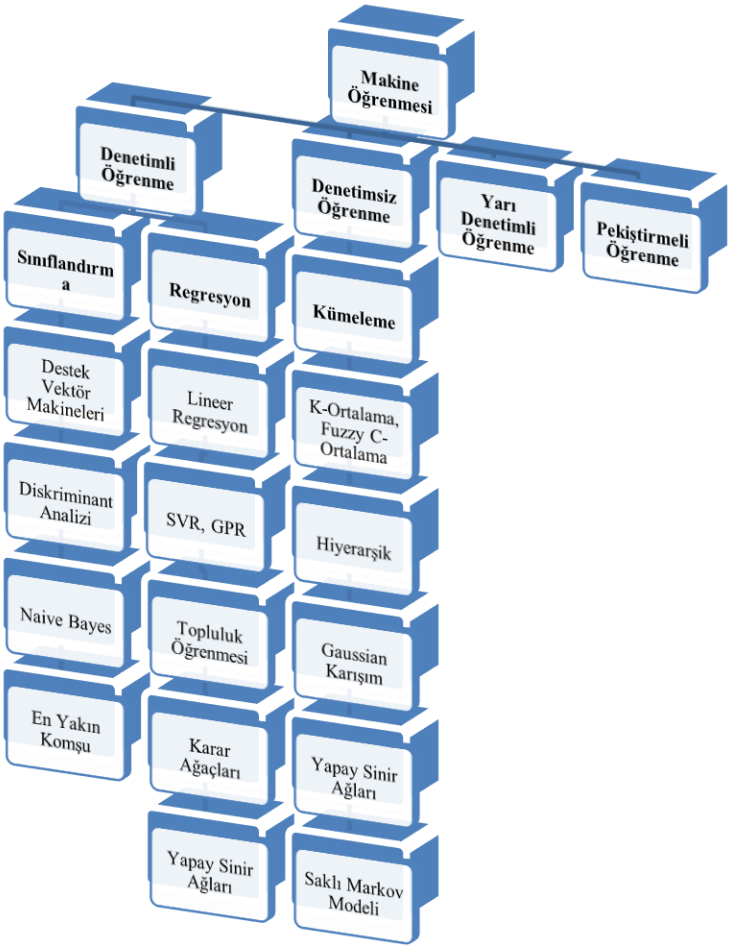


Şekil 2 : Makine öğrenmesi adımları.

Büyük verinin son zamanlarda popüler hale gelmesiyle makine öğrenmesi algoritmik işlemler, müşteri kaybı hesaplama gibi bilgisayar temelli finans; tümör tanımlama gibi bilgisayarlı biyoloji, imaj işleme, doğal dil işleme gibi alanlardaki sorunların çözümlenmesinde kullanılmaktadır.

Makine öğrenmesinde genel olarak 3 teknik bulunmaktadır. Bu makine öğrenme yöntemleri temel olarak programcı tarafından makine için tasarlanan algoritmaların nasıl eğitildiği ve hangi algoritmaların kullanıldığına göre farklılık gösterir. Denetimli öğrenme, modeli girdi ve çıktı verilerle eğiterek gelecek verileri tahmin etmesinde kullanılmaktadır. Bu tür makine öğrenme yöntemi, başlığından da anlaşılacağı gibi, programcı veya algoritma tarafından yönlendirilen bir yöntemdir. Bu öğrenme yöntemi, verilen girdilerle ilgili spesifik çıktı türlerini veren etiketli verileri içerir. Yani algoritma çıktıları tahmin etmek zorunda değildir, çünkü çıktılar verisetinde modelin öğrenmesi için kullanılmıştır. Denetimsiz öğrenmede ise; girdi verileri kullanılarak beklenen çıktı verileri bulunmamaktadır ve girdilerden sonuçlar çıkartılmaya çalışılmaktadır. Bu öğrenme yönteminde, algoritma etiketlenmemiş verileri içerir. Bu

tür bir algoritmadaki veriler rehbersizdir. Programcı, makineler için öğrenme kaynağı olan makinedeki verileri besler. Makine, verilerdeki gizli kalıpları ve eğilimleri tespit eder. (Shan, 2016). Denetimsiz öğrenme, bağdaştırma ve kümeleme olmak üzere iki alt başlığa ayrılmaktadır. Bağdaştırma elementlerin diğer elemente bağımlılıklarına dayanırken, kümeleme aynı kategoride sınıflandırmak amacıyla iki benzer elementi kümelemektedir. (Kashyap, 2017). Şekil 3'te yaygın olarak kullanılan makine öğrenmesi algoritmaları yer almaktadır.



Şekil 3 : Yaygın kullanılan ML algoritmaları.

6.1.1 Özellik seçimi

Özellik seçimi, makine öğrenme algoritması tasarımında yer alan ikinci adımdır. İşe yarayacak verilere asıl odaklanma bu süreçte gerçekleşmektedir. (Gopika, 2018). Modeli beslemek amacıyla kullanılacak ham veride işe yarayan veriler olacağı gibi modele herhangi bir katkı sağlamayacak veriler de yer alabilir. Bu süreçte girdi verisi ile ham veri arasında bir tür filtre uygulamak gerekmektedir. Makine öğrenmesi

alanında kullanılan bu filtreye özellik denir. Her bir model farklı amaçlara yönelik kurulduğu için kullanılan özelliklerin de ilgili amaca hizmet etmesi gerekmektedir. Özellik seçiminin amacı makine öğrenme sürecinin etkinliğini arttırmaktır (Raza ve Qamar, 2017). Özellik seçiminde, algoritma verimliliğini artırmak için binlerce özellik setinden en uygun özellikleri bulunur ve eğitim süreci bu özelliklere odaklanır.

Doğru özellik seçiminin bazı avantajları bulunmaktadır (Miche, Bas, Lendasse, Jutten& Simula, n.d.).

1. Aşırı uyumu azaltılmaktadır. Giriş verisi miktarının azaltılması gereksiz verileri ve dolayısıyla veri işlemedeki gürültüyü azaltır.

2. Doğruluk oranı artmaktadır. Modele herhangi bir katkı sağlamayacak verilerim model çıkarılması çıktı sonuçlarının doğruluğunu etkileyen veri miktarının azaltılmasını sağlamaktadır.

3. Modelin eğitim süresi azalmaktadır. Algoritma daha az veriyle beslendiğinde, daha az karmaşık hale gelir ve eğitim süresi azalır.

6.1.2 Özellik çıkarımı

Bu teknik özellik seçimi sürecinden benzer kombinasyonlara sahip özelliklerden yeni özellikler çıkarılması bakımından farklılık göstermektedir. Bu teknik sayesinde verisetindeki özellik sayısı azalmakta ve eski özelliklerin kombinasyonlarından oluşan yeni verisetleri oluşmaktadır. (Guyon, Gunn& Zadeh, 2006).

Özellik çıkarımı sürecinde sıklıkla kullanılan yöntemlerden birisi temel bileşenler analizidir. Temel bileşenler analizine göre bir boyut azaltımı yöntemi olup asıl amacı büyük veri kümesinin önemli bir miktarda bilgi kaybı olmaksızın küçük bir veri kümesine dönüştürme yöntemidir (Howley, Madden, O' Connell& Ryder, n.d.).

Makine öğrenmesi sürecinde özellik sayısının azaltılması ile doğruluk arasında ters bir orantı olduğu varsayılsa da boyut azaltımı sayesinde modelde basitlik elde edilmektedir. Küçük verisetleri çalışması ve görselleştirmesi hızlı olmakla birlikte makine öğrenmesi algoritmalarının hesaplama süreleri de daha çabuktur.

Temel bileşenler analizinde ilk süreç değişkenlerin standardizasyonudur. İlk girdiler standartlaştırılmadan farklı aralıklarda bulunan özelliklerin dağılımı modelin yanlış eğitilmesine neden olmaktadır (Keho, 2012). Modelde kullanılan girdi ölçeklerinin birbirleriyle benzer olması gerekmektedir. Benzer olmadığında daha geniş ölçekli

girdiler diğerlerine baskın olacak ve modelde yanlı sonuçların elde edilecektir. Temel bileşenler analizinin ikinci adımında ise kovaryans matrisi hesaplanmaktadır. Kovasyan matrisi ilgili modeldeki girdilerin ortalamadan hangi ölçüde farklılık gösterdiğini hesaplamaktadır. Üçüncü adımda ise kovaryans matrisinin özvektör ve özdeğerleri hesaplanmaktadır.

Özellik Eliminasyonu boyut azaltma tekniklerinden biri olup modelde kullanılacak olan girdi sayısını azaltmak için kullanılmaktadır. Verideki gürültüyü azaltmak amacıyla verinin en uygun kombinasyonu kullanılır ve böylece en etkin girdi kombinasyonu seçilmektedir. Belli başlı 3 adet rekürsif özellik eliminasyonu bulunmaktadır (Chen ve Jeong, 2007).

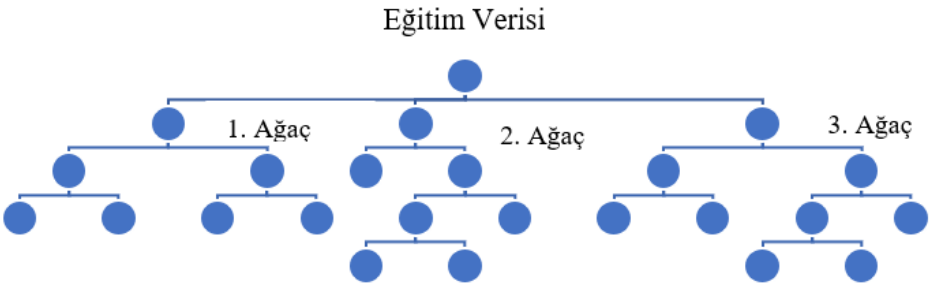
- İleriye doğru seçilim: bu süreçte başlangıçta özellik sayısı 0 olarak başmaktadır. Her bir iterasyonda yeni bir özellik modele eklenmektedir. Her bir özellik eklendikçe model iyileşecek ve belirli bir süre sonunda eklenen yeni özellik modele herhangi bir katkı sağlamayacaktır. Modelin doyma noktasına geldiği bu süreçte iterasyon bitmektedir (Khan, Aelst& Zamar, 2007).
- Geriye doğru eliminasyon: ileriye doğru seçilim modelinin tam tersi şeklinde çalışmaktadır. Öncelikle modele tüm özellikler eklenmekte ve her bir iterasyonda teker teker özellikler çıkarılmaktadır. Çıkarılan en son özelliğin modele katkı sağlamadığı noktada iterasyon sona ermektedir (Samal, Mohanty& Fifarek, 2007).
- Rekürsif özellik eliminasyonu: bu teknik verilen girdi setinden en iyi özellik seti oluşturmak amacıyla kullanılmaktadır. Her bir iterasyonda farklı bir özellik seti kullanılmaktadır ve bu özellik setlerinden yeni bir model oluşturulmaktadır. Modeller en iyi performans gösteren özelliklere göre sıralanmaktadır. Algoritma modele katkı sağlamayan özellikleri modelden çıkarmaktadır. Böylelikle en yüksek sıralamaya sahip özelliklerle model kurulmaktadır.

