T.C.

# MANİSA CELAL BAYAR UNIVERSITY

## GRADUATE SCHOOL OF NATURAL and APPLIED SCIENCE INSTITUTE

**MASTER THESIS**
**COMPUTER ENGINEERING DEPARTMENT**
**COMPUTER ENGINEERING**

# STOCK PRICE PREDICTION USING MACHINE LEARNING ALGORITHMS

**Author Name**
**Umut DÖKMEN**

**Supervisor**
**Assist. Prof Dr Hakan Murat KARACA**

**MANİSA–2023**

| | | |
|---|---|---|
| **Umut DÖKMEN** | **STOCK PRICE PREDICTION USING MACHINE LEARNING ALGORITHMS** | **2023** |

**THESIS APPROVAL**

The thesis study named "Stock Price Prediction" prepared by Umut Dökmen was defended as a **MASTER'S THESIS** in Manisa Celal Bayar University Institute of Science and Technology **Computer Engineering Department** in front of the following jury members on 04/07/2023 and was **unanimously** accepted as successful.

| | |
|---|---|
| **Supervisor** | **Dr. Öğretim Üyesi Hakan M. KARACA**<br>Manisa Celal Bayar Üniversitesi |
| **Jury Member** | **Dr. Öğretim Üyesi Volkan ALTINTAŞ**<br>Manisa Celal Bayar Üniversitesi |
| **Jury Member** | **Dr. Öğretim Üyesi Emre ŞATIR**<br>İzmir Katip Çelebi Üniversitesi |

**COMMITMENT**

I declare that this thesis was written in Manisa Celal Bayar University, Faculty of Engineering Department of Computer Engineering in accordance with academic and ethical rules, and all the literature information used is included in the thesis with reference.

**UMUT DÖKMEN**

## CONTENTS

**Page**

## SYMBOLS AND ABBREVIATIONS LIST

| | |
|---|---|
| **ACT** | Activation function |
| **ANN** | Artificial Neural Network |
| **BRR** | Bayesian Ridge Regression |
| **BWE** | Backward elimination method |
| **CV** | Cross validation |
| **DTR** | Decision tree regression |
| **FFS** | Forward feature selection method |
| **LR** | Learning rate |
| **LSTM** | Long-short term memory |
| **MA** | Maximum absolute value scaler |
| **MAE** | Mean absolute error |
| **MF** | Max features |
| **MIN-MAX** | Min-Max normalization |
| **MLR** | Multiple linear regression |
| **MLN** | Max leaf nodes |
| **MLP** | Multilayer perceptron |
| **MSE** | Mean squared error |
| **MSL** | Min samples leaf |
| **ReLU** | Rectified linear unit |
| **RMSE** | Rooted mean square error |
| **SP 500** | Standard and Poor's 500 |
| **SS** | Standart scaler |
| **SVM** | Support vector machines |
| **SVR** | Support vector regression |
| **TCMB** | Türkiye Cumhuriyeti Merkez Bankası |

**FIGURE LIST**

**Page**

**LIST OF TABLES**

**ACKNOWLEDGEMENT**

My advisor Assist Prof. Dr Hakan Murat Karaca, who supported me at every stage of my work and guided me with his knowledge and experience. My dear friends Research Assistant Turan Göktuğ Altundoğan, Research Assistant Esat Fazlullah Çelik, whose moral support I always felt during my studies, and my family who supported me financially and morally throughout my education life and always stood by me.

Umut DÖKMEN
Manisa, 2023

**ABSTRACT**

**M.Sc Thesis**

**Umut DÖKMEN**

**Manisa Celal Bayar University**
**Graduate School of Applied and Natural Sciences**
**Department of Computer Engineering**

**Supervisor: Assist. Prof. Dr. Hakan Murat KARACA**

Stock prices are difficult to predict as they are affected by many variables. However, it is possible to predict stock prices with today's computers using machine learning algorithms.

In our study, the daily value prediction was made by collecting the data of the first 5 stocks with the highest market value traded in the BIST 100 between 2016-2020 for about 5 years. Multiple linear regression, bayesian regression, random forest regression, decision tree regression, support vector regression, artificial neural network algorithms were applied to include maximum 22 features in machine learning and the results were compared. The most successful result was obtained in the artificial neural networks algorithm. Normalization, cross validation, parameter optimization, feature selection have been applied to achieve the highest success.

**Keywords:(Machine Learning, Algorithms, Stocks, Regression, Supervised Learning)**

**2023, 84 pages**

# ÖZET


**Yüksek Lisans Tezi**

**Manisa Celal Bayar Üniversitesi**
**Fen Bilimleri Enstitüsü**
**Bilgisayar Mühendisliği Anabilim Dalı**

**Danışman: Dr Öğretim Üyesi Hakan Murat Karaca**

Hisse senedi fiyatları birçok değişkenden etkilendiği için tahmin edilmesi zordur. Ancak makine öğrenmesi algoritmalarını kullanan günümüz bilgisayarları ile hisse senedi fiyatlarının tahmini mümkündür.

Çalışmamızda 2016-2020 yılları arasında BIST 100'de işlem gören piyasa değeri en yüksek ilk 5 hisse senedinin yaklaşık 5 yıllık verileri toplanarak günlük değer tahmini yapılmıştır. En fazla 22 öznitelik makine öğrenmesine dahil edilmek üzere çoklu doğrusal regresyon, bayesian regresyon, rastgele orman, karar ağaçları, destek vektör makineleri, yapay sinir ağları algoritmaları uygulanmış ve sonuçlar karşılaştırılmıştır. En başarılı sonuç yapay sinir ağları algoritmasında elde edilmiştir. En yüksek başarı elde edilebilmesi için normalizasyon, çapraz doğrulama, parametre optimizasyonu ve öznitelik seçimi uygulanmıştır.



**Anahtar Kelimeler:(Makine Öğrenmesi, Algoritmalar, Hisse Senedi, Regresyon, Denetimli Öğrenme)**

**2023, 84 sayfa**

## 1. INTRODUCTION

A stock is a part of the principal of a company. People who buy a company's stock share in the profit and loss of that company. The stock signifies the special relationship between the company and the person who buys the stock [1]. Companies open their stocks to investors through the stock market in order to increase their financial capacity and their capital. The expectation that the value of the stock will increase creates demand for that stock. This demand increases the value of the stock. On the contrary, the expectation that the value of the stock will decrease requires selling the stock and the price will decrease. Investors aim to make a profit by buying stocks that will rise in the future. For this reason, it is very important for investors to be able to predict the stock price.

Machine learning is widely used in the field of finance, as it is in many fields. Many companies use machine learning in stock trading. It is able to make very wise investment decisions and reduce financial risks for people. Many studies have shown that machine learning-based applications are more successful than traditional stock trading strategies. These results increase the applications of artificial intelligence and machine learning in the field of finance day by day [2,3].

Stock prices are volatile. There are many internal and external factors that affect the stock price. Internal factors, profit distribution policy, capital increase, financial structure, management, field of activity of the enterprise [4]. In this study, Opening Price, High Price, Low Price, Volume, Net Profit for the Period, Resource and Dividend Income Factors, which are internal variables affecting the stock price, are included in the calculations. External factors included in the calculation in this study: BIST Stars Tradded Value, BIST Stars Tradded Volume, BIST 100 Index, BIST 100 Volume (TL), BIST 100 Difference, Dollar-TL, Euro-TL, XAU-USD, Brent Oil, S&P 500 Index, Euro Stoxx 50 Index, Interest and Inflation. These factors affect the stock price differently. The effect of these factors on the stock price can be calculated by statistical methods. But there are other factors that affect the stock price. These are the political situation in the country and the world, financial expectations, sectoral expectations, unexpected events. The political situation in the country and the world or natural disasters cannot be predicted by numerical methods. However,

since the policies of the country and the world will affect the external factors used in this study, the indirect effect on the stock can be calculated.

In this study, the daily closing price of the stocks of the top 5 companies with the highest market value in BIST 100 is estimated. For the training of machine learning models, approximately 5 years of historical data between 2016 and 2020 were collected and combined from Borsa İstanbul, investing.com and isyatırım.com. In the study, 80% of the data set was used for training and 20% for testing. In order for machine learning algorithms to give more successful results, data preprocessing has been done. The effect of the normalization methods used in the data set on the model success was investigated. The parameter changes in the models were examined and their effects on the results were investigated. Optimum parameters are selected to get the highest success. By using feature selection methods, the success of the model has been increased. The success of machine learning algorithms used in the study has been compared.

The study consists of 5 chapters. In the first chapter, general information about the study is given and the study is summarized. In the second part, information about the theoretical concepts used in the study is given and a literature review is made. In chapter 3, the machine learning algorithms used in the study are introduced, the tools and technologies used in the study are explained, the steps and methods used in the study are explained. In Chapter 4, the variables affecting the results of the study were evaluated. In the 5th capter, the results are analyzed.

One of the aims of the study is to predict the stock price and give direction to the investors. Another purpose is to analyze the factors affecting stock prices with machine learning methods and to give an idea to financial analysts. Another purpose is to optimize prediction success by using internal and external factors that affect the price of stocks together as features.

## 2. MACHINE LEARNING

An algorithm is a set of rules in a specific order to solve a problem, achieve a goal, or do a job. A machine learning algorithm is a program which convert data set to a model. It makes predictions from inputs. The model provides predictions.

Machine learning is an artificial intelligence field that aims to give the machine the ability to learn without programming it directly. There are broadly three types of machine learning: supervised learning, unsupervised learning, and reinforcement learning. In stock price prediction, the supervised learning technique, which covers all prediction problems, is used because the future price is predicted from the past, known data set. Since the price which is the output we get in the stock price prediction is numerical, the task is called prediction. To predict stock price, the computer learns patterns from past stock prices. The difference between the predicted price and the actual price is called the loss function. The machine improves its performance a little more with each experience. In practice, experience means training data. Therefore, we cannot easily distinguish between machine learning and statistical approaches. The goal of supervised learning is minimizing the loss function. In the stock price prediction machine tries to minimize the difference between actual stock price with predicted stock price. In supervised learning the machine learns a predictive model that maps the features of the data to an output. Machine aims to learn a model predicting parameters [5,6].

On the other hand, in unsupervised learning, there is no specific output like clustering tasks.

### 2.1 Literature Review

Selçuk BALİ, Mehmet Ozan CİNEL, Ali Haydar GÜNDAY examined the factors affecting BIST100 in their study named “Computation Of The Effects Of Basic Macroeconomic Factors Which Affect The Share Prices To BIST 100 Index. Features are determined as Interest Rate, Money Supply, Industrial Production Index,

Inflation, Gross Domestic Product. Monthly datas for the period January 2003-May 2013 were used as the data set. The effect of the determined macroeconomic features

on the BIST 100 index was examined using multiple linear regression. In the multiple linear regression model, an inverse ratio was determined between the inflation and interest rate and the BIST 100 index, and a direct ratio between the gross domestic product and industrial production and the BIST 100 index. It is concluded that there is no relationship between money supply and BIST100. [7]

Linear Regression, Three Month Moving Average, Exponential Smoothing, Time Series Forecasting algorithms were used in the study named "Stock Market Prediction Using Machine Learning (ML)Algorithms" by M Umer Ghania, M Awaisa and Muhammad Muzammula. Apple, Google and Amazon stock data obtained from Yahoo Finance. The stock market trend for the next month is predicted and success is measured. It was observed that the results of Exponential Smoothing gave more accuracy after all the results were scaled down. [8]

Sumeet Sarode, Harsha G. Tolani, Prateek Kak, Lifna C used historical real-time data with news analysis in her study. For prediction, LSTM (Long Short-Term Memory), an artificial neural network architecture, was used. The news was collected from many companies, filtered and analyzed. Analysis results are aggregated to predict future increases. This study presents a system that decides whether to buy shares of different companies using artificial neural networks.[9]

The aim of the study " Stock Price Prediction Using Machine Learning Techniques" by Mehak Usmani, Syed Hasan Adil, Kamran Raza and Syed Saad Azhar Ali is to predict the end-of-day closing performance of Karachi Stock Exchange (KSE) using machine learning algorithms. Oil rates, gold and silver rates, interest rate, foreign exchange rate, news and social media are used as model features. Simple moving average (SMA) and Autoregressive Integrated Moving Average (ARIMA) from old statistical techniques are taken as inputs. Single-layer Perceptron Single-Layer Perceptron (SLP), Multi-Layer Perceptron (MLP), Radial Basis Function (RBF) and Support Vector Machine (SVM), which are machine learning algorithms, were analyzed and the findings obtained from the results were compared. As a result, it was determined that the Multi Layer Perceptron showed the best performance when compared to other techniques used. It was found that the feature that most affected the KSE index was the oil ratio.[10]

Abhinav Tipirisetty, in his study named "Stock Price Prediction using Deep Learning", analyzed the previous studies for stock price prediction and introduced a new approach for this. In this study, stock prices are taken as time series data and artificial neural networks are trained to learn patterns. In addition to the quantitative analysis of the stock trend, it also analyzed textual public news from online news sources. A more accurate hybrid model was created by combining numerical analysis with textual analysis. When numerical analysis was performed using the LSTM model and MSE was found 0.000453821. When the SVM model is used, MSE gives the value 0.0007262213. When text analysis is used, the stock forecast yielded an MSE result of 0.00037560132 with 78% accuracy. Therefore, the accuracy is increased when textual information is used in stock price prediction. [11]

Shubba Sinngh has collected 10 years of data from Yahoo Finance in his thesis called Stock Prediction using Machine Learning. He tried to predict the prices of Apple, Bank of America and Mc Donald stocks. In addition to linear regression, LSTM, one of the neural network-based algorithms, is used. He has included High, Low, Open, Close, Volume" columns as features. He used RMSE as evaluation metric. RMSE is 2.04 for linear regression and 0.43 for LSTM [12].

Yixin Guo tried to predict the S&P 500 index in her study called Stock Price Prediction Using Machine Learning. In this study, LSTM was examined theoretically and the study progressed on this model. Unlike qualitative analysis, quantitative analysis was used. Arima and garch models are used and compared with LSTM. MSE was used as the performance metric. The most successful result was obtained in LSTM, but the model in which 3 models were used together was more successful than the model in which LSTM was used alone [13].

### 2.2 Supervised Learning

There are two type of supervised learning problem. Regression and classification. Since price prediction will be made, our problem is a prediction problem. How much money do we make in return when we invest more money in digital advertising? When this loan is given to the customer, will the customer be

able to pay it back or not? Will the stock market index increase or decrease tomorrow?

Supervised learning can only applied to be labeled data. There is a dataset in supervised learning problems and this dataset contains training instance with correct labels. For example, suppose we have a dataset of handwritten numbers. Learning which digit these handwritten digits correspond to, a supervised learning classification algorithm tags the correct number each image corresponds to, and with it takes of thousands of handwritten digit pictures. In this way, the algorithm learns the relationship between images and numbers. It uses this relationship to classify images that the machine has never seen before, that is, without labels. Logistic regression, support vector machines, decision trees, and random forest are supervised learning.

Classification problems such as whether an e-mail is spam, whether a person has cancer or not are in the form of supervised learning. These problems have a set of variables measured or preset as input. These variables affect one or more outputs. Outputs are obtained using inputs. In statistics, the inputs can generally be called predictors or independent variables. Outputs are used as responses or dependent variables. Some of our possible inputs in stock price prediction will be features such as total transaction volume, total number of transactions, market value, USD exchange rate, gold price, BIST30 index value, BIST100 index value, daily change total transaction volume. The dependent variable that we aim to find in this study, the stock price will be represented by y and the independent variables affecting this price will be represented by Xs.

$$Y = f(X) + \epsilon$$

Epsilon represents the error in our model. This error is a theoretical limit around the performance of our algorithm.