6.2 Rassal Orman Algoritması

Rassal orman algoritması topluluk öğrenmesine dayalı denetimli öğrenme yöntemlerinden biridir. Topluluk öğrenmesi farklı ya da aynı tür algoritmaları daha güçlü bir model oluşturmak amacıyla birçok defa birlikte kullanmaya dayalı bir

yöntemdir. Rassal orman algoritması aynı türden olan çoklu algoritmaları yani çoklu karar ağaçlarını bir araya getirerek karar ağacı ormanı oluşturmaktadır. Ilgili algoritma hem regresyon hem de sınıflandırma problemlerinin çözümü için kullanılmaktadır.

Karar ağaçları her bir dalı düğüm olarak nitelendirilen hiyerarşik ağaç benzeri yapılardır. Ancak karar ağaçları aşırı uyum sağlama (over fitting) eğilimindedir. Aşırı uyum belirli bir sonuca erişebilmek amacıyla ağaca daha fazla sayıda düğüm ekleyerek ağacın derinliğini arttırıp karmaşık hale getirmektedir. Rassal orman çok sayıdaki karar ağacından oluşan ve karar ağaçlarındaki aşırı uyum sorununu gidermek amacıyla kullanılan bir yöntemdir. Karar ağaçları belirli verisetinden Rassal özellikler seçilerek Rassal oluşturulmaktadır. Rassal orman algoritması karar verirken ya da tahminde bulunurken karar ağaçlarından edinilen en çok sayıdaki oyu temel almaktadır. Rassal orman oluşturulurken özellikler bagging yöntemi ile Rassal seçilmektedir. Verisetindeki özelliklerden alt eğitim kümeleri Rassal özelliklerin yer değiştirilmesiyle seçilmektedir. Bir özellik farklı alt eğitim kümelerinde aynı anda tekrarlanabilmektedir. Örneğin veriseti 20 özellik içeriyorsa ve farklı karar ağaçlarının oluşturulması için 5 özellik seçilmesi gerekiyorsa bu 5 özellik rastsal olarak seçilmekte ve herhangi bir özellik bir veri altsetinden daha fazla sette yer alabilir. Bu sayede rastsallık elde edilmekte ve aşırı uyum problem ağaçlar arasındaki korelasyonun azalmasıyla ortadan kaldırılmaktadır. Özellikler seçildiğinde ağaçlar en iyi ayrım noktasında göre oluşturulur. Her ağaç oy olarak bahsedilen bir çıktı verir. Rassal orman tarafından en fazla sayıdaki oyu alan sonuç seçilmektedir, sürekli değişkenlerin olması durumunda tüm çıktıların ortalaması nihai çıktı olarak sayılmaktadır. Aşağıdaki şekilde eğitim kümesi girdi verisinin örneklemleri kullanılarak bootstrap yöntemi ile eğitilmektedir. Test kümesinde sonuç test datasının her bir ağaçtan gelen tahminlerin ortalaması alınarak hesaplanmaktadır. Şekil 4'te Rassal Orman algoritmasında karar alma mekanizmasına yer verilmiştir.



Şekil 4 : Rassal Orman Teorisi (Mennitt, Sherrill& Fristrup, 2014).

Karar ağacı algoritması kullanırken takip edilecek temel adımlar aşağıdadır. (Sandri ve Zuccolotto, 2006).

1. Verisetinden N rassal kayıt seçilmektedir.
2. N kayıda dayalı karar ağacı oluşturulmaktadır.
3. Algoritmada yer alacak ağaç sayısına karar verilerek 1. 2. Adımlar tekrarlanmaktadır.
4. Regresyon problem durumunda, her bir yeni kayıt için ormandaki her ağaç Y (çıktı) için bir değer tahmin eder. Son değer ormandaki tüm ağaçların tahmin ettikleri değerlerin ortalamasıdır. Sınıflandırma problem durumunda ise her bir yeni kaydın dahil olduğu kategori algoritma tarafından tahmin edilmektedir. Son olarak yeni kayıt en çok oy alan kategoriye eklenmektedir.

Her algoritmada olduğu gibi Rassal orman algoritmasının da bazı avantaj ve dezavantajları bulunmaktadır. Avantajlarına göre; (Tsipouras, Tsouros, Smyrlis, Giannakeas& Tzallas, 2018).

1. Rassal orman algoritması önyargılı bir algoritma değildir çünkü çoklu ağaçlar bulunmakta ve her ağaç belirli bir veri altkümesi kullanılarak eğitilmektedir.
2. Algoritma sabittir. Verisetine yeni bir veri noktası eklemek bile algoritmanın genel sonucu etkilemeyebilir. Çünkü yeni veri sadece tek bir ağacı etkileyebilir ve ormandaki tüm ağaçları etkilemesi oldukça zordur.
3. Ilgili algoritma hem kategorik hem de sayısal özelliklerde iyi çalışmaktadır.
4. Rassal orman algoritması veride kayıp değerler bulunması durumunda ya da veri doğru ölçeklendirilmemiş olması durumunda da iyi çalışmaktadır.

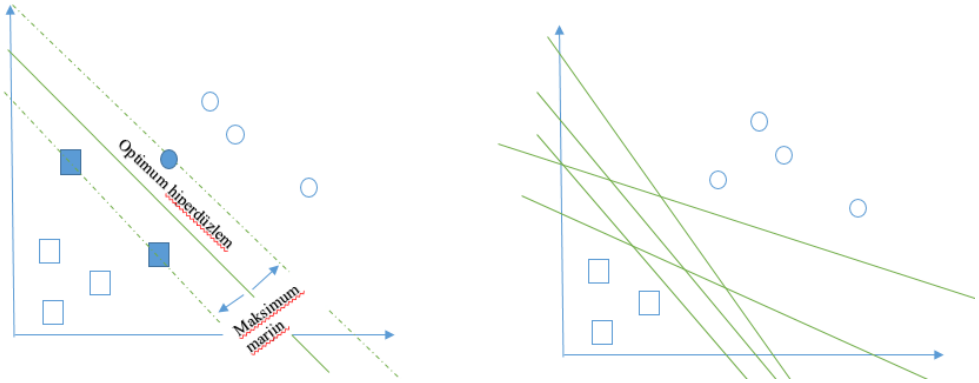
Dezavantajlarına göre; (Tsipouras vd., 2018).

1. Rassal orman teorisinin en önemli dezavantajı karmaşık yapıdır. Çok sayıdaki karar ağaçlarının bir araya getirilmesi daha fazla hesaplama kaynağı gerektirmektedir
2. Diğer algoritmalara kıyasla karmaşık yapısı sayesinde modelin eğitim süreci daha uzun sürmektedir.

6.3 Destek Vektör Makineleri

Destek Vektör makineleri Vapnik tarafından ortaya atılan bir yöntemdir (Vapnik, 1995). DVM temelinde bir regresyon modelidir. Denetimli öğrenme yöntemi olan DVM genellikle regresyon ve sınıflandırma analizlerinde kullanılmaktadır. DVM aynı zamanda olasılıksal ikili lineer sınıflandırıcı olarak da bilinmektedir; ancak farklı çekirdek (kernel) tiplerinin kullanımı ile lineer olmayan sınıflandırma sorunları için de kullanılmaktadır. DVM metin sınıflandırma, imaj sınıflandırma, permütasyon testleri, hava ve satış tahminleri gibi birçok gerçek dünya problemlerinin çözümü için kullanılmaktadır. DVM maksimum marjin sınıflandırıcılarındandır. Sınıfları ayırmak için maksimum marjin hiper düzlemi oluştururlar. DVM'nin amacı N özellik sayısı olmak üzere N boyutlu bir uzayda veri noktalarını sınıflandıran hiper düzlemi oluşturmaktadır.

İki sınıflı veri noktalarını ayırabilmek için birçok farklı hiper düzlem seçilebilir. DVM'nin amacı maksimum marjine sahip hiper düzlemi oluşturmaktır. Maksimum marjin iki sınıfa ait veri noktaları arasında maksimum uzaklığı sağlayan uzaklıktır. Marjin uzaklığını maksimize etmek ileride modele eklenecek veri noktalarının daha doğru bir şekilde sınıflandırılmasını sağladığı için modeli güçlendirmektedir. Destek vektörleri hiper düzleme yakın olan veri noktaları olup hiper düzlemin pozisyon ve yönelimini etkilemektedir. Bu destek vektörleri kullanılarak sınıflandırıcının marjini maksimize edilmektedir (Vapnik ve Cortes,1995). Şekil 5'te lineer DVM ile ilgili hiperdüzlem ve marjin genişliği gösterilmiştir.



Şekil 5 : Destek Vektör Makineleri.

6.4 Veri Seti

Bu çalışmada Dow Jones Endüstri Ortalaması ve Nasdaq 100 kapanış fiyatları kullanılarak endeks hareketinin hareketi tahmin edilmiştir. Tahminlerde DVM ve Rassal Orman modelleri kullanılmıştır. çalışmanın amacı önce teknik göstergeleri girdi olarak kullanarak DVM ve Rassal Orman algoritmalarının tahmin başarısını ölçmek ve sonrasında modele ekonomik girdileri dahil ederek doğruluk skorunun nasıl değiştiğini gözlemlemektir.

Endeks açılış, kapanış, en yüksek, en düşük, hacim ve teknik göstergeler finansal data sağlayan bir API üzerinden sağlanmıştır. Program yazılırken Python 2.7 versiyonu kullanılmıştır. Modellerin kurulmasında Python scikit-learn kütüphanesi kullanılmıştır.

Çalışmada modelleri eğitmek amacıyla diğer çalışmalarda da sıklıkla kullanılan aşağıdaki teknik göstergeler ve ekonomik değişkenler girdi olarak seçilmiştir (“Tradingview”, t.y.).

Basit hareketli ortalama: hesaplanan süre aralığındaki fiyatların toplanarak ilgili süre aralığındaki periyot sayısına bölünmesi ile hesaplanmaktadır. Örneğin 5 günlük hareketli ortalama hesaplanmak istendiğinde son 5 günün kapanış fiyatı 5’e bölünmektedir.

Ağırlıklı hareketli ortalama: ağırlıklı hareketli ortalama hesaplanan süre içerisindeki son günlerdeki fiyatlara ağırlık verilerek hesaplanmaktadır.

Üssel hareketli ortalama: Üssel hareketli ortalama ağırlıklı hareketli ortalama çeşidi olup basit hareketli ortalamaya kıyasla güncel fiyat verilerine daha fazla ağırlık

vermektedir. Üssel hareketli ortalama son zamanlardaki fiyat değişikliklerine basit hareketli ortalamaya kıyasla daha hızlı tepki vermektedir.

Hareketli ortalama yakınsama/uzaklaşma: trend takip eden momentum indikatörü olan bu gösterge ilgili kıymetin iki farklı hareketli ortalamaları ile hesaplanmaktadır. İlgili kıymetin 26 periyot üssel hareketli ortalamasından 12 periyot üssel hareketli ortalamasının farkı alınarak hesaplanmaktadır.

Momentum göstergesi: momentum göstergesi piyasadaki fiyat değişimlerinin gücünü gösteren bir göstergedir. Hesaplanırken son fiyat ile geçmişteki fiyat karşılaştırılarak farkı alınmaktadır. 5 günlük bir momentum hesaplandığında 5 gün önceki fiyat>son fiyat ise momentum negatiftir.