Suppose we are trying to classify an entity as a cat or not a cat. In this classification problem, it is necessary to construct an image classification algorithm consisting of if then statements describing combinations of pixel shines. Supervised machine learning solves this problem using a computer. The computer can create heuristics by identifying models in the data set. There is a difference between human

learning and machine learning. Machine learning runs on a computer hardware. But human pattern matching works in a biological brain. By running machine-labeled training data in supervised learning, it learns from scratch the relationship between features such as total number of transactions, market value, USD exchange rate, gold price, BIST30 index value and stock price. In supervised learning, machine tries to learn from scratch how features such as total number of transactions, market value, USD exchange rate, gold price, BIST30 index value affect the stock price by running labeled training data. This learned function takes training data x as input and uses it to predict the unknown stock price y. Supervised learning aims to predict y most accurately, given examples where x is known, and y is not known. It has basically two tasks: prediction and classification.

### 2.2.1 Regression

Regression predicts numeric continuous variable y. It can predict stock prices or human age from x input features. Stock price prediction is a classic regression problem. X input properties are numeric. In order for the model to learn the relationship f between x and y well, it needs to make as many training observations as possible. The data set is divided into training and testing. In this separation, the training dataset becomes about 3 times the test dataset. There are tags in the training set. The model learns from these labeled examples. The test dataset does not have labels and the value to be predicted is not yet known. In order for the model to perform well, it must be able to generalize to situations that it has not encountered before in the test data.

$$Y = f(X) + \varepsilon, \text{ where } X = (x_1, x_{2,}\ldots,x_n)$$

where X can be any dimension. X is a vector when x is one-dimensional and a matrix when it is two-dimensional. Since there are many variables that affect the stock price, our model will be multidimensional. In the stock price prediction, each row in the dataset will be the daily datas belong to that company, each column will be the feature. As the number of feature changes, the success of the model may also change.

#### 2.2.1.1 Linear Regression (Ordinary Least Squares)

Our model will probably not be linear as there are multiple inputs that affect the stock price. Linear models don't work well also with image recognition problems. Therefore, while describing the linear regression, it is possible to proceed through the income prediction problem. Let the person's income be predicted according to the level of education. Here, the time spent in education will be our independent variable, and the income of the person will be the dependent variable that is tried to be predicted. The ordinary least squares method aims to learn a linear model to predict a new $y$ from an unprecedented $\boldsymbol{x}$ with as few errors as possible. Linear regression is a parametric method that estimates $\boldsymbol{y}$ given an $\boldsymbol{x}$. The model will actually be a linear function of $\boldsymbol{x}$.

$$\hat{y} = \beta_0 + \beta_1 * x + \epsilon$$

Here it is clearly seen that there is a linear relationship between $\boldsymbol{x}$ and $\boldsymbol{y}$. Each unit increase or decrease in $\boldsymbol{x}$ causes a constant increase or decrease in y. Where $\beta_0$ is the point where the line intersects $\boldsymbol{y}$ and $\beta_1$ is the slope of the line. $\beta_1$, that is, the slope determines how much the income changes when the education year changes. In linear regression, the aim is to find model parameters $\beta_0$ and $\beta_1$ that minimize the error in the prediction. In order to find the optimum parameters, first of all, it is necessary to define a loss function that calculates the error of the model. Then the model is made as accurate as possible by finding the parameters that minimize the loss. Since it is the only independent variable, it is possible to show the model with a two-dimensional graphic. But in real world problems it is difficult to show more than 3 dimensions since there are usually more than 2 attributes.

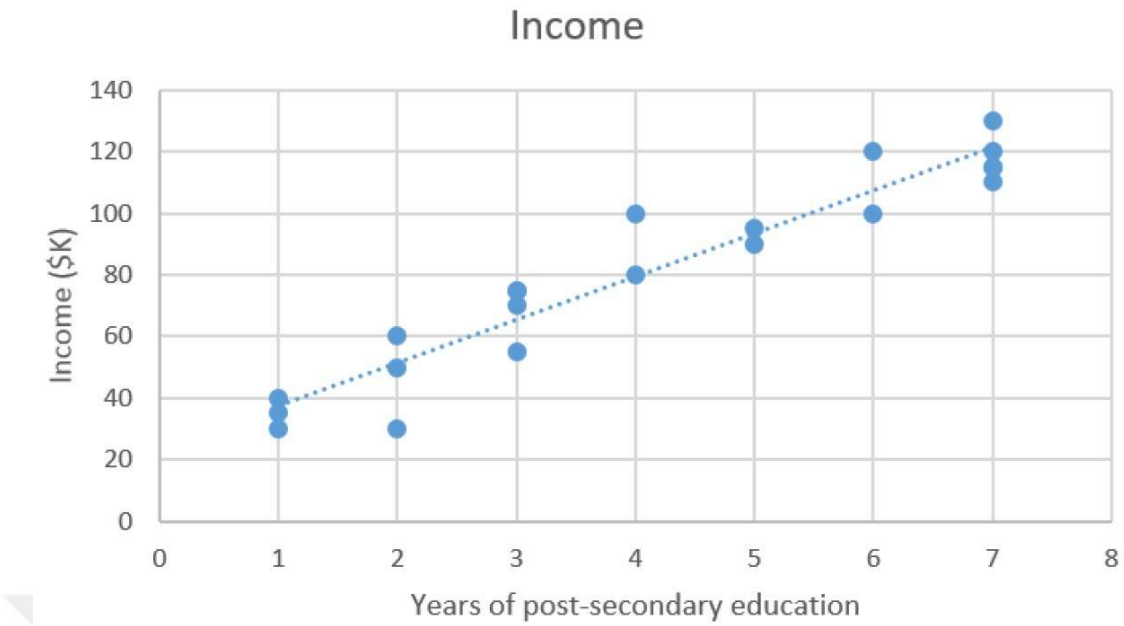



**Figure 2.1** Linear Regression Model

We find the loss function when we take and add the distances between each real data point y and the data point ŷ in the model we find. Since distance can never be negative, it would be wise to take the square or absolute value of the distances and add them. The difference between two points could be negative if we didn't take the square or absolute value of the distances. When negative values and positive values reset each other, this would cause us to calculate the error incorrectly and lead us to wrong results.

$$\text{Loss} = \frac{\sum_1^n((\beta_1 x_i + B_0) - y_i)^2}{2*n}$$

where n is number of observations. $(\beta_1 x_i + B_0)$ is predicted dependent value. $y_i$ is real targed variable. To find the optimum beta parameters that minimize the loss function, a closed-form solution can be made using calculus. But if the complexity of the loss function increases, it may become impossible to find a closed-form solution with calculus. Therefore, an iterative approach called gradient descent is used to minimize the complex loss function [14].

#### 2.2.1.2 Regression as Parameter Fitting

The closed form of linear regression is $w = (A^T A)^{-1} A^T b$ . This equation creates some problems when calculating in practice. Matrix inversion can be slow for large systems. Furthermore, formulation is fragile. It is difficult to apply the linear algebraic structure here to general optimization problems.

There is a more efficient alternative way to define and solve linear regression problems. This method provides faster algorithms, more accurate numerical results and can be easily adjusted to other learning algorithms. This method turns linear regression into a parameterization problem. It uses search algorithms to find parameters closest to the true value.

In linear regression, it is tried to find the line that passes closest to the real data over all possible coefficients set. It is aimed to find the line y = f(x) that minimizes the sum of errors at all training points. It's aimed to find w coefficient vector minimize $\sum_{i=1}^{n}(y_{i-} f(x_i))^2$ cost function where $f(x) = w_0 + \sum_{i=1}^{m-1} w_i x_i$.

To show it more concretely, y can be modeled as a linear function of x as $y = w_0 + w_1 x$. It is taken the squared sum of the distance between the real data points and the found line.

Every possible pair ($w_0$, $w_1$) will form a line. But here it is aimed to find the pair ($w_0$, $w_1$) that minimizes the loss function $J(w_0, w_1)$, where

$$J(w_0, w_1) = \frac{1}{2n} \sum_{i=1}^{n}(y_i - f(x_i))^2$$

$$= \frac{1}{2n} \sum_{i=1}^{n}(y_i - (w_0 + w_1 x_i))^2$$

In this equation, the coefficient 1/(2n) is put at the beginning of the equation for technical reasons. This move does not affect the optimization results. Since 1/(2n) will be the same for every ($w_0$, $w_1$) pair, it has no effect on parameter selection.

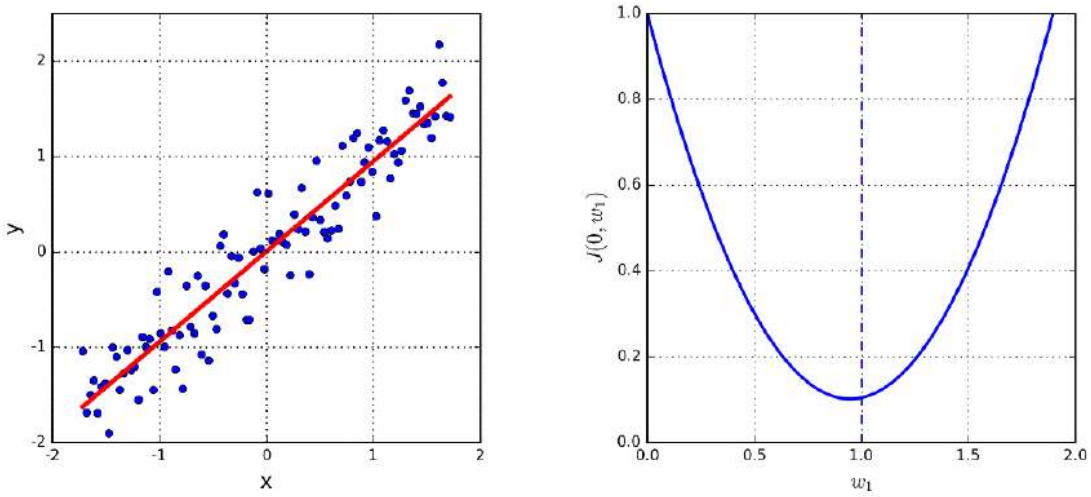



**Figure 2.2** Linear Model and Loss Function

The optimum regression line $y = w_1x$ (left) is found by defining $w_1$, which is defined by the minimum of a convex function and minimizes the error.

the whole point is to get the parameters $w_0$ and $w_1$ right in the equation $y = w_0 + w_1x_i$ . A random set of ($w_0$, $w_1$) pairs can be tried to find the most correct ($w_0$, $w_1$) pair and the error can be kept in $J$ ($w_0$, $w_1$). But it is not possible to find the best solution by trying randomly. In order to make a more systematic search, it is necessary to use a feature hidden in the loss function.

#### 2.2.1.3 Convex Parameter Spaces

$J$ ($w_0$, $w_1$) loss function defines a surface in ($w_0$, $w_1$) space. The aim is to find the minimum z-value where $z = J$ ($w_0$, $w_1$).

If $w_0$ =0 in $y = w_0 + w_1x$ equation, in this case, it is only the coefficient $w_1$ that needs to be found. The coefficient $w_1$ is actually equal to the slope of the straight line in y= $w_0 + w_1x$ equation and the regression line passes through the starting point. Some slopes will cross more closely to the actual data points in figure 2.2 (left).

Figure 2.2 (on the right) shows the relationship of the error with $w_1$. The loss function creates a parabola. At the bottom of the parabola is a single minimum value. The x value of this minimum point creates the most accurate slope $w_1$.

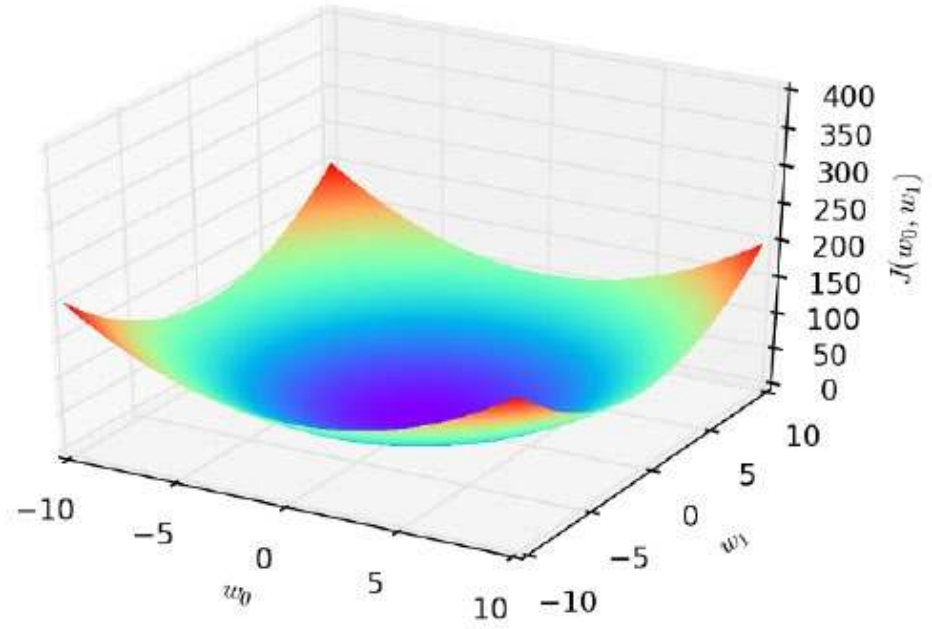



**Figure 2.3** Linear regression defines a convex parameter space.

In linear regression, each point represents a possible line. The minimum point defines the best line. Figure 2.3 shows the surface formed in the regression problem in dimension ($w_0$, $w_1$). The loss function $J$ ($w_0$, $w_1$). has a single smallest z-value for the two optimal parameters of the surface. The loss function always remains convex for any number of dimensions regression problems.

Whether a function is convex, or concave can be understood by looking at the derivatives of that function. At the point where the first derivative is equal to zero, it has a local maximum or local minimum. According to the sign of the second derivative, it can be understood that the value at that point is the maximum or minimum point of the function.

#### 2.2.1.4 Gradient Descent Search

In order to find the minimum of a convex function simply, it is necessary to start from a random point and proceed step by step in the down direction. It is not possible to advance beyond the global minimum point. The most suitable ($w_0$, $w_1$) parameters of the regression line are at this point.

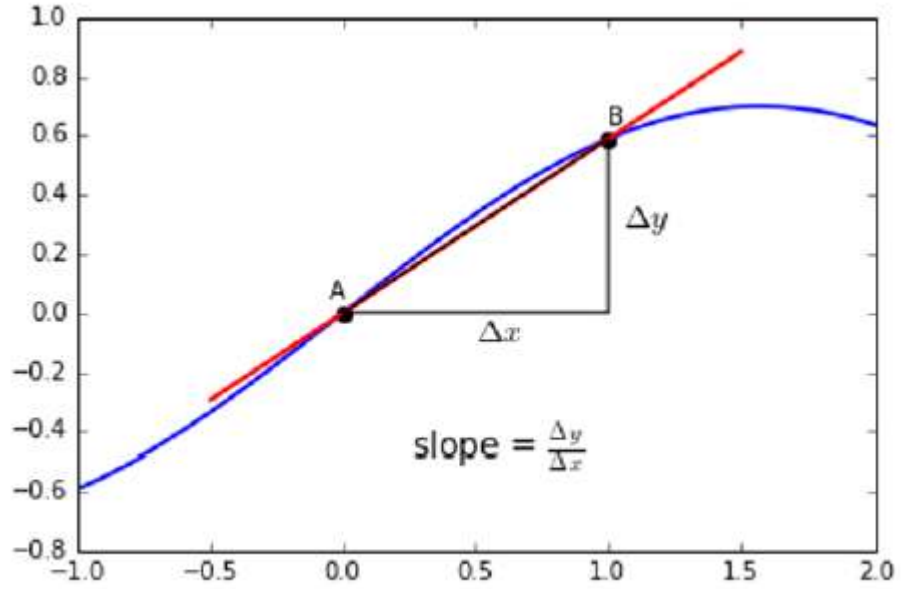



**Figure 2.4** Derivative at a point.

To find the global minimum point, one must go down from the starting point. There is a method to determine the down direction. Let $w_0 = 0$ since it is easier to understand the univariate case. In this case, the slope $w_1$ must be found. Let the current slope be $x_0$. In these conditions, it is only possible to move to the right and left. It starts by going one small step in each direction of $x_0 - \epsilon$ and $x_0 + \epsilon$.
If $J(0, x_0 - \epsilon) < J(0, x_0)$, than to go down you have to move left. If $J(0, x_0 + \epsilon) < J(0, x_0)$, than to go down you have to move right. If both of these conditions are not provided, J cannot be lowered further, it means the minimum point has been reached. If at a point the slope is positive, the minimum point is to the left of the curve, if the slope is negative, the minimum point is to the right of the curve. The magnitude of the slope is proportional to the step difference between

J(0, $x_0 - \epsilon$) and J(0, $x_0$).