Stokastik Osilatör: stokastik osilatör bir kıymetin geçmiş fiyatları ile bugünkü fiyatları arasında bir ilişki kurmak için hesaplanan göstergedir. %K ve %D değerleri ile hesaplanan göstergede %K, son 14 günlük en yüksek ve en düşük kapanış değeri ile son kapanış değeri arasındaki fark kullanılarak hesaplanırken %D değeri %K değerinin 3 günlük hareketli ortalamasıdır.

Göreceli güç endeksi: hesaplanan periyottaki kapanış değerlerinin bir önceki periyottaki kapanış değerleri ile kıyaslanarak hesaplanan bir göstergedir. Kısa-orta vadeli eğilimin yönü hakkında bilgi vermektedir.

Ortalama yönsel endeks: teknik analizde sıkça kullanılan bu gösterge trendin gücünü ölçmek amacıyla kullanılmaktadır. Trend negatif ya da pozitif olabilmektedir.

Larry William's R: piyasada aşırı alım ya da aşırı satım olup olmadığını ölçen bir göstergedir. İndikatördeki 80-100 aralığındaki değişim hissede aşırı satım; 0-20 arasındaki değişim ise aşırı alım olduğunu göstermektedir.

Birikim/Dağılım İndikatörü: Kümülatif bir indikatör olan birikim dağılım hacim ve fiyatı kullanarak ilgili hissenin biriktirildiğini mi dağıtıldığını mı ölçen bir göstergedir.

Emtia kanal endeksi: hissenin kapanış fiyatı ile ilgili periyot önceki ağırlıklı ortalamaları karşılaştırmaktadır. Hisse fiyatının kendi istatistiksel ortalamasından ne kadar farklılaştığını ölçmektedir.

Aaron göstergesi: "AroonUp" ve "AroonDown" şeklinde iki gösterge kullanılarak hesaplanır. AroonUp belirlenen periyotta en yüksek fiyat görüldükten sonra geçen

periyot sayısı; AroonDown ise belirlenen periyotta en düşük fiyat görüldükten sonra geçen periyot sayısını ifade etmektedir.

Bollinger Bantları: fiyatın ortalamadan ne kadar uzaklaştığını ölçmek için kullanılmaktadır. Orta Bollinger çizgisi 20 günlük hareketli ortalamadan hesaplanır ve üst Bollinger çizgisi orta Bollinger çizgisine 2 standart sapma eklenerek; alt Bollinger çizgisi ise 2 standart sapma çıkarılarak hesaplanmaktadır.

Denge İşlem Hacmi: fiyatlardaki değişimle hacmi beraber değerlendirerek oluşturulan bir göstergedir. İlgili kıymetin son kapanışı bir öncekinden büyük ise tüm işlem hacminin bu kıymeti alanlara ait olduğu varsayılmakta; tersi durumda ise bu kıymeti satanlara ait olduğu varsayılmaktadır.

Ekonomik değişkenler olarak Brent oil futures, Gold futures, DAX, Nikkei 225, S&P 500, Shangai Composite, USD/EUR-GBP-JPY verileri kullanılmıştır. İlgili verilerin 2012-2018 yılları arasında açılış, kapanış, en yüksek, en düşük, fiyat değişim değerleri alınmış ve modelde fiyat değişiklik verileri kullanılmıştır. Modelde kullanılan teknik göstergelerin açıklamalarına aşağıda yer verilmiştir. Değişkenlerin doğru kombinasyon ile seçilmesi ve modele katkı sağlayan özelliklerin seçilmesi amacıyla rekürsif özellik eliminasyonu yöntemi kullanılmıştır.

Her bir girdiye ait veri setinde null değer olup olmadığı kontrolü yapılmış ve hiçbir güne ait null değer bulunmamıştır. Verilerin istatistiksel özetleri yapılmıştır. Modelde sadece değişim verileri kullanılacağı için her bir özelliğe ait "change" değeri ilgili değişkenin adıyla adlandırılmış, böylece tüm değerler birleştirilirken verinin hangi değişkene ait olduğu belirtilmiştir. Kullanılan değişkenlere ait istatistiksel özet tabloya Çizelge 2'de yer verilmiştir.

Çizelge 2 : Ekonomik girdilere ait özet tablolar.

```
brent_oil.describe()
```

	Price	Open	High	Low
count	1548.000000	1548.000000	1548.000000	1548.000000
mean	78.919231	78.931654	79.784367	78.013714
std	29.378784	29.342472	29.370363	29.297804
min	27.880000	27.990000	28.750000	27.100000
25%	50.800000	50.882500	51.547500	50.060000
50%	66.230000	66.075000	66.905000	65.050000
75%	108.767500	108.682500	109.460000	107.792500
max	126.220000	126.160000	128.400000	125.000000

```
USD_EUR .describe()
```

	Price	Open	High	Low
count	1566.000000	1566.000000	1566.000000	1566.000000
mean	0.829482	0.829388	0.832929	0.826056
std	0.073472	0.073471	0.074230	0.072684
min	0.717700	0.717700	0.718900	0.714500
25%	0.760300	0.760200	0.763525	0.756925
50%	0.827750	0.826300	0.830150	0.824700
75%	0.896375	0.896300	0.900000	0.892100
max	0.962700	0.962700	0.967100	0.960000

```
USD_GBP .describe()
```

	Price	Open	High	Low
count	1566.000000	1566.000000	1566.000000	1566.000000
mean	0.674902	0.674777	0.677634	0.672232
std	0.066260	0.066275	0.067085	0.065594
min	0.582600	0.582500	0.583500	0.581600
25%	0.626025	0.625950	0.628400	0.624025
50%	0.650000	0.649900	0.652350	0.647350
75%	0.740125	0.739800	0.743475	0.735775
max	0.830000	0.829600	0.846100	0.827700

```
USD_JPY .describe()
```

	Price	Open	High	Low
count	1566.000000	1566.000000	1566.000000	1566.000000
mean	104.228889	104.191194	104.648091	103.726015
std	13.553118	13.564863	13.623529	13.492504
min	76.190000	76.180000	76.240000	76.020000
25%	98.237500	98.155000	98.705000	97.742500
50%	104.850000	104.835000	105.350000	104.300000
75%	113.897500	113.912500	114.355000	113.325000
max	125.620000	125.580000	125.860000	124.520000

```
DAX.describe()
```

	Date	Price	Open	High	Low	Vol.	Change %
count	1520	1520	1520	1520	1520	1520	1520
unique	1520	1517	1519	1516	1518	1406	471
top	Jun 11, 2012	9,339.17	11,881.49	11,881.81	9,805.55	82.31M	0.19%
freq	1	2	2	2	2	4	17

```
gold_features.describe()
```

	Date	Price	Open	High	Low	Vol.	Change %
count	1564	1564	1564	1564	1564	1564	1564
unique	1564	1280	1283	1283	1314	223	447
top	Jun 11, 2012	1,273.70	1,252.70	1,254.50	1,218.60	0.01K	0.01%
freq	1	5	5	4	5	127	16

```
Nikkei.describe()
```

	Date	Price	Open	High	Low	Vol.	Change %
count	1502	1502	1502	1502	1502	1502	1502
unique	1502	1502	1500	1502	1500	1118	491
top	Jun 11, 2012	14,326.46	19,953.06	14,839.79	17,869.74	-	-0.39%
freq	1	1	2	1	2	29	12

```
SP.describe()
```

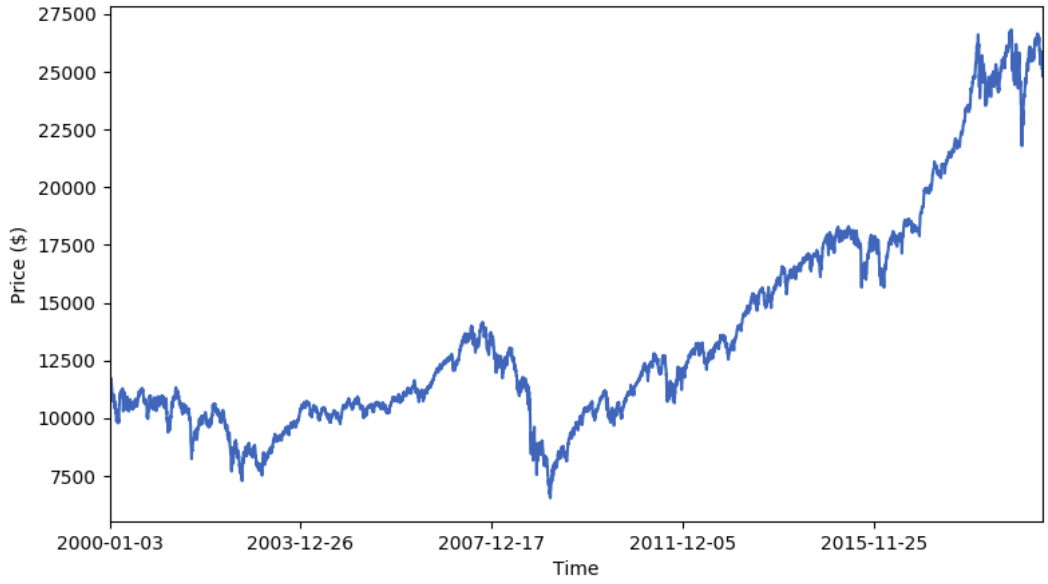
	Date	Price	Open	High	Low	Vol.	Change %
count	1509	1509	1509	1509	1509	1509	1509
unique	1509	1500	1497	1494	1501	1	349
top	Jun 11, 2012	2,268.90	1,413.09	2,578.29	2,041.88	-	-0.01%
freq	1	2	2	2	3	1509	18

```
Shanghai .describe()
```

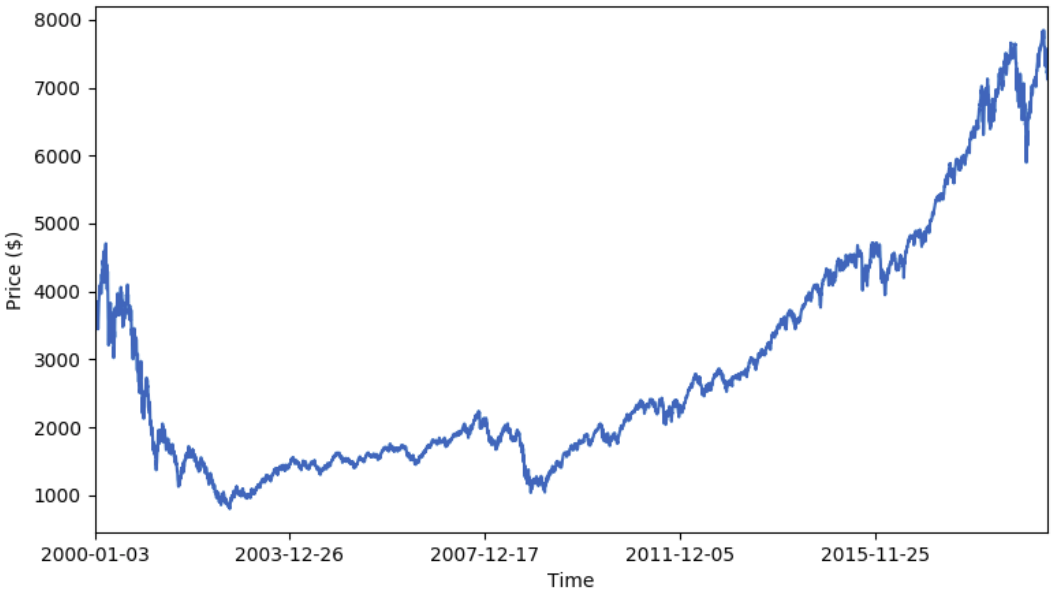
	Date	Price	Open	High	Low	Vol.	Change %
count	1459	1459	1459	1459	1459	1459	1459
unique	1459	1453	1448	1450	1452	1160	476
top	Jun 11, 2012	2,073.10	2,023.15	3,219.52	2,132.63	7.16B	0.07%
freq	1	2	2	2	2	6	15