In a multidimensional regression problem, it is possible to move in a wider range of directions. Multiple sizes can be cut with diagonal movements. When working with more than one dimension, it is necessary to calculate the partial derivative of the loss function for each dimension:

Following pseudocode represent regression gradient descent search in two dimensions. Here *i* variable is iteration number of the computation.

Repeat until converge {

$$w_0^{i+1} \coloneqq w_0^i - \alpha \frac{\partial}{\partial w_0} J\left(w_0^i, w_1^i\right)$$

$$w_1^{i+1} \coloneqq w_1^i - \alpha \frac{\partial}{\partial w_1} J\left(w_0^i, w_1^i\right)$$

}

Where $w_0^{i+1}$ is a coefficient of next iteration of $w_0^i$ in J = $w_0$ +$w_1x$ equation, $w_1^{i+1}$ is a coefficient of next iteration of $w_1^i$ in J = $w_0$ +$w_1x$ equation and α is learning rate.

The rate of cross-progression through dimensions can be slow. The size of the steps determines the speed. The magnitude of the partial derivatives indicates the orthogonality in that direction.

$$\frac{\partial}{\partial w_j} = \frac{2}{\partial w_j} \frac{1}{2n} \sum_{i=1}^{n} (f(x_i) - b_i)^2$$

$$\frac{\partial}{\partial w_j} = \frac{1}{2n} \sum_{i=1}^{n} (w_0 + (w_1 x_i) - b_i)^2$$

#### 2.2.1.5 Right Learning Rate

The derivative of the loss function determines a correct direction for finding the minimum error parameters in the regression problem. But it does not determine how far to go in that direction. The gradient descent algorithm works step by step. Finds the right direction, takes a

step and repeats until it reaches the target. The size of the steps is called the learning rate. It determines the speed of finding the minimums. Taking very small steps will prolong the process of finding the minimum as shown figure 2.5(left). Taking steps that are too big can cause the minimum point to be missed. For this reason, an optimum value for the step size should be determined.

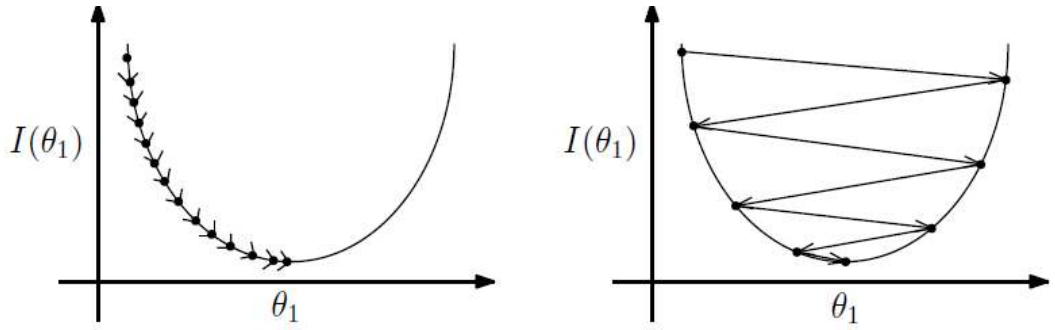



**Figure 2.5** Learning rate.

In principle, the search starts with a great learning rate at the beginning. The closer you get to the target, the lower the learning rate. When progress through the optimization is slow, the step size can be multiplied by a scalar (for example 3). When it is understood that the value of J(w) is increasing, it means that the minimum point has been exceeded. In this case, the learning rate should be reduced. For example, it can be multiplied by 1/3.

There may be many local minimum points when the loss function is not convex as in figure 2.6. The gradient descent search also finds local minimums for non-convex surfaces. However, there is no guarantee that this minimum is a global minimum.

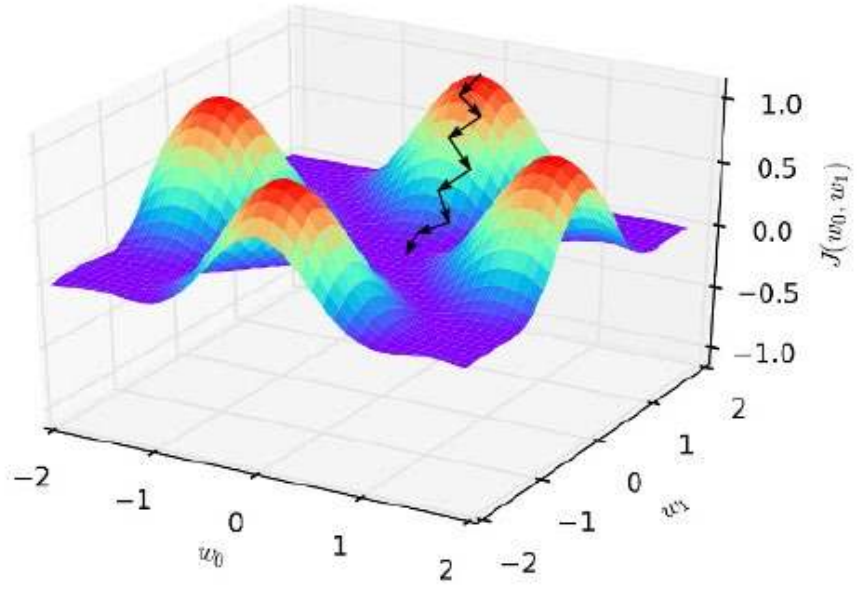



**Figure 2.6** Gradient descent search calculates local minimum point for non-convex surfaces. But it may not be global optimum solution.

Although the gradient descent search does not guarantee an optimal minimum for the non-convex loss function, in practice there is a way to find the global minimum. For this, the best local minimum can be decided after starting repeatedly from different starting points and finding all local minimums.

#### 2.2.1.6 Stochastic Gradient Descent

The algebraic expression of the loss function is actually quite expensive.

$$\frac{\partial}{\partial w_j} = \frac{2}{\partial w_j}\frac{1}{2n}\sum_{i=1}^{n}(f(x_i) - b_i)^2$$

$$= \frac{2}{\partial w_j}\frac{1}{2n}\sum_{i=1}^{n}(w_0 + (w_1 x_i) - b_i)^2$$

It is necessary to calculate the best change direction and speed of each j-dimension insert at all n training points. Evaluation of partial derivatives of all dimensions for each step takes a linear time relative to the number of samples. This

process takes long time. This method can instead be averaged using a small number of samples to estimate the derivative.

Stochastic gradient descent is an optimization method for estimating the derivative at the current position with a random sample of a small set of training data. The smaller the data size used, the faster the evaluation. Optimizing learning speed and data size provides very fast optimization for convex functions. [15]

### 2.2.3 Classification

There is a relationship between classification and clustering problems. The clustering problem is to identify groups of similar data points. It is based on learning the structure of an already separated dataset called a category or class. These categories are learned with a model. This model is used to predict the class labels of the unseen data set for unknown labels. The input to the classification problem is a data set that has been prior separated into different classes. These data are actually training data. The batch identifiers of these classes are the class labels. The learned model is called the training model. The entire set of unseen data to be classified is called test data. All the unseen data points to be classified are called test data. The algorithm that generates the training model can also be called the learner.

The classification is supervised learning because a sample dataset is used to learn the structure of batches. Batches learned by a classification model are usually related to the resemblance structure of the feature variables. Sample training data guides how groups are defined in classification. Given a set of training data set, each combined with a class label, the class of the unseen test instance is determined.

Generally, the classification problem has two steps: training and testing.

**Training stage:**

At this stage, a training model is created using the training data. This can actually be thought of intuitively as a mathematical summary of the labeled groups in the training dataset.

**Testing stage:**

At this stage, the training model is used to identify the class of unseen test samples.

The classification problem captures a grouping concept over a sample data set. So, the classification problem is stronger than the clustering problem. Such a technique can be applied to a wide variety of problems where groups are defined according to external application-specific features. Some examples are as follows:

Customer target marketing: In this case groups or tags refer to users who have an interest in a particular product and the other group is users who have no interest in that product. Usually there are training examples of previous purchasing behaviors in the dataset. These training samples are used to learn whether a customer with a known demographic, but unknown purchasing behavior will buy the specified product.

Medical disease management: In last years, data mining methods have been widely used in the diagnosis of medical diseases. Features are determined from the patient's medical tests and treatments and class labels are treatment outcomes. In this method, it is aimed to predict the treatment results with the model built on the features.

Document categorization and filtering: In applications such as the news service, documents must be classified in real time. These classifications are used in web portals to organize documents under specific headings. When document examples in each topic are available, features correspond to words in the document. The class labels correspond to headings such as sports, humor, culture and politics.

Multimedia data analysis: Photograph, audio, video or other large volumes of multimedia data may need to be classified. There may be previous examples of certain activities from certain users linked to sample videos. These can be used to determine if certain videos describe certain activities. This problem can be modeled as a binary classification problem that corresponds to whether a particular activity occurs or not.

Classification applications vary in terms of learning ability based on examples. The training dataset is assumed to have n data points and D features or dimensions. Each data point in D is combined with a label drawn from 1 to k. A training model is created using D to predict the label of unknown test samples. The output of a classification algorithm can be of two types, label prediction and numerical score. In label prediction, one label is predicted for each test sample. In numerical score, for example, a student assigns a score to each sample-label pair that measures his or her propensity for a particular class. This score can be converted into a label estimate using the maximum value of the score in different classes. The use of scores has the advantage of comparing and ranking different test samples according to their tendency to belong to a particular class. Such points are useful when one of the classes is too underrated. Through to a numerical score, the top-ranked candidates for that class can be determined.

There is an important distinction in the design process of these two models. In the label prediction model, the training model does not need to calculate the trend with respect to different test samples. The model only needs to consider the relative trend towards different labels for a given sample. The numerical score model also needs to normalize the classification scores across different test samples in order to be contrasted meaningfully for ranking.

Sometimes the performance of the classification model may be poor if the training dataset is small. In similar situations, the model may identify specific properties of the training data set. That is, such models can accurately predict the labels of the samples used to create them. But they can't perform well on unseen test samples. This condition is called overfitting.

Major models designed for data classification are decision trees, rule-based classifiers, instance-based classifiers, probabilistic models, neural networks, support vector machines [16].

### 2.2.4 Performance Metrics in Regression Models

Regression analysis has an important place in supervised machine learning. There is regression analysis in a large number of machine learning. There is no single agreed-upon metric to evaluate the results of the regression. In most of the studies, mean square error (MSE), rooted mean square error (RMSE), mean absolute error (MAE) are used. These ratios are useful but have a common disadvantage. Their values range from zero to infinity. A single value of these cannot adequately evaluate regression performance.

The mean abolute error (MAE) calculates the difference between the actual value and the line that best fits the data, that is, the predicted value. Because MAE is easy to interpret, it is used in most regression problems.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

The mean square error (MSE) is the sum of the squares of the difference between the true value and the predicted value. It is a slightly modified version of absolute error. Squared to prevent a negative value from overtaking positives. Since the MSE curve is differentiable, MSE can be used as a loss function. If there are very contradictory values from the data set, that is, very large and very small values, MSE is large. Therefore, it may not be reliable to calculate MSE for a data set with many outliers.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

The coefficient of determination is a statistical measure that represents the rate of variance in the dependent variable that can be estimated from the independent variables shown as R-square in statistics. When the coefficient of determination is correctly predicted, it is more predictive than the mean error metrics. Therefore, it is

recommended to use R-square in evaluating regression analysis in any scientific field.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

The disadvantage of $R^2$ is that when the number of features is increased, the value of $R^2$ increases or remains constant. Sometimes success may drop when adding attributes to the system. But $R^2$ continues to increase even if success drops. This causes the system to be evaluated incorrectly. To prevent this mistake, an adjusted $R^2$ measure has been created.

As p features are added, the denominator will decrease, the value of $R^2$ will remain constant or increase slightly. When a feature that increases the success of the model is added, the value of $R^2$ will increase, 1- $R^2$ will decrease, and the denominator will increase as p increases. As a result, since the adjusted $R^2$ value will increase, the success of the model will increase, and the evaluation will be successful.

$$R_a^2 = 1 - \left(\left(\frac{n-1}{n-p-1}\right) \text{ x } (1 - R^2)\right)$$

Where n: number of data points

y: actual value

$\hat{y}$: predicted value

$\bar{y}$: mean of dependent variable

$\sum$: sum of absolute value of residual

n: number of data points

p: number of independent variables

$R_a^2$: adjusted $R^2$

[17, 19].

# 3. PREDICTING STOCK PRICE

## 3.1 Python

Python is an interpreted, modular, object-oriented, high-level programming language. Simple indentation-based syntax makes the language easy to learn. It works on almost all platforms, especially Unix, Linux, Mac, Windows. Python enables rapid application development as it has built-in data structures combined with dynamic typing and linking. Python's easy-to-learn syntax improves readability and reduces maintenance cost. The python interpreter and standard library are freely available and distributable for all major platforms. [1, 20].

### 3.1.1 Pandas

In the field of data science, libraries were needed in data preparation. Pandas is a python library consisting of many data structures and tools that can be used in many fields, especially in statistics, social sciences and finance. It provides many data processing tools. Pandas can perform powerful statistical calculations and basic visualizations. It provides structures for performing data manipulations. Pandas can work with many data formats such as excel, csv, python, numpy, sql, html. The Pandas document is very rich, but the syntax is a bit difficult to understand [21, 22, 23].

### 3.1.2 Numpy

One purpose of using Python libraries is to enable the processing of large data and to improve computational algorithms. Using these libraries creates namespaces that work with modules. Python modules include objects, functions, bundled classes, constants.

Numpy creates a multidimensional array from a given table. It works with matrices using logical, bitwise, functional operations using Python's syntax and semantics.

Numpy provides various object-oriented approaches, mathematical and logical operations using ndarray [24].

### 3.1.3 Matplotlib

Matplotlib is a python package primarily intended for the visualization of scientific, financial and engineering data, creating simple and complex graphs with a few commands. It is one of the most used data visualization libraries in python. Matplotlib generates histograms, plots, bar charts, error charts, power spectra etc. Matplotlib is divided into three sections: Pylab interface, matplotlib frontend, matplotlib backend. With Matplotlib, Postscript or PNG formatted output can be generated to be added to dynamically created web pages [25, 26].

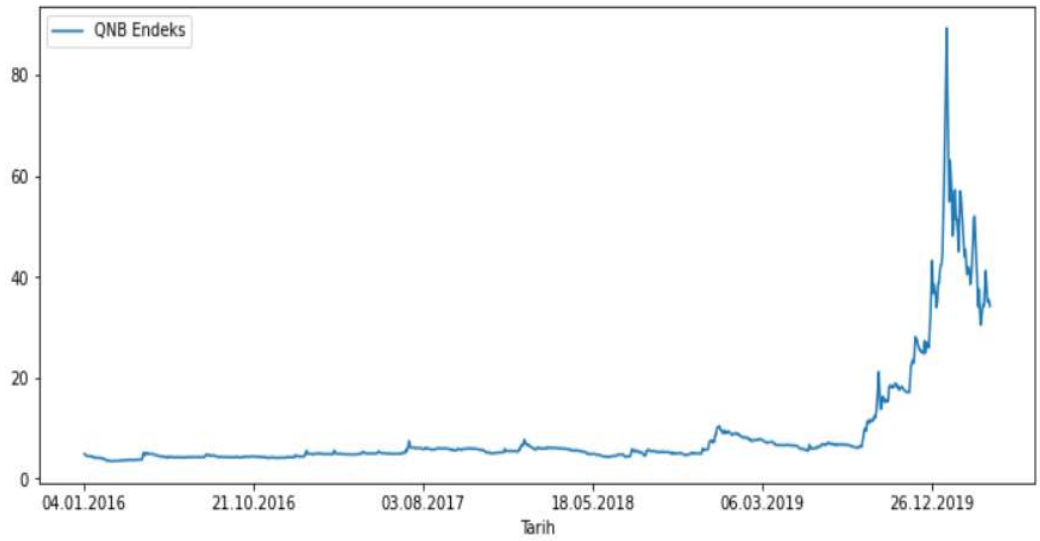



**Figure 3.1** QNBFB Endeks vs Time graph using matplotlib.

## 3.2 Obtaining Data Set

The total market value of the stocks in Borsa Istanbul is approximately 5 trillion 304 billion TL as of 2023. The total value of the top 5 stocks with the highest market value is approximately 982 billion TL. The ratio of the top 5 stocks with the largest market value to the total market value is approximately 18 % [27].

In this study, it is aimed to predict the next day closing price of the 5 stocks with the highest market value. Datas has been collected Borsa Istanbul formal website and investing.com. The daily index, opening, high, low, volume, difference, exchange rates, golden ounce price, brent oil, S&P 500, information of each stock are taken from the 'investing.com' website. Market transaction volume, Bist 100 transaction amount, Bist100 index, Bist 100 volume, Bist 100 difference information were obtained from borsaistanbul.com. Net profit for the period, resource, dividend income are collected from www.isyatirim.com.tr. Data from different sources are combined in a single table. Since financial data is not open on holidays, only working days are added to the data set. While adding internationally valid feautures such as brent oil, S&P 500, lines corresponding to holidays in Turkey were not included in the data set.

**Table 3.1** Names of stocks examined.

| STOCK CODE | COMPANY NAME |
|---|---|
| QNBFB | QNB FINANSBANK |
| ENKAI | ENKA İNŞAAT VE SANAYİ A.Ş |
| FROTO | FORD OTOMOTİV SANAYİ A.Ş |
| EREGL | EREĞLİ DEMİR VE ÇELİK FABRİKALARI T.A.Ş |
| KCHOL | KOÇ HOLDİNG A.Ş |

**Table 3.2** Explanation of Features

| Feature | Explanation |
|---|---|
| date | Indicates the date on which the relevant feature was obtained. |
| close price | Indicates the closing value of the stock on the specified date [28]. |
| opening price | Indicates the opening price of the relevant stock [28]. |
| high | Refers to the highest value of the related feature (column) during the day [28] |
| low | Refers to the lowest value of the related attribute (column) during the day [28]. |
| volume | Indicates the trading volume of the relevant stock during the day. |

| | |
|---|---|
| difference % | Indicates the change in the day-to-day price of the relevant stock as a percentage [28]. |
| BIST Stars Traded Value (TL) | Number of transactions in the star market. |
| BIST Stars Traded Volume | It is the trading volume of the market in which the shares with a market value of 300 million TL and above of the portion offered to the public in the first listing to the stock exchange are traded [29] |
| BIST 100 index | It consists of the 100 stocks traded in Borsa Istanbul with the highest market value and trading volume and is the main index of the Equity Market [30]. |
| BIST 100 volume (TL) | It is the total value of daily trading transactions in the BIST 100 [31]. |
| BIST 100 difference | It is the change of the BIST100 Index Value information announced by Borsa Istanbul at the end of the trading day according to the value of the next day [32]. |
| Dollar-TL | TL equivalent of the dollar on the relevant date. |
| Euro-TL | TL equivalent of the euro on the relevant date. |
| XAU-USD | The dollar price of an ounce of gold on the relevant date. |

Following table 3.2

| | |
|---|---|
| Brent Oil | Dollar price of brent oil on the relevant date. |
| S&P 500 | Stock market index of 500 major US stocks by Standard and Poor's [1]. |
| Euro Stoxx 50 | Stock index of 50 stocks from 11 Eurozone countries designed by Stoxx [1]. |
| Interest | The policy interest rate used by the Turkey Central Bank is the interest rate applied in one-week repo transactions. Decisions on policy rates are taken by the Monetary Policy Committee (MPC) [33]. |
| Net profit for the period | It indicates the net profit of the company in that period. |
| Resource | It is the ownership of assets that may have debts or other obligations attached to them [1]. |
| Dividend income | Dividend income is the share of the period profit obtained by a business, given to the shareholders of the company in stock or cash [18]. |

### 3.3 Data Preprocessing

There are many factors that affect the success of the machine learning algorithm. The most important of these is the representation and quality of the data set. Data must be preprocessed to improve quality. Machine learning suffers when there is too much irrelevant and redundant data. In machine learning studies, a significant amount of time is spent in data preprocessing. Data preprocessing is unavoidable as it is impossible to have a preprocessing algorithm that works on all datasets, providing reliable and effective performance. In data preprocessing, operations such as data cleaning, normalization, conversion, feature selection are performed [34, 35].

**Table 3.3** First 10 rows for QNBFB dataset

| | A | B | C | D | E | F | G | H | I | J | K | L |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Tarih | QNB Endek | Açılış | Yüksek | Düşük | Hac. | Fark % | Yıl.Paz.İş.Hacmi | Yıl.Paz.İş.Mik | Bist 100 En | Bist 100 Ha | Bist 100 Fark |
| 2 | 04.01.2016 | 4,901 | 4,990 | 5,088 | 4,883 | 416,42K | -2,29% | 2530463335.51 | 468305746 | 705,18 | 487,22M | -1,69% |
| 3 | 05.01.2016 | 4,776 | 4,910 | 4,963 | 4,759 | 246,13K | -2,55% | 3376683814.33 | 636213207 | 706,88 | 678,65M | 0,24% |
| 4 | 06.01.2016 | 4,687 | 4,759 | 4,812 | 4,661 | 201,43K | -1,86% | 4281398197.22 | 757615393 | 711,98 | 796,18M | 0,72% |
| 5 | 07.01.2016 | 4,465 | 4,652 | 4,652 | 4,367 | 362,86K | -4,74% | 3969258225.92 | 693901938 | 714,96 | 891,37M | 0,42% |
| 6 | 08.01.2016 | 4,456 | 4,501 | 4,598 | 4,429 | 184,71K | -0,20% | 3932396226.39 | 655776938 | 706,13 | 728,24M | -1,24% |
| 7 | 11.01.2016 | 4,447 | 4,438 | 4,607 | 4,394 | 337,82K | -0,20% | 3470801003.52 | 603576147 | 710,49 | 683,81M | 0,62% |
| 8 | 12.01.2016 | 4,474 | 4,465 | 4,518 | 4,412 | 139,23K | 0,61% | 4364891762.89 | 771524655 | 717,40 | 863,73M | 0,97% |
| 9 | 13.01.2016 | 4,501 | 4,474 | 4,643 | 4,465 | 444,92K | 0,60% | 4304671655.64 | 737618439 | 725,09 | 797,33M | 1,07% |
| 10 | 14.01.2016 | 4,456 | 4,474 | 4,545 | 4,438 | 111,38K | -1,00% | 4291616670.19 | 752426387 | 719,41 | 808,61M | -0,78% |

| M | N | O | P | Q | R | S | T | U | V | W |
|---|---|---|---|---|---|---|---|---|---|---|
| Dolar-TL | Euro-TL | XAU-USD | Brent Petrol | SP 500 | Euro Stox | Faiz % | Enfla | Dönem Net K | Özkaynak | Temettü |
| 2,9475 | 3,2107 | 1,074 | 37,22 | 2.012,7 | 3.164,76 | 7,5 | 9,6 | 161.970.000 | 9.166.382.000 | 2000 |
| 2,9803 | 3,2085 | 1,077 | 36,42 | 2.016,7 | 3.178,01 | 7,5 | 9,6 | 161.970.000 | 9.166.382.000 | 2000 |
| 3,0094 | 3,2369 | 1,094 | 34,23 | 1.990,3 | 3.139,32 | 7,5 | 9,6 | 161.970.000 | 9.166.382.000 | 2000 |
| 3,0221 | 3,2799 | 1,109 | 33,75 | 1.943,1 | 3.084,68 | 7,5 | 9,6 | 161.970.000 | 9.166.382.000 | 2000 |
| 2,993 | 3,3029 | 1,104 | 33,55 | 1.922,0 | 3.033,47 | 7,5 | 9,6 | 161.970.000 | 9.166.382.000 | 2000 |
| 3,0243 | 3,299 | 1,094 | 31,55 | 1.923,7 | 3.027,49 | 7,5 | 9,6 | 161.970.000 | 9.166.382.000 | 2000 |
| 3,0377 | 3,2934 | 1,087 | 30,86 | 1.938,7 | 3.064,66 | 7,5 | 9,6 | 161.970.000 | 9.166.382.000 | 2000 |
| 3,0193 | 3,2941 | 1,093 | 30,31 | 1.890,3 | 3.073,02 | 7,5 | 9,6 | 161.970.000 | 9.166.382.000 | 2000 |
| 3,0328 | 3,2852 | 1,078 | 31,03 | 1.921,8 | 3.024,00 | 7,5 | 9,6 | 161.970.000 | 9.166.382.000 | 2000 |

In the volume column (D), BIST 100 volume column(K) there are some characters different than number as “K” and “M”. K means thousand and M means million. It is necessary to convert these characters into corresponding numbers because the machine learning algorithm will not know which number the characters correspond to. % symbol is deleted from difference (G) column and BIST 100 difference (L) column.

### 3.3.1 Normalization

If the aim in machine learning is to minimize the error, applying normalization is an effective method. Different feature normalization methods can be used when the actual distribution of features is not known beforehand.

The standard scaler is a method in which the distribution approaches normal by averaging each feature and scaling its variance to 1. In the formula, the mean is subtracted from the true value and divided by the variance.

$$\hat{X}_i = \frac{x_i - \bar{X}}{\sigma}$$

Where $\hat{X}_i$ : normalization version of x

σ: standard derivation

$\bar{X}$: mean

$x_i$: each observation from a sample

The maximum absolute value scaler (MA) is a scaling method that normalizes each feature by dividing each sample by the maximum absolute value of the feature. The MA can also process sparse data and is reversible. MA is very sensitive to very large and very small values and is more suitable for normally distributed data.

$\hat{X}_i = \frac{x_i}{X_{MA}}$

Where $x_i$: each observation from a sample

$\hat{X}_i$: each normalized observation

$X_{MA}$: maximum absolute value

[36].

Min max normalization is one of the most used methods to standardize information. For each component, it converts the element's base estimate to zero, the extreme value to 1, and the other values to a decimal between 0 and 1.

$$\hat{X}_i = \frac{x_i - x_{min}}{x_{max} - x_{min}}$$

Where $x_{min}$: minimum value in X feature

$x_{max}$: maximum value in X feature

$\hat{X}_i$: scaled X [37].

In this study, the most successful result was obtained in the model established with min max normalization. Table 4.1 shows the success obtained according to the normalization method used in the data.

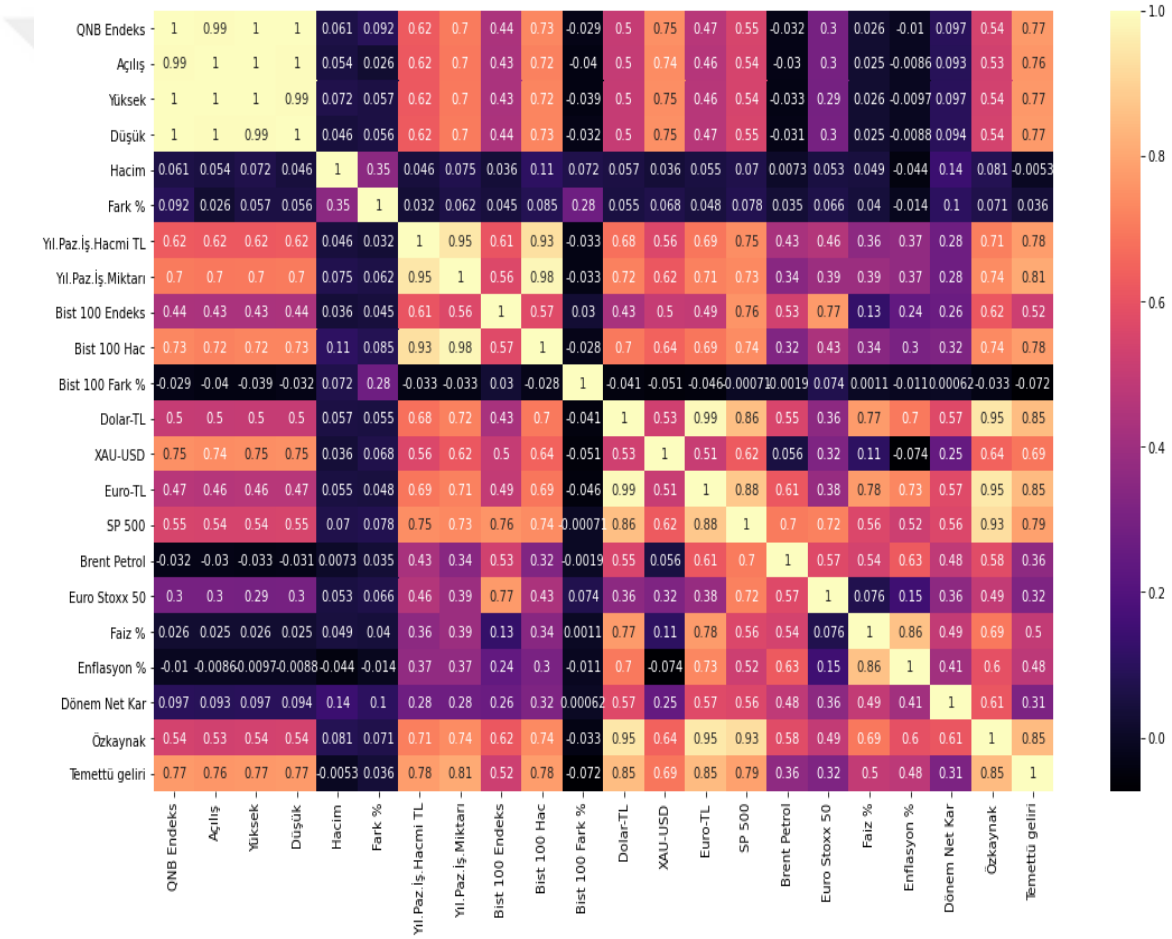
### 3.2 Analysis of Features

Seaborn is a library for making informative and statistical graphs in python. A heatmap is a graphing function in the seaborn library that uses colors to visualize the value of data.

In this study, the target is the stock price as the dependent variable, since the desired value is the stock price. The remaining features are independent variables. It is important to analyze which features affect the stock price and how much. The heatmap function in the seaborn library is used for this analysis.

-

**Table 3.4** Heatmap of the data set




According to the heat map, the factors that affect the stock price the most are the features closest to 1 in the heat map. According to the analyzed data set, the

current opening high and low values affect the stock price the most. The reason for this is that the stock price we are looking for is very close to the opening, high and

low values of the same day. This was actually something we could see before heatmap. Here there are other values close to 1. For example, with a value of 0.77 in dividend income, it is seen that it significantly affects the stock price. It is understood that the gold ounce price, which comes after that, with a value of 0.75, is also an feature that affects the stock price. Bist 100 volume and star market trading volume are among the attributes that significantly affect the dependent variable. These ratios can be used later when selecting features to increase success.