Modelde kullanılan teknik göstergeler ve DJIA ile NDX endekslerine ilişkin bilgiler Alpha Vantage üzerinden sağlanmıştır. API anahtarı ile erişim sağlanan Vantage üzerinden sorgu kısıtlaması nedeniyle modelde 65 saniyelik sleep zamanı uygulanmıştır. Sınıflandırmak için bir sonraki günün kapanış fiyatının yönü kullanılmıştır. Fiyatın aşağı yönlü hareketi olması durumunda 0; tersi hareketinde ise 1 kullanılmıştır. Fiyatların yukarı yönlü hareketinin aşağı yönlü hareketinden daha fazla olması sebebiyle azınlık ve çoğunluk sınıflarındaki verilerin dengeli hale gelmesine yönelik azınlık sınıfı upsample edilerek çoğunluk sınıfı ile yeniden birleştirilmiştir. Veri normalizasyonu amacıyla Python standard scaler kütüphanesi kullanılarak tüm verilerin dağılımı standart sapması 1 ortalaması 0 olacak şekilde normal dağılıma uygun hale getirilmiştir. API üzerinden alınan DJIA ve NDX kapanış verilerini gösteren grafikler matplotlib kütüphanesi kullanılarak grafik haline

getirilmiştir. Şekil 6-7 DJIA ve NDX endekslerinin fiyat-zaman grafiğini belirtmektedir.



Şekil 6 : DJIA fiyat-zaman grafiği.



Şekil 7 : NDX fiyat-zaman grafiği.

6.5 Model Sonuçları

Sadece teknik göstergelerin girdi olarak kullanıldığı durumda modelde aşağıdaki sonuçlar elde edilmiştir.

DJIA için herhangi bir parametre optimizasyonu yapmadan test verisi üzerinde aşağıdaki sonuçlar elde edilmiştir. Buna göre Çizelge 3'te de görüleceği gibi Rassal orman algoriması daha yüksek bir doğruluk değeri sağlamıştır.

Çizelge 3 : DJIA test verisi.

	Doğruluk	**F1 Skoru**	**ROC Eğrisi**
DVM	56.17 %	0.5799	0.5612
Rassal Orman	66.67%	0.6707	0.6685

Parametre optimizasyonu yapılmasının ardından Çizelge 4'te de görüleceği üzere DVM modeli daha yüksek bir doğruluk elde etmiştir.

Çizelge 4 : DJIA grid araması verisi.

	Doğruluk	**F1 Skoru**	**ROC Eğrisi**
DVM	66.67 %	0.6747	0.6677

Modelin doğruluğunu teyit etmek amacıyla validasyon verisi üzerinden DJIA için aşağıdaki sonuçlar elde edilmiştir. Çizelge 5 validasyon verisinin sınıflandırma sonuçlarını göstermektedir.

Çizelge 5 : DJIA validasyon verisi.

	Doğruluk	**F1 Skoru**	**ROC Eğrisi**
DVM	51.69 %	0.5649	0.5121

NDX test verisi ile parametrelerde değişiklik yapmadan Rassal orman algoritması DVM'den daha yüksek bir doğruluk payı vermiştir. Çizelge 6'da NDX test verisinin DVM sınıflandırma sonucu yer almaktadır.

Çizelge 6 : NDX test verisi.

	Doğruluk	**F1 Skoru**	**ROC Eğrisi**
DVM	57.83 %	0.5205	0.5738
Rassal Orman	68.67 %	0.6533	0.6837

Parametre optimizasyonunun ardından DVM modeli daha yüksek bir doğruluk payı elde etmiştir. Çizelge 7'de grid search verisi yer almaktadır.

Çizelge 7 : NDX grid araması verisi.

	Doğruluk	**F1 Skoru**	**ROC Eğrisi**
DVM	63.86 %	0.5833	0.6336

Modelin doğruluğunu test etmek amacıyla validasyon verisinde de NDX için DVM modeli daha yüksek doğruluk skoru vermiştir. Çizelge 8'de de validasyon verisinin DVM algoritması kullanılarak yapılan sınıflandırma sonucu yer almaktadır.

Çizelge 8 : NDX validasyon verisi.

	Doğruluk	**F1 Skoru**	**ROC Eğrisi**
DVM	51.13 %	0.5411	0.5100

Sadece teknik göstergelerin girdi olarak kullanıldığı endeks yönünün tahmininde DVM algoritması rassal orman algoritmasından daha yüksek doğruluk elde etmiştir.

Modele ekonomik göstergeler dahil edildiğinde rekürsif özellik eliminasyonu algoritması kullanılarak modele katkı sağlamayan özellikler model dışı tutulmuştur. DJIA için DVM modeli kullanıldığında tüm özellikler aşağıdaki şekildeki gibi sıralanmıştır. DVM algoritmasının halihazırda doğruluk skoru düşük olması sebebiyle herhangi bir özelliğin eliminasyonu doğruluğu büyük ölçüde azalttığı için tüm özellikler modelde girdi olarak kullanılmıştır. Çizelge 9'da RFE algoritma sonuçları yer almıştır.

Çizelge 9 : RFE algoritma sonucu DJIA- DVM.

Nr.	Özellik	Sıra
0	SMA	1
1	EMA	1
2	MACD_Signal	1
3	MACD_Hist	1
4	MACD	1
5	RSI	4
6	ADX	2
7	CCI	3
8	Aroon Up	1
9	Aroon Down	4
10	Real Middle Band	1
11	Real Upper Band	1
12	Real Lower Band	1
13	Chaikin A/D	1
14	OBV	4
15	WMA	1
16	MOM	1
17	WILLR	2
18	SlowD	1
19	SlowK	1
20	4. close	1
21	Change_oil	3
22	Change_DAX	3
23	Change_gold	4
24	Change_Nikkei	3
25	Change_SP	2
26	Change_Shanghai	4
27	Change_USD_EUR	2
28	Change_USD_GBP	3
29	Change_USD_JPY	2

Modele ekonomik göstergeler dahil edildiğinde DJIA endeksi için DVM algoritmasının doğruluk skoru %72'ye yükselmiştir. DJIA'ya ait sınıflandırma

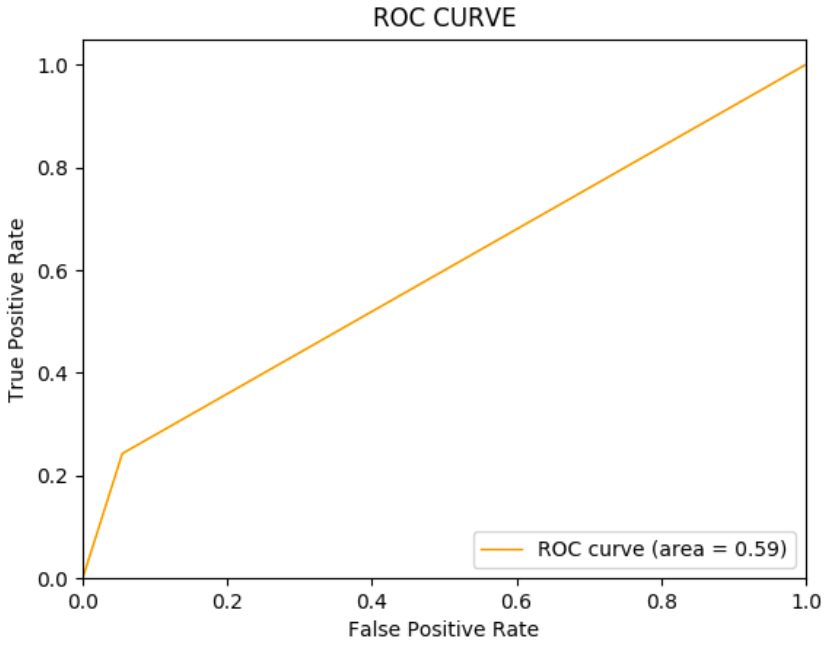
raporuna aşağıdaki şekilde yer verilmiştir. Rapora göre 0 test verisi kullanılarak yapılan sınıflandırma ölçütleri; 1 ise tahmin edilen değerler üzerinden hesaplanan skorlardır. İlgili çizelgeyi yorumlamak için duyarlılık (precision), kesinlik (recall) ve f-skoru (f1-score) üzerinde durulması gerekmektedir. Modelin tahmin başarısını ölçmek için kullanılan bu değerler karışım matrisinde bulunan True Positive (Gerçek Pozitif), True Negative (Gerçek Negatif), False Negative (Yanlış Red) ve False Positive (Yanlış Kabul) sonuçlarından hesaplanmaktadır. TP ve TN modelin tahminin gerçekle uyuştuğu durumdur. FN model tarafından negatif tahmin edilip; gerçekte pozitif olan değer sayısını; FP ise model tarafından pozitif tahmin edilip gerçekte negatif olan sayıyı ifade etmektedir. Buna göre duyarlılık gerçekte pozitif değer sayısının doğru olarak pozitif tahmin edilen örnek sayısı ve negatif olarak tahmin edilen pozitif örnek sayısına oranıdır. Kesinlik ise gerçekte pozitif değer sayısının doğru olarak pozitif tahmin edilen örnek sayısı ve yanlışlıkla pozitif olarak değerlendirilen örnek sayısına oranıdır. Duyarlılık ve kesinlik birbirleriyle ters orantılıdır. F skoru duyarlılık ve kesinlik ölçütlerinin harmonik ortalamasıdır. Şekil 8'de DJIA için yapılan sınıflandırma sonucu yer almaktadır.

	Duyarlılık	**Kesinlik**	**F1 Skoru**
0	0.72	0.95	0.82
1	0.68	0.24	0.36
Doğruluk			0.72
Makro Ort.	0.70	0.59	0.59
Ağırlıklı Ort.	0.71	0.72	0.67

Şekil 8: DJIA – DVM sınıflandırma raporu.

DJIA için DVM modeli kullanılarak yapılan sınıflandırmanın başarısı ROC eğrisi kullanılarak grafik haline getirilmiştir. ROC eğrisi sınıflandırma problemlerinin başarısını ölçmek için kullanılan bir yöntemdir. Modelin sınıfları ayırt etmede ne kadar başarılı olduğunu göstermektedir. Area Under Curve değeri ne kadar yüksekse sınıflandırma o kadar başarılı olmuştur. ROC eğrisi Gerçek Pozitif oranı y ekseninde ve Yanlış Kabul oranı x ekseninde yer alacak şekilde çizilmektedir. Gerçek pozitif

oranı (TPR) duyarlılık değeri olup FPR, FP sayısının gerçekte negatif ve gerçekte negatif olup model tarafından doğru olarak hesaplanan sayıya oranıdır. Şekil 9'da DJIA için ROC curve eğrisi yer almaktadır.



Şekil 9 : DJIA- DVM ROC eğrisi.