### 3.5 Predicting Stock Price Using Machine Learning Algorithms

The stages applied in machine learning are:

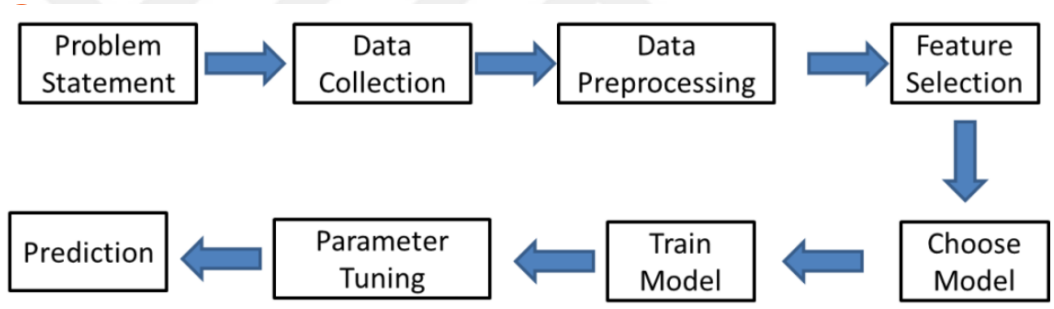



**Figure 3.2** Machine learning steps

#### 3.5.1 Creating Machine Learning Model

In the machine learning model, the data set is divided into 80% training and 20% testing. The aim was to predict the closing price of the stock for current day. Therefore, the y dependent variable is the stock price, represented by x, the independent variables being the remaining features.

In the first stage, all the features were included in the system and the model was created. No feature selection and normalization has been done. The aim here is to determine which methods and which algorithms make the best predictions. Success

found using no methods is compared to the success of predictions found using certain methods.

### 3.5.2 Cross Validation

Cross validation is a resampling method to avoid memorization and generalize the model. In cross validation, the data set is divided into subsamples. Separate training and test samples are created for each sub-sample. The training and testing part of each sub-samples are different samples. The model learns from different parts of the data in each sub-sample. The model's estimate error is calculated for all sub-samples and their average is the model's error. In this study, k-fold cross validation technique was used [38, 39].

### 3.5.3 Feature Selection

Through to feature selection methods, the computation time of machine learning algorithms can be reduced, prediction success can be increased, and data can be better understood. There are many methods for feature selection in the literature. These methods can be roughly classified as filter methods, wrapper methods, embedded and hybrid methods.

The purpose of feature selection is not to include unnecessary features that negatively affect the model and cause a decrease in success. Which feature combination will give the most successful result can be found by brute force method by trying one by one. However, this job is feasible only in models with very few features. It will be very expensive to calculate this in models with many features [40, 41].

In this study, backward elimination, and forward selection methods, which are the filter methods, were used. These methods have a stepwise approach. In the forward selection method, the most significant feature is included in the model and the algorithm is started. For this, a significance level or a p value is selected in the first step. Usually, this value is 0.05. In the second step, a regression model is created for the attribute found in the data set. For example, in this study, 18 regression models were created since 18 features were used in this method. In the second step, a regression model is created for each feature in the data set. For example, in this study, 18 regression models were created since 18 features were used in this method.

After fitting these 18 regression models, the p value is calculated for each model and the lowest p value is identified. In the third step, the feature with the lowest p value is added to all other features. In this case, n-1 (17 according to our model) binary feature clusters are formed. Models with these binary attributes are refitted and the p-value of each is recalculated. In step 4, the feature with the lowest p value is determined again. If the p value of this model with the lowest p value is less than 0.05, the feature with the lowest p value determined in the last step is added to all other models. Basically step 3 is repeated each time adding a new feature. This loop is terminated when the p value is not less than the determined significance value (0,05). Finally, the optimum feature subset is determined. The model created with this specified feature set gives the highest success.

The backward elimination algorithm can be thought of as the reverse of the forward selection algorithm. The first step of the backward elimination algorithm is the same as forward selection. A significance level is determined. In the second step, all the features available are included in the machine learning model. In the third step, the feature with the highest p value is determined. In step 4, if the feature with the highest p value determined is greater than the most significant value determined in the first step, this feature is removed from the model and step 5 is passed. If the p value of the feature with the highest p value is less than the most significant level, the algorithm is terminated. In the 5th step, the model is fitted again with the remaining features after the feature extracted in the previous step and the 3rd step is passed. This loop continues until, in step 4, the p value of the feature with the highest p value from the remaining features in the model is lower than the most significant level [42].

### 3.6 Machine Learning Algorithms Using in the Project

#### 3.6.1 Multiple Linear Regression

Multiple linear regression is a linear regression with multiple independent variables. The equation form is also similar to simple linear regression. Both types of regression are ultimately linear.

$$y_i = \beta_i + \beta_1 x_{1i} + \beta_2 x_{2i} + \cdots + \beta_n x_{ni} + e_i$$

The dependent variable y in this study is the “stock price” we aim to find. The independent variables, represented by x, are the features in the model for “qnbfb model” such as Opening, High, Low, Difference, Star Market Transaction Volume, Star Market Transaction Amount, Bist 100 Index, Bist100 Volume, Bist100 Difference %, Dollar-TL, XAU-USD, Euro- TL, SP 500, Brent Oil, Euro Stoxx 50, Interest %, Inflation %, Period Net Profit, Resources, Dividend income.

In the model established with the Enkai data set, the Opening, High and Low features are not added to the model as independent variables. The reasons for this are explained in chapter 4. Enkai Volume, BIST Stars Tradded Value(TL), BIST Stars Tradded Volume, Bist 100 index, Bist100 volume, Bist100 Difference %, Dollar-TL, XAU-USD, Euro-TL, SP 500, Brent Oil, Euro Stoxx 50, Interest %, Inflation %, Period Net Profit, Resources, Dividend Payments are included in the model created with Enkai data.

Table 3.5 Result of Multiple Linear Regression

| Test Name | Algorithm | R Square | MAE | MSE | Feature Number | Method |
|---|---|---|---|---|---|---|
| Test_enkai1 | Multiple Linear Regression | 0.95749 | 103.95 | 17892 | 18 | No normalization |
| Test_enkai2 | Multiple Linear Regression | 0.95874 | 0.15832 | 0.04125 | 18 | Standar Scaler Normalization |
| Test_enkai3 | Multiple Linear Regression | 0.94626 | 0.03950 | 0.00249 | 18 | Min-Max Normalization |

**Table 3.6** Result of real Enkai Index versus Predicted Enkai Index Using Min-Max Normalization, Multiple Linear Regression, 18 features (Test_enkai3).

y_test(actual) vs y_prediction

| | 0 | 0 |
|---|---|---|
| 0 | 4.926 | 4.914848 |
| 1 | 3.545 | 3.603694 |
| 2 | 2.371 | 2.322375 |
| 3 | 3.190 | 3.274086 |
| 4 | 3.382 | 3.302645 |
| .. | ... | ... |
| 348 | 3.708 | 3.409929 |
| 349 | 3.350 | 3.191327 |
| 350 | 3.225 | 3.536712 |
| 351 | 4.771 | 4.605244 |
| 352 | 3.141 | 3.323899 |

[353 rows x 2 columns]

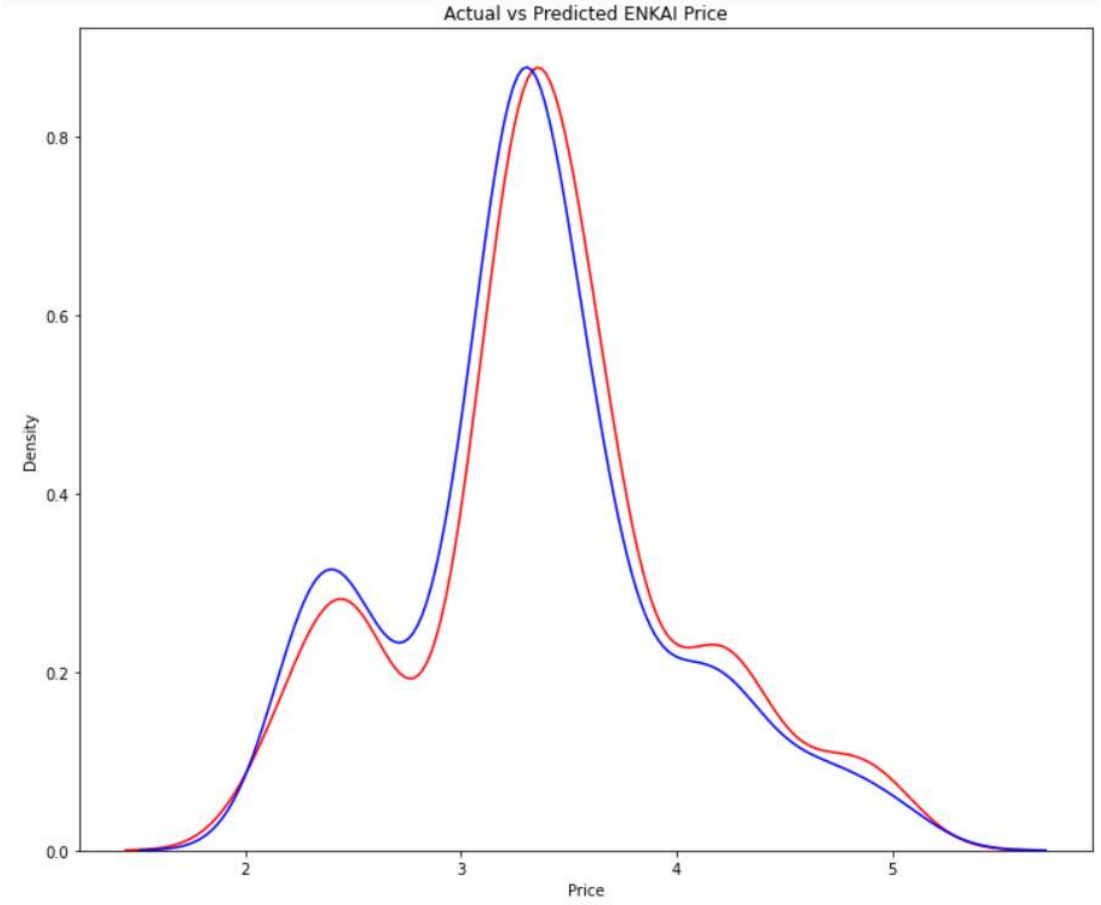


**Figure 3.2** Result of real Enkai Index(Red Line) versus Predicted Enkai Index(Blue Line) Using Min-Max Normalization, Multiple Linear Regression, 18 features (Test_enkai3).

### 3.6.2 Support Vector Regression

Support vector regression (SVR) is a supervised machine learning method used in prediction problems. Regression analysis is performed to analyze the relationship between a dependent variable and one or more independent variables. SVR formulates an optimization problem by learning a regression function to map input prediction variables to observed output values. SVR is another version of support vector machines which is classification algorithm. However SVM produces a class label i.e. a binary output. SVR is the solution to the regression problem consisting of a real-valued function prediction. The aim in SVR is to find the optimal width hyperplane containing the most appropriate line, that is, the maximum data point. SVR does not try to minimize the difference between the actual value and the predicted value as in other regression models. It tries to best fit the data within a certain threshold value. The distance between the boundary line and the hyperplane is called the threshold value [42, 43].

Support vector regression gives better results with standard scaler normalization. In this study, the data set for the support vector regression algorithm is normalized with a standard scaler. Several models were created with different kernel parameters and the results were compared.

**Table 3.5** SVR results

| Test Name | Algorithm | R Square | MAE | MSE | Feature Number | Method | Parameters (kernel) |
|---|---|---|---|---|---|---|---|
| Test5_qnbfb | Support Vector Regression | 0.91587 | 0.08878 | 0.08412 | 22 | Ss normalization | rbf |
| Test6_qnbfb | Support Vector Regression | 0.99717 | 0.03834 | 0.00282 | 22 | Ss normalization | linear |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Test7_qnbfb | Support Vector Regression | 0.98984 | 0.06627 | 0.01015 | 22 | Ss normalization | poly |

### 3.6.3 Decision Tree

Linear regression and logistic regression models cannot be successful when the relationship between the features and the dependent variable is not linear or when the features interact with each other. In such cases, tree-based models can be used. Tree models split data multiple times according to certain cutoff values in the features. Many subsets are created by splits from all data. The final, terminal subsets form leaves, and the other inner subsets form nodes. The average of the training data from this subdivided subset is used to estimate the outcome at each leaf node [15].

In this study, the best result obtained in the decision tree algorithm was with default parameters. Table 3.6 shows the two best results obtained with the decision tree.

**Table 3.6** Decision tree results

| Test Name | Algorithm | R Square | MAE | MSE | Feature Number | Normalization | Parameters |
|---|---|---|---|---|---|---|---|
| Test_qnbfb_8 | Decision Tree | 0.98571 | 0.32403 | 1.90896 | 22 | none | default |
| Test_qnbfb_9 | Decision Tree | 0.95665 | 0.57219 | 5.7922 | 22 | none | mln=20 msl=10 mf=20 |

### 3.6.4 Random Forest

The random forest algorithm is based on drawing more than one decision tree for the same dataset and using these decision trees together. Random forest algorithm can be used for classification and regression. While getting the regression result, the average of more than one separated decision tree is taken

. Since random forest is a powerful algorithm, its application field is extensive [44].

**Table 3.7** Random Forest results

| Test Name | Algorithm | R Square | MAE | MSE | Feature Number | Normalization Method | Parameters |
|---|---|---|---|---|---|---|---|
| Test_qnbfb_10 | Random Forest | 0.98418 | 0.31307 | 2.11380 | 22 | none | 10 estimators |
| Test_qnbfb_11 | Random Forest | 0.98329 | 0.29724 | 2.23181 | 22 | none | 50 estimators |
| Test_qnbfb_12 | Random Forest | 0.98346 | 0.30236 | 2.21020 | 22 | none | 100 estimators |
| Test_qnbfb_14 | Random Forest | 0.98239 | 0.30224 | 2.35262 | 22 | none | 300 estimators |
| Test_qnbfb_15 | Random Forest | 0.98944 | 0.03195 | 0.01055 | 22 | Standard scaler | 10 estimators |
| Test_qnbfb_16 | Random Forest | 0.98955 | 0.02891 | 0.01044 | 22 | Standard scaler | 50 estimators |
| Test_qnbfb_17 | Random Forest | 0.98970 | 0.02713 | 0.01029 | 22 | Standard scaler | 300 estimators |
| Test_qnbfb_18 | Random Forest | 0.98962 | 0.02714 | 0.01037 | 22 | Standard scaler | 500 estimators |
| Test_qnbfb_19 | Random Forest | 0.98946 | 0.02718 | 0.01053 | 22 | Standard scaler | 1000 estimators |

Table 3.7 shows random forest regression results in this study. Estimators parameter mean number of tree in the forest. When normalization was not applied, the increase in the number of estimators in the forest decreased the success of the model. However, changing the number of estimators did not cause a significant difference in success.

A different situation is observed when the data used in the random forest algorithm is normalized with the standard scaler. Normalizing the data before using this algorithm has increased success. It has been observed that the success increases when the estimator value is increased up to 300 in parameter changes. However, a slight decrease in success was observed when the estimator value was increased by more than 300. The highest success in the random forest algorithm is achieved when the data is normalized and the forest consisting of 300 estimators is used.

### 3.6.5 Bayesian Regression

Bayesian regression is a type of linear regression based on Bayes' theory. In the Bayesian approach, the uncertainty in the w vector is characterized by a probability distribution p(w). Bayes' theorem applies this distribution through observations of data points and the likelihood function of the data.

**Table 3.8** Bayesian Regression Results

| Test Name | Algorithm | R Square | MAE | MSE | Feature Number | Normalization Method |
|---|---|---|---|---|---|---|
| Test_qnbfb20 | Bayesian Ridge | 0.99856 | 0.18214 | 0.19126 | 22 | none |
| Test_qnbfb21 | Bayesian Ridge | 0.99673 | 0.00370 | 0.00005 | 22 | Min-max normalization |

As can be seen from Table 3.8, applying min-max normalization in the bayesian regression algorithm has reduced the error considerably.