DJIA için yapılan RFE algoritması sonucunda girdilerin sıralamasına çizelgede yer verilmiştir. Buna göre önem sırası sadece 1 olan girdiler modele eklenmiştir. 'MACD_Hist', 'ADX', 'CCI', 'Real Middle Band', 'Chaikin A/D', 'OBV', 'MOM' 'WILLR', 'Change_oil', 'Change_DAX', 'Change_gold', 'Change_Nikkei', 'Change_SP', 'Change_Shanghai' ve 'Change_USD_EUR' modelde girdi olarak kullanılmıştır. Modelde kullanılan girdi sayısı 29'tan 14'e düşmüştür. Çizelge 10'da RFE algoritmasının sonuçları yer almaktadır.

Çizelge 10 : RFE algoritma sonucu DJIA -RF.

Nr.	Özellik	Sıra
0	SMA	7
1	EMA	7
2	MACD_Signal	4
3	MACD_Hist	1
4	MACD	6

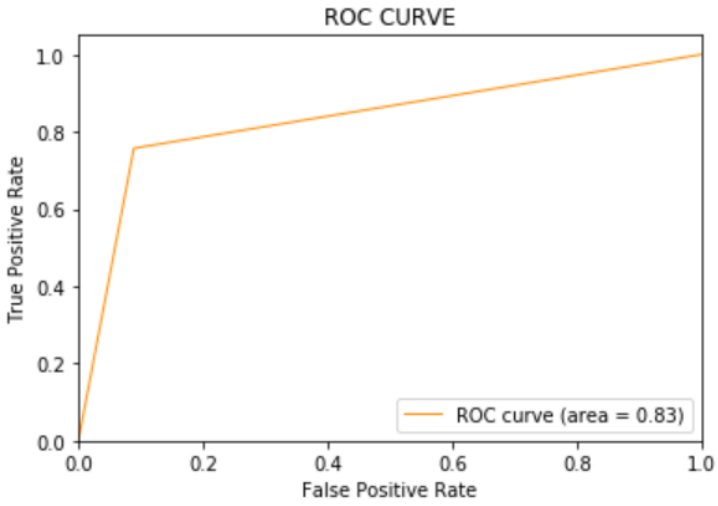
5	RSI	6
6	ADX	1
7	CCI	1
8	Aroon Up	9
9	Aroon Down	9
10	Real Middle Band	1
11	Real Upper Band	8
12	Real Lower Band	5
13	Chaikin A/D	1
14	OBV	1
15	WMA	8
16	MOM	1
17	WILLR	1
18	SlowD	4
19	SlowK	2
20	4. close	5
21	Change_oil	1
22	Change_DAX	1
23	Change_gold	1
24	Change_Nikkei	1
25	Change_SP	1
26	Change_Shanghai	1
27	Change_USD_EUR	1
28	Change_USD_GBP	3
29	Change_USD_JPY	3

DJIA için rassal orman algoritması kullanılarak yapılan sınıflandırmanın doğruluk oranı ise % 86 olarak hesaplanmıştır. Ilgili modele ait sınıflandırma raporu aşağıdaki

şekilde yer almıştır. Buna göre ekonomik girdiler dahil edildiğinde sınıflandırma başarısı en iyi model DJIA endeksi için rassal ortam algoritması olmuştur. Şekil 10'da DJIA için yapılan Rassal orman algoritma sonucu yer almaktadır. Şekil 11'de ise ilgili ROC eğrisine ait grafik yer almaktadır.

	Duyarlılık	Kesinlik	F1 Skoru
0	0.89	0.91	0.9
1	0.8	0.76	0.78
Doğruluk			0.86
Makro Ort.	0.84	0.83	0.84
Ağırlıklı Ort.	0.86	0.86	0.86

Şekil 10 : DJIA- Rassal Orman sınıflandırma raporu.



Şekil 11: DJIA- Rassal Orman ROC eğrisi.

NDX için DVM modeli kullanılarak kurulacak modele girdi olacak özellikleri belirlemek amacıyla RFE algoritması uygulanmıştır. DVM algoritmasının hem DJIA hem de NDX için tahmin gücünün oldukça düşük kalması sebebiyle tüm girdiler modelde veri olarak kullanılmıştır. Çizelge 11'de Rassal orman algoritmasında kullanılacak girdilerin sıralaması yer almaktadır.

Çizelge 11 : RFE Algoritması NDX-DVM.

Nr.	Özellik	Sıra
0	SMA	1
1	EMA	1

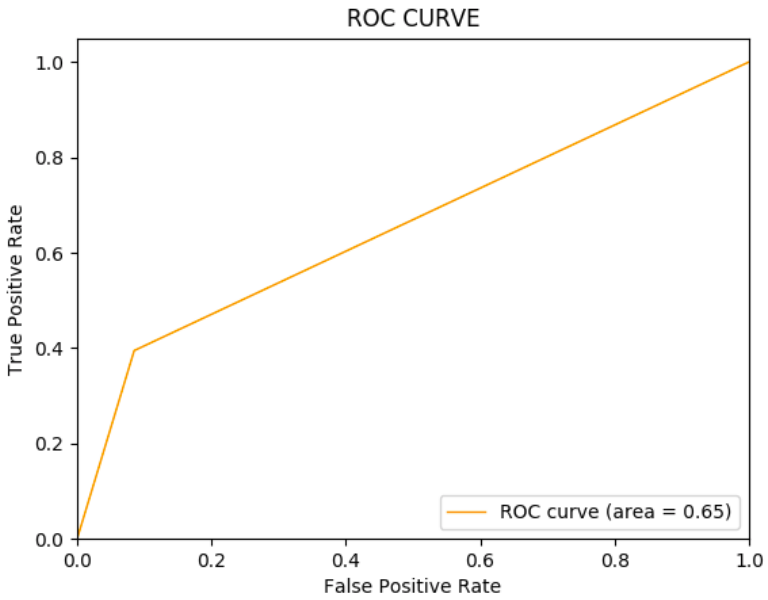
2	MACD_Signal	1
3	MACD_Hist	1
4	MACD	4
5	RSI	1
6	ADX	4
7	CCI	2
8	Aroon Down	3
9	Aroon Up	2
10	Real Upper Band	1
11	Real Lower Band	1
12	Real Middle Band	1
13	Chaikin A/D	2
14	OBV	1
15	WMA	1
16	MOM	1
17	WILLR	1
18	SlowK	1
19	SlowD	1
20	4. close	1
21	Change_oil	4
22	Change_DAX	3
23	Change_gold	3
24	Change_Nikkei	2
25	Change_SP	3
26	Change_Shanghai	4
27	Change_USD_EUR	2
28	Change_USD_GBP	3
29	Change_USD_JPY	4

NDX sadece teknik göstergelerle yapılan sınıflandırmada DVM algoritması daha yüksek bir performans elde etmiştir. Ekonomik göstergeler girdi olarak dahil edildiğinde DVM modelinin doğruluğu % 74 olarak gerçekleşmiş olup sadece teknik göstergelerin eklenmesine kıyasla artış göstermiştir. Şekil 12'de NDX'e ait DVM sınıflandırma raporu yer almaktadır.

	Duyarlılık	Kesinlik	F1 Skoru
0	0.75	0.92	0.83
1	0.7	0.39	0.5
Doğruluk			0.74
Makro Ort.	0.73	0.65	0.67
Ağırlıklı Ort.	0.73	0.74	0.72

Şekil 12 : NDX- DVM sınıflandırma raporu.

Şekilde görüleceği üzere ekonomik girdiler eklendiğinde NDX için DVM algoritmasında AUC değerinde de artış sağlanmış; model performansı iyileştirilmiştir. Şekil 13'te NDX için yapılan DVM ROC eğrisi yer almaktadır.



Şekil 13 : NDX DVM ROC eğrisi.

Rassal orman algoritmasının doğruluk oranı ise NDX için % 85 olarak hesaplanmış olup hem sadece teknik göstergelerin dahil edildiği modele kıyasla daha yüksek bir doğruluk elde etmiş; hem de DVM'nin doğruluk skorunu geçmiştir.

NDX için yapılan RFE algoritmasında girdilerin sıralanması aşağıdaki gibi gerçekleşmiştir. Buna göre modelde sıralama sonucu 1 olan özellikler girdi olarak kullanılmıştır. Bu sayede girdi sayısı 29'dan 15'e düşmüştür. Çizelge 12'de RFE algoritması kullanılarak hazırlanan özelliklerin sıralaması yer almaktadır. Şekil 14-15'te ise bu özellikler kullanılarak yapılan sınıflandırma raporu ve ROC eğrisine yer verilmiştir.

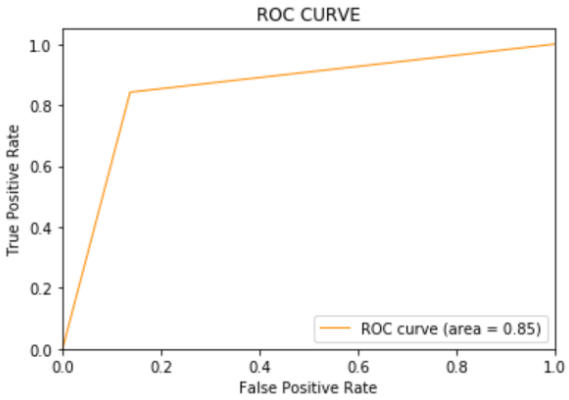
Çizelge 12 : RFE Algoritması NDX-RF.

Nr.	Özellik	Sıra
0	SMA	8
1	EMA	7
2	MACD_Signal	1
3	MACD_Hist	1
4	MACD	1
5	RSI	5

6	ADX	1
7	CCI	1
8	Aroon Down	9
9	Aroon Up	9
10	Real Upper Band	1
11	Real Lower Band	6
12	Real Middle Band	8
13	Chaikin A/D	3
14	OBV	1
15	WMA	3
16	MOM	5
17	WILLR	6
18	SlowK	1
19	SlowD	2
20	4. close	7
21	Change_oil	1
22	Change_DAX	1
23	Change_gold	1
24	Change_Nikkei	1
25	Change_SP	1
26	Change_Shanghai	1
27	Change_USD_EUR	4
28	Change_USD_GBP	1
29	Change_USD_JPY	4

	Duyarlılık	Kesinlik	F1 Skoru
0	0.92	0.86	0.89
1	0.75	0.84	0.8
Doğruluk			0.84
Makro Ort.	0.83	0.85	0.84
Ağırlıklı Ort.	0.86	0.86	0.86

Şekil 14 : NDX Rassal Orman sınıflandırma raporu.



Şekil 15 : NDX Rassal Orman ROC eğrisi.

Şekillerden anlaşılacağı üzere hem NDX hem de DJIA için sadece teknik göstergelerin girdi olarak kullanıldığı duruma kıyasla ekonomik girdilerin modele dahil edilmesi her iki algoritma için de doğruluğu artırmıştır. Sadece teknik göstergelerin kullanıldığı durumda DJIA ve NDX için DVM algoritması daha iyi bir performans sergilerken modele ekonomik girdiler eklendiğinde rassal orman algoritması hem DJIA hem de NDX için daha yüksek bir doğruluk ortaya koymuştur.

7- SONUÇ VE ÖNERİLER

Bu çalışmada NDX ve DJIA endekslerinin bir sonraki günkü kapanış değerinin yönü Rassal Orman ve DVM algoritmaları kullanılarak tahmin edilmiştir. Çalışmanın amacı hangi algoritmanın daha yüksek tahmin performansı elde ettiğini görmek ve sadece teknik göstergelerin kullanılmasının tahmin açısından değerini önemini saptamakdır. Bu çalışma Dow Jones Industrial Average ve Nasdaq 100 endekslerinin günlük kapanış verilerini Rassal orman ve destek vektör makineleri kullanarak tahmin etmiştir. Sadece teknik göstergelerin girdi olarak kullanıldığı modelde her iki endeks için de DVM algoritması başarılı çıkarken; ekonomik girdilerin eklendiği modelde Rassal orman algoritması başarılı çıkmıştır. Buna göre sadece teknik göstergelerin kullanıldığı duruma kıyasla ekonomik girdilerin eklenmesi her iki endeks ve algoritma için de doğruluk oranını önemli ölçüde artırmıştır. Modeli kurmaya geçmeden önce veri setlerini model, eğitim ve validasyon verisi olarak ayırırken test ve validasyon verisinin öğrenme verisinden sonra gelmesi, öğrenme verisinin gelecekteki verilerle eğitilmemesi açısından önemlidir.

Aynı zamanda modeli kurarken sınıflar arası dengesizlik olduğu gözlemlenmiş, bu amaçla veriler yeniden örneklendirilmiştir. Çalışmada bugüne kadar yapılmış olan çoğu araştırmada kullanılan teknik göstergeler ve ekonomik girdiler özellik olarak kullanılmıştır ve herhangi bir özellik seçim algoritması uygulanmamıştır. Ancak bazı algoritmalar bazı girdilerle daha yüksek bir doğruluk elde etmektedir. Recursive Feature Elimination (RFE) denilen algoritma kullanılarak özellikler önem sırasına göre eklenebilir ve böylelikle modele çok katkısı olmayan özellikler modele dâhil edilmez.

Modele girdi olarak dâhil edilen özelliklerin boyutu scikit learn kütüphanesindeki bazı özellikler kullanılarak daraltılabilir. Principal Component analysis kullanılarak benzer şekildeki özellikler kümelendirilerek modele eklenebilir. Ancak bu çalışmada böyle bir teknik kullanılmamıştır.

Modelde sadece DVM ve Rassal orman algoritmaları kullanılmış diğer makine öğrenmesi algoritmalarına yer verilmemiştir. Son zamanlarda hisse senedi fiyat tahmininde sıklıkla kullanılan yapay sinir ağları da tahmin amacıyla kullanılabilir, böylelikle model sonuçlarının daha doğru bir şekilde karşılaştırılması sağlanılabilir.

Modelde hedef olarak sadece bir sonraki günün kapanış değeri kullanılmıştır. Modelin karşılaştırmalı analizini daha doğru bir biçimde ölçmek için farklı hedef değişkenleri de kullanılabilir. Örneğin modelde hem kapanış, hem açılış hem de getiri hedef değişkenlerine bakılarak hangi modelin hangi hedefe yönelik daha iyi tahmin yaptığı analiz edilebilir.

Modele eklenen ekonomik girdilerden endeks değerlerine yönelik herhangi bir seçim kriteri eklenmemiştir. Nikkei, DAX, S&P ve Shangai Composite endeksleri modele rassal dahil edilmiştir. Ancak modelin tahmin özelliklerinin geliştirilmesi açısından korelasyon matrisi yapılarak DJIA ve NDX endeksleriyle dünyadaki hangi başka endekslerin korelasyonlu olduğu analiz edilerek modelde kullanılabilir.

DVM modelinde RBF çekirdek fonksiyonları sınıflandırma başarısında önemlidir. Bu sebeple farklı çekirdek fonksiyonları ve parametreleri seçilerek modelin sınıflandırma gücü karşılaştırılabilir. Ayrıca sadece DVM ve Rassal Orman algoritmalarının dışında diğer sınıflandırma algoritmaları da modele dahil edilerek sınıflandırma başarısı karşılaştırılabilir.

Son olarak çalışma sonucunda işlem stratejisi uygulanarak modelin bize gelecek fiyat ya da getiri ile ilgili bilgi vermesi sağlanabilirdi. Örneğin modelin tahminin fiyatların düşeceği yönünde olması durumunda “sat”; tersi durumda “al” şeklinde bir işlem modeli uygulanarak modelin getiri grafiği eklenebilir.

KAYNAKLAR

Afşar, M. (2006). Finansal Sistem ve İşleyişi.

Aras, G. (2003). *Sermaye piyasalarının gelişmesinde kurumsal yatırımcıların rolü: OECD ülkeleri ve Türkiye örneği*. İstanbul: Kurumsal Yatırımcı Yöneticileri Derneği.

Cao, L. (1997). Practical method for determining the minimum embedding dimension of a scalar time series. *Physica D* 110, 43–50

Cao, L., & Tay, F. E. (2001). Financial Forecasting Using Support Vector Machines. *Neural Computing & Applications, 10*(2), 184-192. doi:10.1007/s005210170010

Chen, A., Leung, M. T., & Daouk, H. (2003). Application of neural networks to an emerging financial market: Forecasting and trading the Taiwan Stock Index. *Computers & Operations Research, 30*(6), 901-923. doi:10.1016/s0305-0548(02)00037-0

Chen, A., Leung, M. T., & Daouk, H. (2003). Application of neural networks to an emerging financial market: Forecasting and trading the Taiwan Stock Index. *Computers & Operations Research, 30*(6), 901-923. doi:10.1016/s0305-0548(02)00037-0

Chen, X., & Jeong, J. C. (2007). Enhanced recursive feature elimination. *Sixth International Conference on Machine Learning and Applications (ICMLA 2007)*. doi:10.1109/icmla.2007.35

Coşkun, M. (2010). *Para ve sermaye piyasaları: Kurumlar, araçlar, analiz*. Ankara: Detay Yayıncılık.

De Oliveira, F. A., Nobre, C. N., &Zárate, L. E., (2013). Applying Artificial Neural Networks to prediction of stock price and improvement of the directional prediction index –Case study of PETR4, Petrobras, Brazil. *Expert Systems with Applications, 40*, 7596-7606. http://doi.org/10.1016/j.eswa.2013.06.071

Devroye, L. (1978). A universal k-nearest neighbor procedure in discrimination. In Proceedings of the 1978 *IEEE Computer Society Conference on Pattern Recognition and Image Processing*, pages 142-147. IEEE Computer Society, Long Beach, CA.

Ding, Y., Song, X., & Zen, Y. (2008). Forecasting financial condition of Chinese listed companies based on support vector machine. *Expert Systems with Applications, 34*(4), 3081-3089. doi:10.1016/j.eswa.2007.06.037

Gestel, T. V., Suykens, J., Baestaens, D., Lambrechts, A., Lanckriet, G., Vandaele, B., . . . Vandewalle, J. (2001). Financial time series prediction using least squares support vector machines within the

evidence framework. *IEEE Transactions on Neural Networks, 12*(4), 809-821. doi:10.1109/72.935093

Gopika, N., & M.e., A. M. (2018). Correlation Based Feature Selection Algorithm for Machine Learning. 2018 3rd International Conference on Communication and Electronics Systems (ICCES). doi:10.1109/cesys.2018.8723980

Howley, T., Madden, M. G., O'Connell, M., & Ryder, A. G. (n.d.). The Effect of Principal Component Analysis on Machine Learning Accuracy with High Dimensional Spectral Data. *Applications and Innovations in Intelligent Systems XIII*, 209-222. doi:10.1007/1-84628-224-1_16

Hu, Y., Su, P., Hao, X., & Tang, F. (2009). The Long-Term Predictive Effect of SVM Financial Crisis Early-Warning Model. *2009 First International Conference on Information Science and Engineering*. doi:10.1109/icise.2009.1230

Huang, W. (2004). Forecasting stock market movement direction with support vector machine. *Computers & Operations Research*. doi:10.1016/s0305-0548(04)00068-1

Huang, W. (2004). Forecasting stock market movement direction with support vector machine. *Computers & Operations Research*. doi:10.1016/s0305-0548(04)00068-1

Huerta, R., Elkan, C., & Corbacho, F. (2011). Nonlinear Support Vector Machines Can Systematically Identify Stocks with High and Low Future Returns. *SSRN Electronic Journal*. doi:10.2139/ssrn.1930709

Isabelle Guyon, I., Gunn, S., & Zadeh, L. (2006). Feature Extraction. *Studies in Fuzziness and Soft Computing*. doi:10.1007/978-3-540-35488-8

Kara, Y., Boyacioglu, M. A., & Baykan, Ö . K. (2011). Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange. Expert Systems with Applications, 38, 5311-5319. https://doi.org/10.1016/j.eswa.2010.10.027

Kashyap, P. (2017). I Am the Future: Machine Learning in Action. *Machine Learning for Decision Makers*, 235-260. doi:10.1007/978-1-4842-2988-0_6

Keho, Y. (2012). The Basics of Linear Principal Components Analysis. *Principal Component Analysis*. doi:10.5772/38577

Khan, J. A., Aelst, S. V., & Zamar, R. H. (2007). Building a robust linear model with forward selection and stepwise procedures. *Computational Statistics & Data Analysis, 52*(1), 239-248. doi:10.1016/j.csda.2007.01.007

Kim, K. (2003). Financial time series forecasting using support vector machines. *Neurocomputing,55*(1-2), 307-319. doi:10.1016/s0925-2312(03)00372-2

Kumar, M., & M., T. (2006). Forecasting Stock Index Movement: A Comparison of Support Vector Machines and Random Forest. *SSRN Electronic Journal*. doi:10.2139/ssrn.876544

Levine, R. (2004). Finance and Growth: Theory and Evidence. doi:10.3386/w10766

Lin, Y., Guo, H., & Hu, J. (2013). An SVM-based approach for stock market trend prediction. *The 2013 International Joint Conference on Neural Networks (IJCNN)*. doi:10.1109/ijcnn.2013.6706743

Mcgroarty, F., Booth, A., Gerding, E., & Chinthalapati, V. L. (2018). High frequency trading strategies, market fragility and price spikes: An agent based model perspective. *Annals of Operations Research.* doi:10.1007/s10479-018-3019-4

Mennitt, D., Sherrill, K., & Fristrup, K. (2014). A geospatial model of ambient sound pressure levels in the contiguous United States. *The Journal of the Acoustical Society of America*, 135(5), 2746-2764. doi:10.1121/1.4870481

Miche, Y., Bas, P., Lendasse, A., Jutten, C., & Simula, O. (n.d.). Advantages of Using Feature Selection Techniques on Steganalysis Schemes. *Computational and Ambient Intelligence Lecture Notes in Computer Science*, 606-613. doi:10.1007/978-3-540-73007-1_73

Niaki, S. T., & Hoseinzade, S. (2013). Forecasting S&P 500 index using artificial neural networks and design of experiments. *Journal of Industrial Engineering International, 9*(1). doi:10.1186/2251-712x-9-1

O'Connor, N., & Madden, M. G. (n.d.). A Neural Network Approach to Predicting Stock Exchange Movements using External Factors. *Applications and Innovations in Intelligent Systems XIII*, 64-77. doi:10.1007/1-84628-224-1_6

Öğücü, M. O. (2006). Yapay Sinir Ağları ile Sistem Tanıma, Tez, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü.