### 3.6.6 Artificial Neural Networks

A neural network is an oriented structure that connects a simple input layer called a neuron to the output layer with weighted connections to larger structures. A neuron is connected with n input channels, each expressed in synaptic weight $w_i$. Each input from the neuron is multiplied by its weight and they are summed. An optional bias might be added to this sum. The summed result is then put into an activation(threshold) function. This function can be sigmoid, hyperbolic tangent,

ReLU or any other function. The input produces an output after filtering it with the activation function.

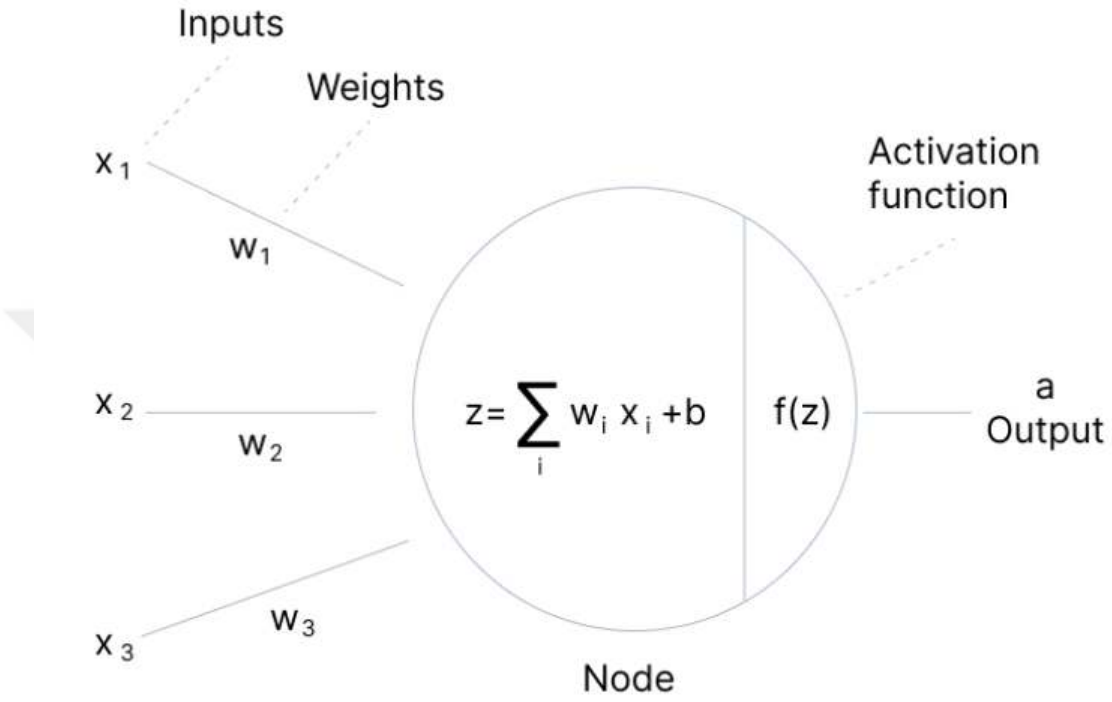



**Figure 3.3** The structure of a simple neural network

A neural network can be single-layer or multi-layered. Neural networks used in practice are often multi-layered. In multilayer neural networks, there are intermediate layers called hidden layers between input and output neurons. When there is no connection between neurons in the same layer, a neuron in a layer is connected to all neurons in the adjacent layer with a weight value. Following figure shows multi layer perceptron. Here, there are n dimensional input and k dimensional outputs. $w_{ij}$ represents weights between input layer and hidden layer. $i$ equals number of input features. $j$ represents hidden layer number. $h_{jk}$ represents weights between hidden layer and output layer. Here, the output of one layer becomes the input of the next layer.

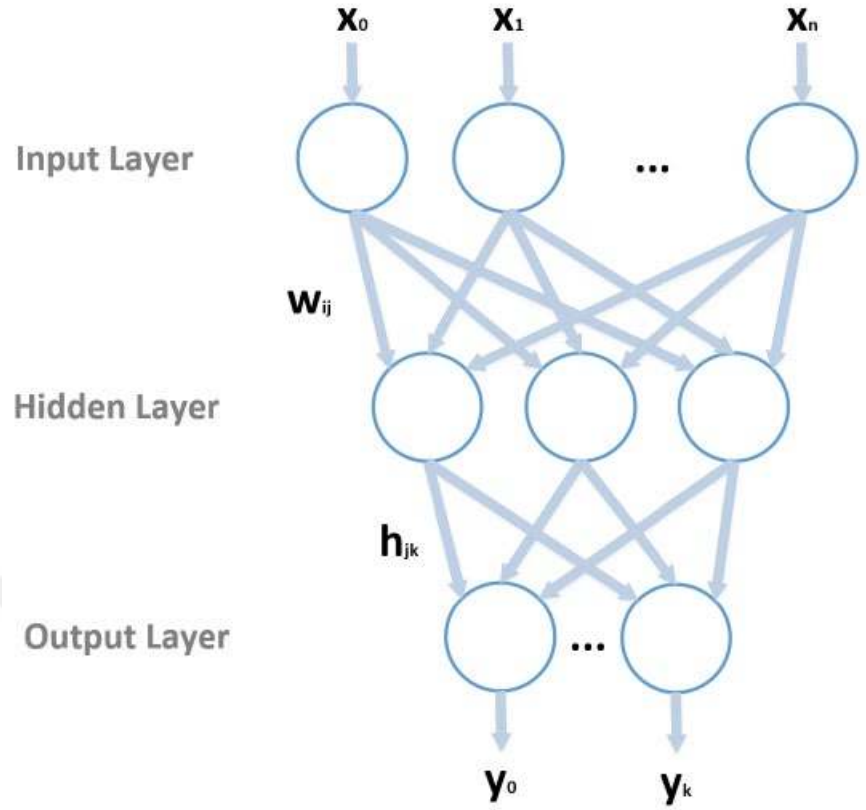



**Figure 3.4** Multi Layer Perceptron

Here, the output of one layer becomes the input of the next layer. z represents hidden layer nodes. y represents output layer nodes. $f_a$ is an activation function.

$$z_j^{input} = w_{0j}x_0 + w_{1j}\,x_1 + \ldots + w_{nj}\,x_n = \sum_i^n \left(w_{ij}x_i\right) \quad \ldots\ldots(1)$$

$$z_j^{output} = f_a^{hidden}(z_j^{input} + b_j^{hiddden}) \quad \ldots\ldots\ldots\ldots (2)$$

by the same method, equality is obtained as follows:

$$y_k^{input} = h_{0k}z_0^{output} + h_{1k}\,z_1^{output} + \ldots + h_{pk}\,z_p^{output} = \sum_j^p \left(h_{jk}z_j\right) \ldots\ldots(3)$$

$$y_k^{output} = f_a^{output}(y_k^{input} + b_k^{output}) \ldots\ldots\ldots(4)$$

After adding the input features by multiplying them with the coefficients, these operations, which are put into the threshold function and made towards the output, are referred to as “forward propagation” in the literature. In this way, the neural network becomes nonlinear. Artificial neural networks provide modeling of nonlinear structures. Once it is determined whether each neuron passes the threshold function, an output value appears after forward propagation is complete. This output value is actually the estimated value. A loss function is created by calculating the difference between the actual value and the predicted value.

$$L = \frac{1}{2} \sum_n \left\| y_n^{target} - y_n^{predicted} \right\|^2 \quad \text{.....(5)}$$

The aim was always to minimize this difference. The most common way to reduce this difference is the back-propagation algorithm. These weights are updated at each step using the back propagation algorithm, and the difference between the actual value and the predicted value decreases at each step.

Initially, it was assumed that there were n attributes as input. From the equations 3,4 and 5 above, the following equation can be obtained:

$$L = \frac{1}{2} \sum_n \sum_k \left( f_a^{output} \left( \sum_j h_{jk} z_j^{output} \right) - y_k^{target} \right)^2 = \frac{1}{2} \sum_n \sum_k \delta_k^{\,2}$$

To find the minimum difference, the derivative of the loss function is calculated. Since there are nested functions, the chain rule is applied here when making derivatives. These functions depend on weights and biases. However, deviations can be neglected for simple syntax.

$$\frac{\partial L}{\partial h_{jk}} = \sum_n \delta_k \cdot \frac{\partial f_a^{output}}{\partial y_k^{input}} \cdot \frac{\partial y_k^{input}}{\partial h_{jk}} = \sum_n \delta_k \cdot \frac{\partial f_a^{output}}{\partial y_k^{input}} \cdot z_j^{output}$$

$$= \sum_n \propto_k z_j^{output}$$

Similarly, the derivative of the loss function to the coefficient w is obtained using the chain rule:

$$\frac{\partial L}{\partial w_{ij}} = \sum_n \sum_k \delta_k . \frac{\partial f_a^{output}}{\partial y_k^{input}} . \frac{\partial y_k^{input}}{\partial z_j^{output}} . \frac{\partial z_j^{output}}{\partial z_j^{input}} . \frac{\partial z_j^{input}}{\partial w_{ij}}$$

$$= \sum_n \sum_k \propto_k . \frac{\partial y_k^{input}}{\partial z_j^{output}} . \frac{\partial z_j^{output}}{\partial z_j^{input}} . x_i$$

$$= \sum_n \sum_k \propto_k . h_{jk} . x_i . \frac{\partial z_j^{output}}{\partial z_j^{input}}$$

As it can be understood from these equations, $\propto$ is proportional to $\delta$. Real world problems can have more than one hidden layer. In this case, the above operations are repeated iteratively until the first layer. Gradient descent algorithm is used here. Therefore, the weights are updated iteratively until they converge to the real value.

$$h_{jk}^{t+1} = h_{jk}^t - \eta \frac{\partial L}{\partial h_{jk}}$$

$$w_{ij}^{t+1} = w_{ij}^t - \eta \frac{\partial L}{\partial w_{ij}}$$

where $\eta$ is learning rate [45, 46].

**Table 3.9** Artificial Neural Network Results

| Test Name | Algorithm | R Square | MAE | MSE | Feature Number | Normalization Method | Parameters |
|---|---|---|---|---|---|---|---|
| Test_qnbfb_22 | Neural Networks | 0.90933 | 0.02597 | 0.00164 | 22 | Min-Max | activation='relu' epochs = 25 test size=0.33 |
| Test_qnbfb_23 | Neural Networks | 0.99839 | 0.15525 | 0.15636 | 22 | Min-Max | activation='relu' epochs = 1000 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | | | | test size=0.33 |
| Test_qnbfb_24 | Neural Networks | 0.99824 | 0.20012 | 0.17400 | 22 | Min-Max | activation='relu' epochs = 1000 test size=0.4 |

# 4. RESULTS OF MACHINE LEARNING ALGORITHMS

This study and other studies in the literature give the result that there is a relationship between the stock market and macroeconomic variables. There are macroeconomic variables (such as interest rate, inflation, exchange rate, oil prices, gold prices) as well as intra-firm factors (such as firm performance, dividends, incomes, changes in the board of directors) that affect the stock price traded in the stock markets. In this study, internal factors and macroeconomic variables affecting stock prices were taken as features and it was investigated how much these features affect stock prices. Accordingly, machine learning models were established, and feature selections were made to increase success. Algorithm performances and used methods were compared.

The success of the test results in the studies established with the QNBFB data set was very high. The reason for this is that the target variable to be found is very close to 3 features. The QNB index values are very close to the “Opening, High, and Low” features. According to table 3.4, these 3 independent variables in the QNB index have a high correlation. Knowing the "Opening, High and Low" features of the model while training has greatly increased the success in the prediction. However, this situation is not effective and useful when predicting stock price in practice. Because in practice, the target is to predict the end-of-day closing index. The end-of-day features "low, high, open" may not be known at the beginning. Therefore, the model was created without using these 3 features for training while predicting the "Enkai" stock price in order to be more realistic in its application to daily life.

### 4.1 The Effect of Data Normalization Methods on The Result

**Table 4.1** Multiple linear regression performance according to normalization method.

| Test Name | Algorithm | R Square | MAE | MSE | Feature Number | Method |
|---|---|---|---|---|---|---|
| Test_qnbfb_1 | Multiple linear regression | 0.99856 | 0.18174 | 0.19226 | 22 | none |
| Test_qnbfb_2 | Multiple linear | 0.99859 | 0.01765 | 0.00140 | 22 | Ss normalization |

| | | | | | | |
|---|---|---|---|---|---|---|
| | regression | | | | | |
| Test_qnbfb_3 | Multiple Linear regression | 0.99680 | 0.00366 | 0.00005 | 22 | Min-max normalization |
| Test_qnbfb_4 | Multiple Linear Regression | 0.99640 | 0.00435 | 0.00006 | 22 | MaxAbs normalization |

In the study where the number of machine learning algorithms and features were kept constant, the effect of the normalization method on the model success was investigated. According to Table 4.1, the multiple linear regression algorithm gave more successful results when the data was normalized with the min max normalization method. In max abs normalization, the model success was lower compared to other normalization methods. It can be said that normalization does not increase the value of R square but reduces the error.

**Table 4.2** Comparison of the actual values with the estimated values for multiple linear regression, 22 feature, minmax normalization

| QNBFB Endeks vs Prediction Values | | |
|---|---|---|
| | 0 | 0 |
| 0 | 64.600 | 64.721419 |
| 1 | 5.560 | 5.304490 |
| 2 | 4.332 | 4.384687 |
| 3 | 9.340 | 9.232458 |
| 4 | 5.969 | 6.053082 |
| .. | ... | ... |
| 348 | 5.254 | 5.278767 |
| 349 | 4.884 | 4.860629 |
| 350 | 6.660 | 6.499725 |
| 351 | 38.600 | 39.518939 |
| 352 | 8.680 | 8.469224 |

.

### 4.2 The Effect of Parameter Changing On the Result

The choice of parameters in machine learning models can affect the model's success. Especially in artificial neural networks, the selection of the activation function, the number of epochs, the learning rate are the factors that can affect the success of the model. In this part of the study, how these parameters affect the success of the model is investigated.

Table 4.3 Effect of Parameters to Neural Network Model

| Test Name | Algorithm | R Square | MAE | MSE | Feature Number | Method | Parameters |
|---|---|---|---|---|---|---|---|
| Test_enkai4 | Neural Network | 0.95059 | 0.07713 | 0.04954 | 18 | Min-max normalization | Epoch:100 Learning rate:0.0001 Activation:relu |
| Test_enkai5 | Neural Network | 0.90907 | 0.11593 | 0.09116 | 18 | Min-max normalization | Epoch:100 Learning rate:0.0001 Activation:linear |
| Test_enkai6 | Neural Network | 0.81003 | 0.19233 | 0.19046 | 18 | Min-max normalization | Epoch:15 LR:0.0001 Act. :relu |
| Test_enkai7 | Neural Network | 0.92223 | 0.10089 | 0.07796 | 18 | Min-max | Epoch:50 LR:0.0001 Act. :relu |
| Test_enkai8 | Neural Network | 0.94165 | 0.07906 | 0.05850 | 18 | Min-max | Epoch:100 LR:0.0001 Act. :relu |
| Test_enk | Neural | 0.977 | 0.071 | 0.0224 | 18 | Min-max | Epoch:300 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| ai9 | Network | 57 | 66 | 8 | | | LR:0.0001<br>Act. :relu |
| test_enkai10 | Neural Network | 0.98144 | 0.06047 | 0.018604 | 18 | Min-max | Epoch:1000<br>LR:0.0001<br>Act. :relu |
| test_enkai11 | Neural Network | 0.99467 | 0.05395 | 0.00534 | 18 | Min-max | Epoch:2000<br>LR:0.0001<br>Act. :relu |
| test_enkai12 | Neural Network | 0.99074 | 0.05900 | 0.00927 | 18 | Min-max | Epoch:3000<br>LR:0.0001<br>Act. :relu |
| test_enkai13 | Neural Network | 0.99141 | 0.05848 | 0.00861 | 18 | Min-max | Epoch:4000<br>LR:0.0001<br>Act. :relu |
| test_enkai14 | Neural Network | 0.94953 | 0.06783 | 0.05059 | 18 | Min-max | Epoch:5000<br>LR:0.0001<br>Act. :relu |
| test_enkai15 | Neural Network | 0.92302 | 0.07361 | 0.07718 | 18 | Min-max | Epoch:7000<br>LR:0.0001<br>Act. :relu |

When the effect of the activation function in the artificial neural network model on the model success was examined, the most successful activation function was relu(rectified linear activation function). The second most successful result was obtained in the linear activation function. It was observed that lower success was achieved in sigmoid, tanh, softmax, softsign, softplus, selu, exponential functions, which are among the other activation functions.

Epoch is a parameter that determines how many times the neural network will run across the entire training dataset. For high success in the neural network model, it is necessary to give an optimum value to this parameter. In table 4.3, the effect of epoch number on model success is investigated. The highest success is the result obtained with the epoch number of 2000. The research was started with 15 epochs, and maximum success was achieved when the epoch number reached 2000. When the number of epochs exceeded 2000, the success start

ed to decline. For this reason, the epoch number was kept at 2000 while investigating the success of the learning rate selection on the model. Figure 4.1 shows the effect of the selected epoch number on the success of the model.

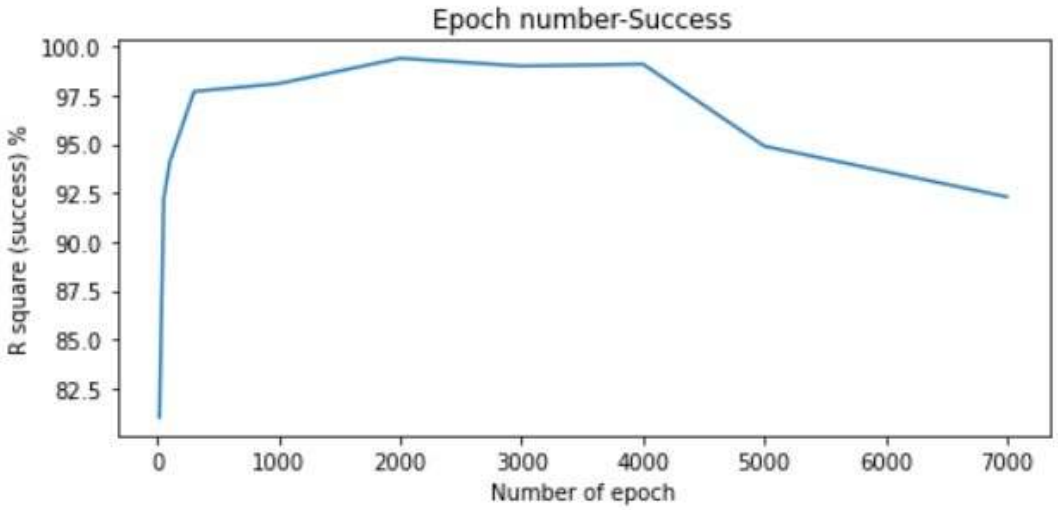



Figure 4.1 Effect of Epoch to Neural Network Model Success.

Table 4.4 Neural Network Result for 2000 epoch, 0.0001 learning rate.

```
Real Endex  vs  Prediction
          0           0
0     3.348    3.347769
1     2.751    2.839229
2     3.599    3.618610
3     2.531    2.569288
4     3.204    3.238695
..      ...         ...
349   3.507    3.523309
350   3.676    3.678339
351   3.315    3.256461
352   3.415    3.476624
353  17.770   17.839800

[354 rows x 2 columns]
```

Table 4.4 shows an prediction made by the neural network model. The prediction in the last line compares the actual value of the enkai stock on October 6, 2022 with the model predicted. The closing price of enkai stock on October 6, 2022 is 17,770. The neural network model in this study predicted this price as 17,839 with "2000" epoch number, "min-max" normalization, "relu" activation function, "0.0001" learning rate.

Table 4.5 Effect of Learning Rate to Model Success

| Test Name | Algorithm | R Square | MAE | MSE | Feature Number | Method | Parameters |
|---|---|---|---|---|---|---|---|
| test_enkai16 | Neural Network | 0.99472 | 0.05320 | 0.00529 | 18 | Min-Max | Epoch:2000 LR:0.0001 Act. :relu |
| test_enkai17 | Neural Network | 0.96653 | 0.06280 | 0.03355 | 18 | Min-Max | Epoch:2000 LR:0.001 Act. :relu |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| test_enkai18 | Neural Network | 0.7409 | 0.0806 | 0.2597 | 18 | Min-Max | Epoch:2000<br>LR:0.01<br>Act. :relu |
| test_enkai19 | Neural Network | 0.9843 | 0.0651 | 0.0157 | 18 | Min-Max | Epoch:2000<br>LR:0.00001<br>Act. :relu |
| test_enkai20 | Neural Network | 0.9752 | 0.0612 | 0.02483 | 18 | Min-Max | Epoch:2000<br>LR:<br>0.00005<br>Act. :relu |
| test_enkai21 | Neural Network | 0.89159 | 0.10148 | 0.10868 | 18 | Min-Max | 0.000001 |
| test_enkai22 | Neural Network | 0.97438 | 0.06688 | 0.02568 | 18 | Min-Max | 0.000005 |
| test_enkai23 | Neural Network | 0.96107 | 0.07221 | 0.03903 | 18 | Min-Max | 0.00002 |
| test_enkai24 | Neural Network | 0.99394 | 0.05420 | 0.00607 | 18 | Min-Max | 0.0002 |
| test_enkai25 | Neural Network | 0.98594 | 0.05751 | 0.01409 | 18 | Min-Max | 0.0004 |

In Table 4.5, there are findings investigating the relationship between learning rate and model success. Accordingly, the choice of learning rate affects success. But there is no direct or inverse proportion between these two properties. The optimum value was found to be 0.001 learning rate. At this learning rate, the model gives maximum success.

### 4.3 Prediction of Closing Price

In order to make an up-to-date forecast, it was desired to predict the closing index of the QNBFB index on October 6, 2022. For this prediction, the model of multiple linear regression created with min max normalization in this study was used. Feature values of October 6, 2022 were entered into the model as input. As a result, using multiple linear regression the closing index of QNBFB is 40,57419. The actual value is 39.0.

Table 4.6 Comparison of the actual values with the estimated values. Multiple Linear Regression, 22 features, Min Max normalization.

| QNBFB Endex vs Prediction | | |
|---|---|---|
| | 0 | 0 |
| 0 | 64.600 | 66.816663 |
| 1 | 5.560 | 5.831844 |
| 2 | 4.332 | 4.441124 |
| 3 | 9.340 | 10.019101 |
| 4 | 5.969 | 6.349404 |
| .. | ... | ... |
| 349 | 4.884 | 5.082383 |
| 350 | 6.660 | 7.201492 |
| 351 | 38.600 | 42.316190 |
| 352 | 8.680 | 9.198289 |
| 353 | 39.000 | 40.574195 |

[354 rows x 2 columns]

Table 4.7 Comparison of the actual values with the predicted values. QNBFB, Neural Networks, 22 features, Min Max normalization.

```
Real Endex  vs  Prediction
          0           0
0     4.959    4.902183
1     4.189    4.174437
2     5.790    5.643054
3     4.278    4.247966
4     4.730    4.600707
..      ...         ...
424   6.061    6.045184
425   4.430    4.345840
426   4.394    4.342442
427   4.305    4.230426
428  39.000   39.637070

[429 rows x 2 columns]
```

### 4.4 Effect of Machine Learning Algorithm to Result

In this study, when comparing algorithms, min-max normalization, which was the best normalization method before, was used, except for support vector regression and artificial neural network. In SVR and ANN, on the other hand, standard scaler normalization was used because it gave better results. Models were created with the parameters that gave the best results in parameter comparison before.

The success of the machine learning algorithms used in this study was mostly high. The reason for this may be the sufficient amount of data used in the models, using of optimum normalization, the appropriate parameter selection, the quality of the data set, and the appropriate selection of the features. In feature selection, the heatmap is basically used. Features that do not affect the closing price of the stock were removed from the model and more successful results were obtained with the remaining 18 features.

Table 4.8 Comparing the results of machine learning models.

| Test Name | Algorithm | R Square | MAE | MSE | Feature Number | Methods | Parameters |
|---|---|---|---|---|---|---|---|
| Eregli_test1 | MLR | 0.99320 | 0.01010 | 0.00035 | 18 | Min-max,cv | default |
| Eregli_test2 | SVR | 0.98759 | 0.07178 | 0.00865 | 18 | ss, cv | kernel=linear |
| Eregli_test3 | DTR | 0.99538 | 0.00679 | 0.00022 | 18 | Min-max,cv | default |
| Eregli_test4 | RFR | 0.99537 | 0.00762 | 0.00023 | 18 | Min-Max,cv | n_estimators=300 |
| Eregli_test5 | BRR | 0.99303 | 0.00993 | 0.00035 | 18 | Min-Max,cv | default |
| Eregli_test6 | ANN | 0.99875 | 0.00477 | 0.00006 | 18 | Ss, cv | epoch=1000 lr=0.0001 |

The prediction of the models created in the tables below are compared with the actual values. In the tables on the left are the predicted values obtained from the data in the model. In the table on the right, there is a summary of the closing price prediction for 104 data for the year 2021 that are not in the training dataset. The fact that the model predicts 104 values that are not included in the training data set, which is not seen at all, means that there is no overtraining in the models and the models can generalize.

Table 4.9 Multiple Linear Regression Actual vs Predicted Values

| y_test(actual) | y_prediction |
|---|---|
| 0 | 0 |
| 4.977 | 5.557179 |
| 5.100 | 5.283193 |
| 42.780 | 41.127504 |
| 7.081 | 7.344264 |
| 2.296 | 2.334036 |
| ... | ... |
| 5.829 | 5.874373 |
| 1.852 | 2.100050 |
| 4.787 | 4.624435 |
| 5.190 | 5.426974 |
| 6.604 | 6.542537 |

[236 rows x 2 columns]

| y_test(actual) | y_prediction |
|---|---|
| Eregli Endeks | 0 |
| 25.654 | 24.487722 |
| 26.513 | 25.584867 |
| 27.075 | 25.538296 |
| 26.496 | 25.610675 |
| 26.864 | 26.334213 |
| ... | ... |
| 32.420 | 34.975398 |
| 32.880 | 35.571232 |
| 34.740 | 37.096107 |
| 34.200 | 36.837934 |
| 34.240 | 36.743822 |

[104 rows x 2 columns]

Table 4.10 Support Vector Regression Actual vs Predicted Values

| y_test(actual) | y_prediction |
|---|---|
| 0 | 0 |
| 4.977 | 5.996805 |
| 5.100 | 5.994769 |
| 42.780 | 41.194491 |
| 7.081 | 7.236355 |
| 2.296 | 3.005490 |
| ... | ... |
| 5.829 | 5.535147 |
| 1.852 | 2.694491 |
| 4.787 | 5.063892 |
| 5.190 | 6.040930 |
| 6.604 | 7.525600 |

[236 rows x 2 columns]

| y_test(actual) | y_prediction |
|---|---|
| Eregli Endeks | 0 |
| 25.654 | 23.481958 |
| 26.513 | 24.399632 |
| 27.075 | 24.204015 |
| 26.496 | 24.610227 |
| 26.864 | 25.456261 |
| ... | ... |
| 32.420 | 35.294896 |
| 32.880 | 35.954107 |
| 34.740 | 37.344283 |
| 34.200 | 37.532396 |
| 34.240 | 36.641276 |

[104 rows x 2 columns]

Table 4.11 Decision Tree Regression Actual vs Predicted Values.

| y_test(actual) vs y_prediction | | |
|---|---|---|
| | 0 | 0 |
| 0 | 4.977 | 5.102 |
| 1 | 5.100 | 5.055 |
| 2 | 42.780 | 41.440 |
| 3 | 7.081 | 6.961 |
| 4 | 2.296 | 2.313 |
| .. | ... | ... |
| 231 | 5.829 | 5.537 |
| 232 | 1.852 | 1.661 |
| 233 | 4.787 | 4.752 |
| 234 | 5.190 | 5.310 |
| 235 | 6.604 | 6.800 |

[236 rows x 2 columns]

| y_test(actual) vs y_prediction | | |
|---|---|---|
| | Eregli Endeks | 0 |
| 0 | 25.654 | 23.408 |
| 1 | 26.513 | 25.513 |
| 2 | 27.075 | 24.215 |
| 3 | 26.496 | 24.215 |
| 4 | 26.864 | 24.215 |
| .. | ... | ... |
| 99 | 32.420 | 34.960 |
| 100 | 32.880 | 34.960 |
| 101 | 34.740 | 36.500 |
| 102 | 34.200 | 36.500 |
| 103 | 34.240 | 36.500 |

[104 rows x 2 columns]

Table 4.12 Random Forest Regression Actual vs Predicted Values.

| y_test(actual) vs y_prediction | | |
|---|---|---|
| | 0 | 0 |
| 0 | 4.977 | 5.299873 |
| 1 | 5.100 | 5.374297 |
| 2 | 42.780 | 40.552267 |
| 3 | 7.081 | 6.977167 |
| 4 | 2.296 | 2.348813 |
| .. | ... | ... |
| 231 | 5.829 | 5.746130 |
| 232 | 1.852 | 1.944933 |
| 233 | 4.787 | 4.371297 |
| 234 | 5.190 | 5.180907 |
| 235 | 6.604 | 6.795903 |

[236 rows x 2 columns]

| y_test(actual) vs y_prediction | | |
|---|---|---|
| | Eregli Endeks | 0 |
| 0 | 25.654 | 24.239450 |
| 1 | 26.513 | 25.376003 |
| 2 | 27.075 | 24.821627 |
| 3 | 26.496 | 25.458620 |
| 4 | 26.864 | 25.919260 |
| .. | ... | ... |
| 99 | 32.420 | 34.205427 |
| 100 | 32.880 | 34.027070 |
| 101 | 34.740 | 35.350970 |
| 102 | 34.200 | 35.613210 |
| 103 | 34.240 | 35.328943 |

[104 rows x 2 columns]

Table 4.13 Bayesian Ridge Regression Actual vs Predicted Values.

| y_test(actual) | vs | y_prediction |
|---|---|---|
| | 0 | 0 |
| 0 | 4.977 | 5.555096 |
| 1 | 5.100 | 5.354224 |
| 2 | 42.780 | 41.154862 |
| 3 | 7.081 | 7.284495 |
| 4 | 2.296 | 2.383896 |
| .. | ... | ... |
| 231 | 5.829 | 5.839429 |
| 232 | 1.852 | 2.078415 |
| 233 | 4.787 | 4.569203 |
| 234 | 5.190 | 5.502033 |
| 235 | 6.604 | 6.685541 |

[236 rows x 2 columns]

| y_test(actual) | vs | y_prediction |
|---|---|---|
| | Eregli Endeks | 0 |
| 0 | 25.654 | 24.030278 |
| 1 | 26.513 | 25.018168 |
| 2 | 27.075 | 24.962527 |
| 3 | 26.496 | 25.063810 |
| 4 | 26.864 | 25.819787 |
| .. | ... | ... |
| 99 | 32.420 | 35.484319 |
| 100 | 32.880 | 36.005699 |
| 101 | 34.740 | 37.631877 |
| 102 | 34.200 | 37.437361 |
| 103 | 34.240 | 37.058999 |

[104 rows x 2 columns]

Table 4.14 Neural Network Actual vs Predicted Values.

| y_test(actual) | vs | y_prediction |
|---|---|---|
| | 0 | 0 |
| 0 | 4.977 | 5.412799 |
| 1 | 5.100 | 5.137894 |
| 2 | 42.780 | 41.990055 |
| 3 | 7.081 | 7.081551 |
| 4 | 2.296 | 2.392915 |
| .. | ... | ... |
| 231 | 5.829 | 5.919658 |
| 232 | 1.852 | 1.896114 |
| 233 | 4.787 | 4.646408 |
| 234 | 5.190 | 5.065426 |
| 235 | 6.604 | 6.832731 |

[236 rows x 2 columns]

| y_test(actual) | vs | y_prediction |
|---|---|---|
| | Eregli Endeks | 0 |
| 0 | 25.654 | 27.431393 |
| 1 | 26.513 | 26.643353 |
| 2 | 27.075 | 27.583457 |
| 3 | 26.496 | 26.989741 |
| 4 | 26.864 | 26.284076 |
| .. | ... | ... |
| 99 | 32.420 | 34.401145 |
| 100 | 32.880 | 35.831437 |
| 101 | 34.740 | 35.985915 |
| 102 | 34.200 | 36.425303 |
| 103 | 34.240 | 35.251362 |

[104 rows x 2 columns]

The success of machine learning algorithms applied in this study was close to each other. In order to measure the success of the machine learning models, mean square error, mean absolute error and r square are used as evaluation scales. In addition, the data that is not seen in the training set was predicted and these metrics were also evaluated. With all these evaluation techniques, the ANN gave the best results. The most unsuccessful result was obtained in support vector regression.