Palacios, H. J., Pantoja, G. A., Navarro, A. A., Puetaman, I. M., & Toledo, R. A. (2016). Comparativa entre CRISP-DMy SEMMA para la limpieza de datos en productos MODIS en un estudio de cambio de cobertura y uso del suelo: Comparative between CRISP-DM and SEMMA for data cleaning of MODIS products in a study of land use and land cover change. *2016 IEEE 11th Colombian Computing Conference (CCC)*. doi:10.1109/columbiancc.2016.7750789

Prasaddas, S., & Padhy, S. (2012). Support Vector Machines for Prediction of Futures Prices in Indian Stock Market. *International Journal of Computer Applications, 41*(3), 22-26. doi:10.5120/5522-7555

Raza, M. S., & Qamar, U. (2017). Introduction to Feature Selection. Understanding and Using Rough Set Based Feature Selection: Concepts, Techniques and Applications, 1-25. doi:10.1007/978-981-10-4965-1_1

Refenes, A. N., Zapranis, A., & Francis, G. (1994). Stock performance modeling using neural networks: A comparative study with regression models. *Neural Networks, 7*(2), 375-388. doi:10.1016/0893-6080(94)90030-2

Ruei-Shan, L. (2008). A study on application of smooth support vector classification to stock selection in Taiwan's stock market. *Joint Conference of WEHIA&CIEF*. Taoyuan, Taiwan.

Samal, A. R., Mohanty, M. K., & Fifarek, R. H. (2008). Backward elimination procedure for a predictive model of gold concentration. *Journal of Geochemical Exploration, 97*(2-3), 69-82. doi:10.1016/j.gexplo.2007.11.004

Sandri, M., & Zuccolotto, P. (2006). Variable Selection Using Random Forests. *Studies in Classification, Data Analysis, and Knowledge Organization Data Analysis, Classification and the Forward Search,* 263-270. doi:10.1007/3-540-35978-8_30

Sarıkamış, C. (1980). *Sermaye pazarları*. İstanbul: İstanbul Üniversitesi.

Sewell, M. **(2010).** *The Application of Intelligent Systems to Financial Time Series Analysis*, PhD thesis, PhD dissertation, Department of Computer Science, University College London, University of London.

Suljkic, J. (2014). *Testing the Predictability of Stock Markets on Real Data*, Computer Science and Communications, Kungl. Tekniska H¨ogskolan,.

Suthaharan, S. (2016). Supervised Learning Models. *Machine Learning Models and Algorithms for Big Data Classification Integrated Series in Information Systems,* 145-181. doi:10.1007/978-1-4899-7641-3_7

Tay, F. E., & Cao, L. (2001). Application of support vector machines in financial time series forecasting. *Omega, 29*(4), 309-317. doi:10.1016/s0305-0483(01)00026-3

Tay, F. E., & Cao, L. J. (2001). Improved financial time series forecasting by combining Support Vector Machines with self-organizing feature map. *Intelligent Data Analysis, 5*(4), 339-354. doi:10.3233/ida-2001-5405

Timor, M., Dinçer, H. and Emir, Ş. (2012). Performance comparison of artificial neural network (ANN) and support vector machines (SVM) models for the stock selection problem: An application on the Istanbul Stock Exchange (ISE) - 30 index in Turkey. *African Journal Of Business Management, 6*(3). doi:10.5897/ajbm11.1802

Tsai, C., & Wang, S. (2009). Stock Price Forecasting by Hybrid Machine Learning Techniques. *Proceedings of the International MultiConference of Engineers and Computer Scientists*, 1, 18-20.

Tsipouras, M. G., Tsouros, D. C., Smyrlis, P. N., Giannakeas, N., & Tzallas, A. T. (2018). Random Forests with Stochastic Induction of Decision Trees. *2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)*. doi:10.1109/ictai.2018.00087

TÜMER, Erol. (1979). "Sermaye Piyasası ve İşletmeler Arasındaki İlişkiler", Türkiye'de Sermaye Piyasası Sorunları Semineri, İstanbul: İYSE – TÜSİAD.

Url-1 <*https://www.nasdaq.com/markets/indices/nasdaq-100.aspx*>**,** erişim tarihi 02.04.2019

Url-2 < *https://www.nasdaq.com/investing/glossary/a/algo-trading*>**,** erişim tarihi 02.04.2019

Url-3<*https://www.investopedia.com/articles/active-trading/101014/basics-algorithmic-trading-concepts-and-examples.asp*>**,** erişim tarihi 02.04.2019

Url-4<*https://www.nasdaq.com/forex/education/advantages-of-algo-trading.aspx* >**,** erişim tarihi 02.04.2019

Url-5 <*https://tr.tradingview.com/scripts/*>**,** erişim tarihi 02.04.2019

Vapnik V. (1995) The nature of statistic learning theory. New York: Springer

Vapnik, V. N. (1999). An overview of statistical learning theory. *IEEE Transactions on Neural Networks, 10*(5), 988-999. doi:10.1109/72.788640

Yu, L., Wang, S., & Lai K. L.(2009). A neural-network-based nonlinear metamodeling approach to financial time series forecasting. *Applied Soft Computing,9,*563-574. http://doi.org/10.1016/j.asoc.2008.08.001

Zhang, J., Shan, R., & Su, W. (2009). Applying Time Series Analysis Builds Stock Price Forecast Model. *Modern Applied Science, 3*(5).

EKLER

EK A: Python Kodları

EK A

```
# coding: utf-8

## Importing Libraries

import time
import random
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from alpha_vantage.timeseries import TimeSeries
from alpha_vantage.techindicators import TechIndicators
from sklearn.preprocessing import StandardScaler
from sklearn.utils import resample
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.metrics import accuracy_score, f1_score, roc_auc_score
from sklearn.svm import SVC
from sklearn.svm import LinearSVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.metrics import roc_curve, auc
from prettytable import PrettyTable

## Inserting Extra Features

## Brent Oil Futures Historical Data

brent_oil = pd.read_csv('Brent Oil Futures Historical Data.csv')
brent_oil.head()

## ----------------------------------------------------------------------------------------------------
---

## Dataset analysis

brent_oil[brent_oil.isnull().any(axis=1)].head()
np.sum(brent_oil.isnull().any(axis=1))
brent_oil.isnull().any(axis=0)
brent_oil.info()
brent_oil.describe()
brent_oil=brent_oil.rename(columns={'Change %':'Change_oil'})
brent_oil=brent_oil[['Date','Change_oil']]
brent_oil.head()

## DAX Historical Data
```

```
DAX= pd.read_csv('DAX Historical Data.csv')
DAX.head()

## ------------------------------------------------------------------------------------------
---

## Dataset analysis

DAX[DAX.isnull().any(axis=1)].head()
np.sum(DAX.isnull().any(axis=1))
DAX.isnull().any(axis=0)
DAX.info()
DAX.describe()

## Getting change feature

DAX=DAX.rename(columns={'Change %':'Change_DAX'})
DAX=DAX[['Date','Change_DAX']]
DAX.head()

## Merging The data for change feature

change_feature =pd.merge(brent_oil, DAX, on='Date')
change_feature.head()
gold_features = pd.read_csv('Gold Futures Historical Data.csv')
gold_features.head()

## ------------------------------------------------------------------------------------------
---

## Dataset analysis

gold_features[gold_features.isnull().any(axis=1)].head()
np.sum(gold_features.isnull().any(axis=1))
gold_features.isnull().any(axis=0)
gold_features.info()
gold_features.describe()

## Getting change feature

gold_features=gold_features.rename(columns={'Change %':'Change_gold'})
gold_features=gold_features[['Date','Change_gold']]
gold_features.head()

## Merging The data for change feature

change_feature =pd.merge(change_feature, gold_features, on='Date')
change_feature.head()
```

```
## Nikkei 225 Historical Data

Nikkei= pd.read_csv('Nikkei 225 Historical Data.csv')
Nikkei.head()

## ------------------------------------------------------------------------------------------
---

## Dataset analysis

Nikkei[Nikkei.isnull().any(axis=1)].head()
np.sum(Nikkei.isnull().any(axis=1))
Nikkei.isnull().any(axis=0)
Nikkei.info()
Nikkei.describe()

## Getting change feature

Nikkei=Nikkei.rename(columns={'Change %':'Change_Nikkei'})
Nikkei=Nikkei[['Date','Change_Nikkei']]
Nikkei.head()

## Merging The data for change feature

change_feature =pd.merge(change_feature, Nikkei, on='Date')
change_feature.head()

## S&P 500 Historical Data

SP= pd.read_csv('S&P 500 Historical Data.csv')
SP.head()

## ------------------------------------------------------------------------------------------
---

## Dataset analysis

SP[SP.isnull().any(axis=1)].head()
np.sum(SP.isnull().any(axis=1))
SP.isnull().any(axis=0)
SP.info()
SP.describe()

## Getting change feature

SP=SP.rename(columns={'Change %':'Change_SP'})
SP=SP[['Date','Change_SP']]
SP.head()

## Merging The data for change feature
```

```
change_feature =pd.merge(change_feature, SP, on='Date')
change_feature.head()

## Shanghai Composite Historical Data

Shanghai= pd.read_csv('Shanghai Composite Historical Data.csv')
Shanghai.head()

## ----------------------------------------------------------------------------------------------------
---

## Dataset analysis

Shanghai [Shanghai .isnull().any(axis=1)].head()
np.sum(Shanghai .isnull().any(axis=1))
Shanghai .isnull().any(axis=0)
Shanghai .info()
Shanghai .describe()

## Getting change feature

Shanghai=Shanghai.rename(columns={'Change %':'Change_Shanghai'})
Shanghai=Shanghai[['Date','Change_Shanghai']]
Shanghai.head()

## Merging The data for change feature

change_feature =pd.merge(change_feature, Shanghai, on='Date')
change_feature.head()

## USD_EUR Historical Data

USD_EUR= pd.read_csv('USD_EUR Historical Data.csv')
USD_EUR.head()

## ----------------------------------------------------------------------------------------------------
---

## Dataset analysis

USD_EUR [USD_EUR .isnull().any(axis=1)].head()
np.sum(USD_EUR .isnull().any(axis=1))
USD_EUR .isnull().any(axis=0)
USD_EUR .info()
USD_EUR .describe()

## Getting change feature

USD_EUR=USD_EUR.rename(columns={'Change %':'Change_USD_EUR'})
```

```
USD_EUR=USD_EUR[['Date','Change_USD_EUR']]
USD_EUR.head()

## Merging The data for change feature

change_feature =pd.merge(change_feature, USD_EUR, on='Date')
change_feature.head()

## USD_GBP Historical Data

USD_GBP= pd.read_csv('USD_GBP Historical Data.csv')
USD_GBP.head()

## ----------------------------------------------------------------------------------------------
---

## Dataset analysis

USD_GBP [USD_GBP .isnull().any(axis=1)].head()
np.sum(USD_GBP
      .isnull().any(axis=1))
USD_GBP .isnull().any(axis=0)
USD_GBP .info()
USD_GBP .describe()

## Getting change feature

USD_GBP=USD_GBP.rename(columns={'Change %':'Change_USD_GBP'})
USD_GBP=USD_GBP[['Date','Change_USD_GBP']]
USD_GBP.head()

## Merging The data for change feature

change_feature =pd.merge(change_feature, USD_GBP, on='Date')
change_feature.head()

## USD_JPY Historical Data

USD_JPY= pd.read_csv('USD_JPY Historical Data.csv')
USD_JPY.head()

## ----------------------------------------------------------------------------------------------
---

## Dataset analysis

USD_JPY [USD_JPY .isnull().any(axis=1)].head()
np.sum(USD_JPY .isnull().any(axis=1))
USD_JPY.isnull().any(axis=0)
USD_JPY .info()
```

```
USD_JPY .describe()

## Getting change feature

USD_JPY=USD_JPY.rename(columns={'Change %':'Change_USD_JPY'})
USD_JPY=USD_JPY[['Date','Change_USD_JPY']]
USD_JPY.head()

## Merging The data for change feature

change_feature =pd.merge(change_feature, USD_JPY, on='Date')
change_feature
change_feature = change_feature.dropna(axis = 0, how ='any')
change_feature
change_feature.to_csv('change_feature.csv')
change_feature.head()
change_feature  = pd.read_csv('change_feature.csv', index_col='Date',
parse_dates=['Date'])
change_feature .head()
change_feature=change_feature.drop(columns=['Unnamed: 0'])
change_feature=change_feature.rename(columns={'Date':'date'})
aa=change_feature.iloc[::-1]
aa.to_csv('change_feature.csv')
change_feature  = pd.read_csv('change_feature.csv')
change_feature=change_feature.rename(columns={'Date':'date'})
change_feature.head()

## Cleaning % from all coloumns of change_feature

for letter in '%':
    change_feature['Change_oil']= change_feature['Change_oil'].str.replace(letter,'')

for letter in '%':
    change_feature['Change_DAX']=
change_feature['Change_DAX'].str.replace(letter,'')

for letter in '%':
    change_feature['Change_gold']=
change_feature['Change_gold'].str.replace(letter,'')

for letter in '%':
    change_feature['Change_Nikkei']=
change_feature['Change_Nikkei'].str.replace(letter,'')

for letter in '%':
    change_feature['Change_SP']= change_feature['Change_SP'].str.replace(letter,'')

for letter in '%':
    change_feature['Change_Shanghai']=
change_feature['Change_Shanghai'].str.replace(letter,'')
```

```
for letter in '%':
    change_feature['Change_USD_EUR']=
change_feature['Change_USD_EUR'].str.replace(letter,'')

for letter in '%':
    change_feature['Change_USD_GBP']=
change_feature['Change_USD_GBP'].str.replace(letter,'')

for letter in '%':
    change_feature['Change_USD_JPY']=
change_feature['Change_USD_JPY'].str.replace(letter,'')

change_feature

def get_data(symbol):

    # API KEY: 51N1JMO66W56AYA3
    ti = TechIndicators(key='51N1JMO66W56AYA3', output_format='pandas')
    sma, _ = ti.get_sma(symbol=symbol, interval='daily')
    wma, _ = ti.get_wma(symbol=symbol, interval='daily')
    ema, _ = ti.get_ema(symbol=symbol, interval='daily')
    macd, _ = ti.get_macd(symbol=symbol, interval='daily')
    stoch, _ = ti.get_stoch(symbol=symbol, interval='daily')
    #  Alpha Vantage Times out for more than 5 request
    time.sleep(65)
    rsi, _ = ti.get_rsi(symbol=symbol, interval='daily')
    adx, _ = ti.get_adx(symbol=symbol, interval='daily')
    cci, _ = ti.get_cci(symbol=symbol, interval='daily')
    aroon, _ = ti.get_aroon(symbol=symbol, interval='daily')
    bbands, _ = ti.get_bbands(symbol=symbol, interval='daily')
    time.sleep(65)
    ad, _ = ti.get_ad(symbol=symbol, interval='daily')
    obv, _ = ti.get_obv(symbol=symbol, interval='daily')
    mom, _ = ti.get_mom(symbol=symbol, interval='daily')
    willr, _ = ti.get_willr(symbol=symbol, interval='daily')
    time.sleep(65)
    tech_ind = pd.concat([sma, ema, macd,  rsi, adx, cci, aroon, bbands, ad, obv, wma,
mom, willr, stoch], axis=1)

    ts = TimeSeries(key='51N1JMO66W56AYA3', output_format='pandas')
    close2 = ts.get_daily(symbol=symbol, outputsize='full')
    close = ts.get_daily(symbol=symbol, outputsize='full')[0]['4. close']
    direction = (close > close.shift()).astype(int)
    target = direction.shift(-1).fillna(0).astype(int)
    target.name = 'target'

    data = pd.concat([tech_ind, close, target], axis=1)

    return data
```

```
def get_indicators(data, n):

    hh = data['2. high'].rolling(n).max()
    ll = data['3. low'].rolling(n).min()
    up, dw = data['4. close'].diff(), -data['4. close'].diff()
    up[up<0], dw[dw<0] = 0, 0
    macd = data['4. close'].ewm(12).mean() - data['4. close'].ewm(26).mean()
    macd_signal = macd.ewm(9).mean()
    tp = (data['2. high'] + data['3. low'] + data['4. close']) / 3
    tp_ma = tp.rolling(n).mean()
    indicators = pd.DataFrame(data=0, index=data.index,
                          columns=['sma', 'ema', 'momentum',
                                   'sto_k', 'sto_d', 'rsi',
                                   'macd', 'lw_r', 'a/d', 'cci'])
    indicators['sma'] = data['4. close'].rolling(10).mean()
    indicators['ema'] = data['4. close'].ewm(10).mean()
    indicators['momentum'] = data['4. close'] - data['4. close'].shift(n)
    indicators['sto_k'] = (data['4. close'] - ll) / (hh - ll) * 100
    indicators['sto_d'] = indicators['sto_k'].rolling(n).mean()
    indicators['rsi'] = 100 - 100 / (1 + up.rolling(14).mean() / dw.rolling(14).mean())
    indicators['macd'] = macd - macd_signal
    indicators['lw_r'] = (hh - data['4. close']) / (hh - ll) * 100
    indicators['a/d'] = (data['2. high'] - data['4. close'].shift()) / (data['2. high'] - data['3.
low'])
    indicators['cci'] = (tp - tp_ma) / (0.015 * tp.rolling(n).apply(lambda x: np.std(x)))

    return indicators

def rebalance(unbalanced_data):

    # Separate majority and minority classes
    data_minority = unbalanced_data[unbalanced_data.target==0]
    data_majority = unbalanced_data[unbalanced_data.target==1]

    # Upsample minority class
    n_samples = len(data_majority)
    data_minority_upsampled = resample(data_minority, replace=True,
n_samples=n_samples, random_state=5)

    # Combine majority class with upsampled minority class
    data_upsampled = pd.concat([data_majority, data_minority_upsampled])

    data_upsampled.sort_index(inplace=True)

    # Display new class counts
    data_upsampled.target.value_counts()

    return data_upsampled
```

```
def normalize(x):

    scaler = StandardScaler()
    x_norm = scaler.fit_transform(x.values)
    x_norm = pd.DataFrame(x_norm, index=x.index, columns=x.columns)

    return x_norm


def scores(models, X, y):

    for model in models:
        y_pred = model.predict(X)
        acc = accuracy_score(y, y_pred)
        f1 = f1_score(y, y_pred)
        auc = roc_auc_score(y, y_pred)
        print("Accuracy Score: {0:0.2f} %".format(acc * 100))
        print("F1 Score: {0:0.4f}".format(f1))
        print("Area Under ROC Curve Score: {0:0.4f}".format(auc))


symbolin = 'NDX'  # DJIA, NDX
data = get_data(symbolin)
data.tail(10)
data.describe()
ax = data['4. close'].plot(figsize=(9, 5))
ax.set_ylabel("Price ($)")
ax.set_xlabel("Time")
plt.show()
data_train = data['2012-01-01':'2018-01-01']
data_train = rebalance(data_train)
y = data_train.target
X = data_train.drop('target', axis=1)
X = normalize(X)
data_val = data['2018-01-01':]
y_val = data_val.target
X_val = data_val.drop('target', axis=1)
X_val = normalize(X_val)

## API Data

X
X.to_csv('X.csv')

## Adding Change features in API data

X=pd.read_csv('X.csv')
X =pd.merge(X, change_feature, on='date')
X
y.head()
```

```
y.to_csv('y.csv')
y=pd.read_csv('y.csv')
y=pd.read_csv('y.csv')
y.columns = ['date', 'y']
y.head()
X =pd.merge(X, y, on='date')
X

## Dataset analysis

X [X .isnull().any(axis=1)].head()
np.sum(X
      .isnull().any(axis=1))
X .isnull().any(axis=0)
X .info()
class_y=X.y
labels_X=X.drop(columns=['y','date'])

## SVM Support Vector Machine

y=y
X_train, X_test, y_train, y_test = train_test_split(labels_X, class_y, test_size=0.10,
shuffle=True, random_state=2000)
svc=SVC(random_state=5)
svc= svc.fit(X_train , y_train)
svc
y_pred1 = svc.predict(X_test)
print('Accuracy score= {:.2f}'.format(svc.score(X_test, y_test)))

## Precision, Recall, F1
print('\n')
print("Precision, Recall, F1")
print('\n')
CR=classification_report(y_test, y_pred1)
print(CR)
print('\n')

## ROC CURVE
fpr, tpr, thresholds = roc_curve(y_test, y_pred1)
roc_auc = auc(fpr, tpr)
plt.figure()
plt.plot(fpr, tpr, color='darkorange', lw=1, label='ROC curve (area = %0.2f)' %
roc_auc)
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC CURVE')
plt.legend(loc="lower right")
plt.show()
```

```
## Random Forest

Ran_For= RandomForestClassifier(n_estimators=1000,max_depth=30,
random_state=100,max_leaf_nodes=1000)
Ran_For= Ran_For.fit(X_train , y_train)
Ran_For
y_pred2 = Ran_For.predict(X_test)
print('Accuracy score= {:.2f}'.format(Ran_For.score(X_test, y_test)))

## Precision, Recall, F1

print('\n')
print("Precision, Recall, F1")
print('\n')
CR=classification_report(y_test, y_pred2)
print(CR)
print('\n')

## ROC CURVE

fpr, tpr, thresholds = roc_curve(y_test, y_pred2)
roc_auc = auc(fpr, tpr)
plt.figure()
plt.plot(fpr, tpr, color='darkorange', lw=1, label='ROC curve (area = %0.2f)' %
roc_auc)
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC CURVE')
plt.legend(loc="lower right")
plt.show()

## Comparison of Results

x = PrettyTable()
print('\n')
print("Detailed Performance of the all models")
x.field_names = ["Model", "Accuracy"]
x.add_row(["SVM", 0.72])
x.add_row(["RandomForestClassifier", 0.86])
print(x)
print('\n')
x = PrettyTable()
print('\n')
print("Best Model.")
x.field_names = ["Model", "Accuracy"]
x.add_row(["RandomForestClassifier", 0.86])
print(x)
```

ÖZGEÇMİŞ



Ad-Soyad **: Sezer Aktaş**

Doğum Tarihi ve Yeri **: 03.07.1989 Muş**

E-posta **:** sezeraktas89@gmail.com

ÖĞRENİM DURUMU:

- **Lisans** **:** 2013, Yeditepe Üniversitesi , İİBF, Siyasal Bilimler

MESLEKİ DENEYİM VE ÖDÜLLER:

- 2015-…:Borsa İstanbul Grubu- Takas İstanbul- Borçlanma Araçları Takas Ekibi nbUzman