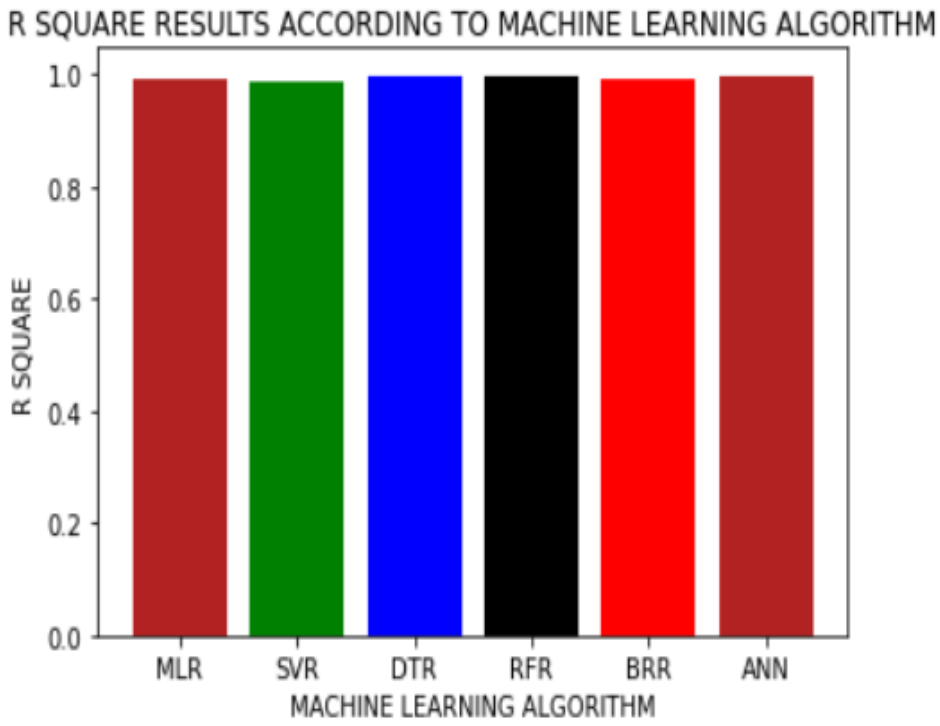



Figure 4.2 R Square results for test datas in machine learning algorithm.

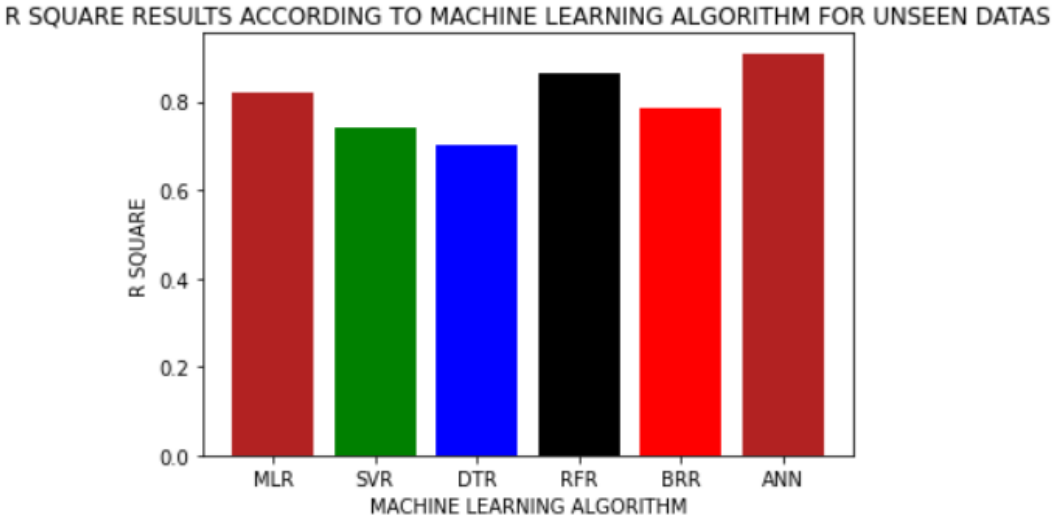



Figure 4.3 R Square results for 2021 unseen datas in machine learning algorithm.

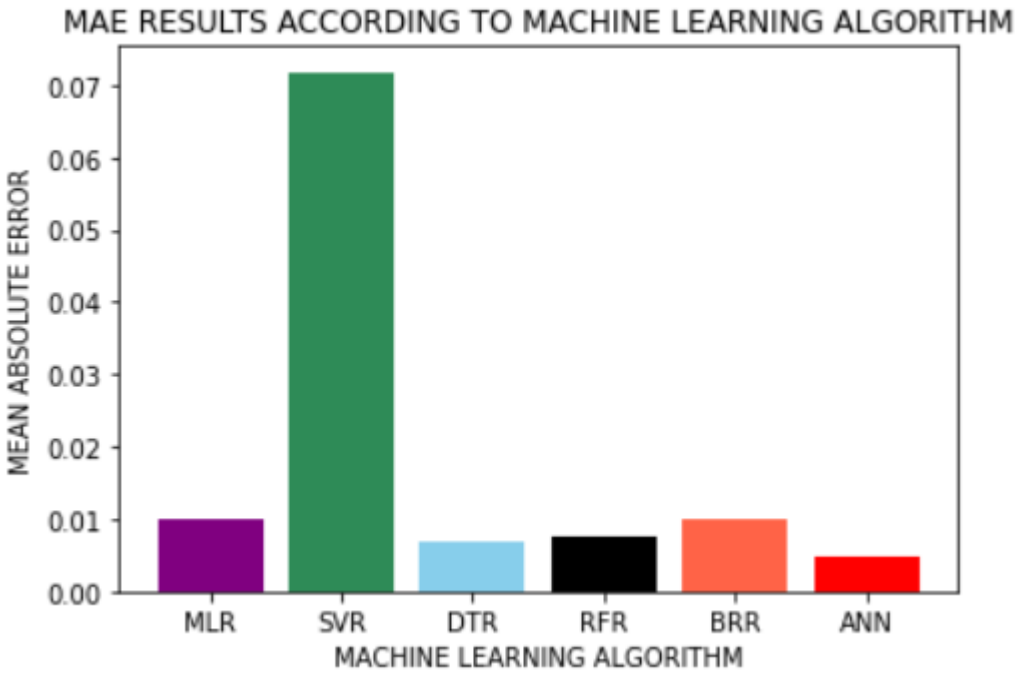



Figure 4.4 MAE results for test datas in machine learning algorithm.

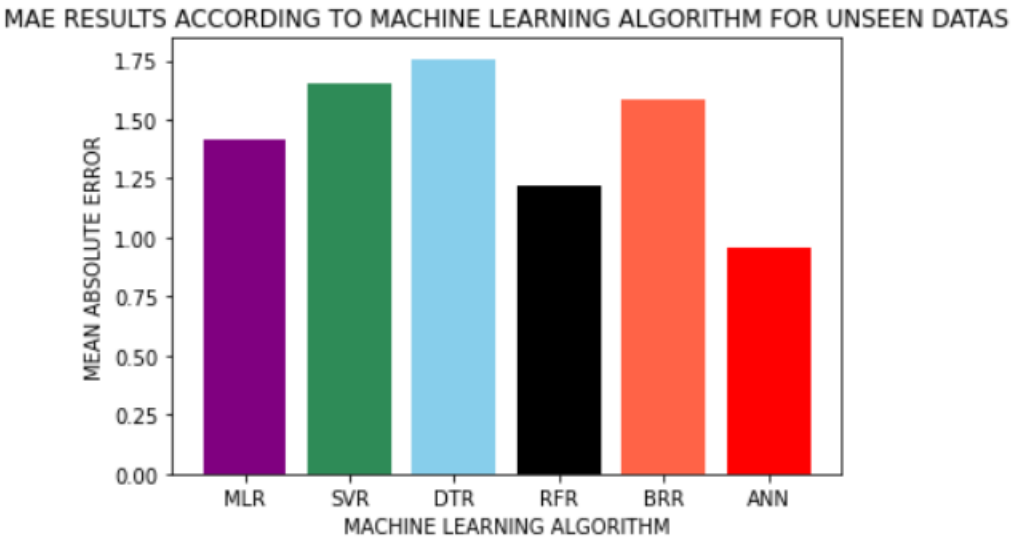



Figure 4.5 MAE results for 2021 unseen datas in machine learning algorithm.

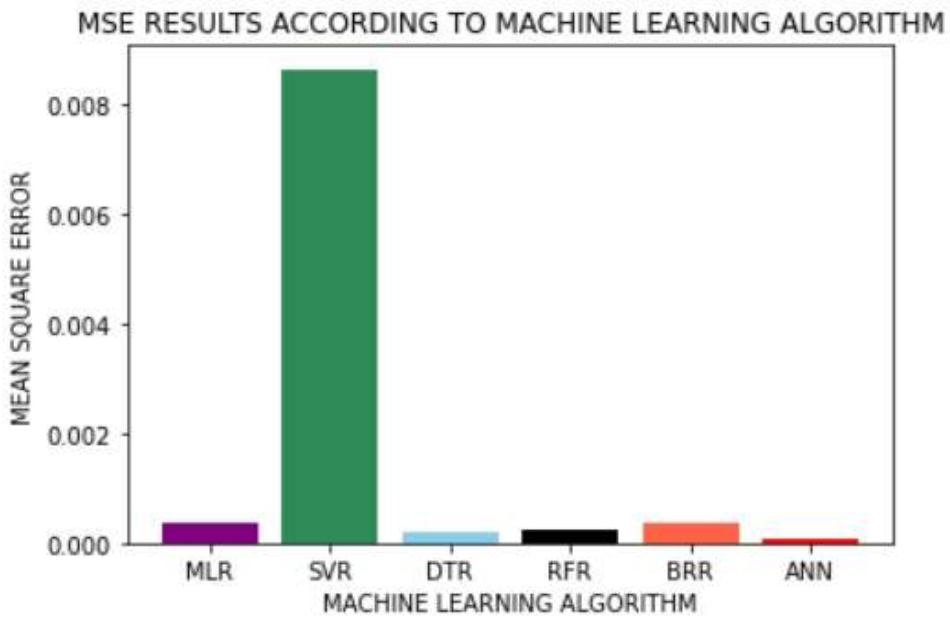



Figure 4.6 MSE results for test datas in machine learning algorithm.

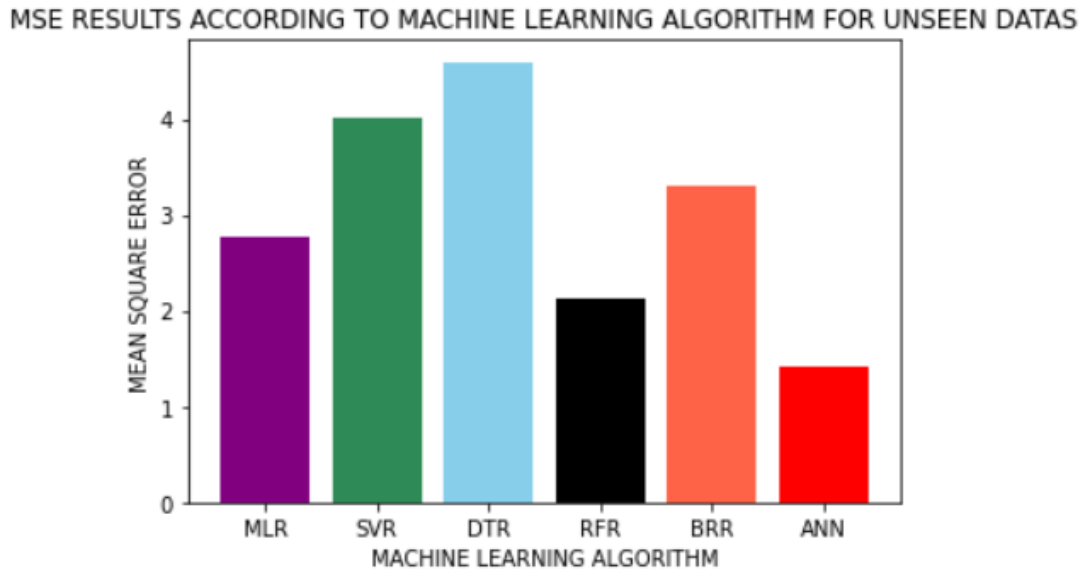



Figure 4.7 MSE results for 2021 unseen datas in machine learning algorithm.

### 4.5 Effect of Feature Selection Methods to Result

Table 4.15 Comparing machine learning methods with and without feature selection method for unseen datas.

| Test Name | Algorithm | R Square | MAE | MSE | Feature Number | Methods | Parameters |
|---|---|---|---|---|---|---|---|
| Eregli_test7 | MLR | 0.8208 | 1.4129 | 2.7662 | 18 | min-max cv | default |
| Eregli_test8 | MLR | 0.8336 | 1.3587 | 2.5685 | 17 | ffs,min-max,cv | default |
| Eregli_test9 | SVR | 0.7402 | 1.6551 | 4.0114 | 18 | ss,cv | kernel=linear |
| Eregli_test10 | SVR | 0.8492 | 1.2793 | 2.3286 | 9 | ffs,ss,cv | kernel=linear |
| Eregli_test11 | DTR | 0.7025 | 1.7568 | 4.5945 | 18 | min-max,cv | default |
| Eregli_test12 | DTR | 0.7659 | 1.4455 | 3.6146 | 12 | ffs,min-max,cv | default |
| Eregli_test13 | RFR | 0.8629 | 1.2241 | 2.1161 | 18 | min-max,cv | n_estimators=300 |
| Eregli_test14 | RFR | 0.8656 | 1.2255 | 2.0748 | 17 | bwe,min-max,cv | n_estimators=300 |
| Eregli_test15 | BRR | 0.7867 | 1.5828 | 3.2934 | 18 | min-max,cv | default |
| Eregli_test16 | BRR | 0.7868 | 1.5468 | 3.2913 | 16 | bwe,min-max,cv | dafault |
| Eregli_test17 | ANN(MLP) | 0.9089 | 0.9544 | 1.4056 | 18 | ss,cv | act=logistic,hidden layer=4 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Eregli_test 18 | ANN(MLP) | 0.9424 | 0.7649 | 0.8885 | 7 | bwe,ss, cv | act=logistic,hidden layer=4 max_iter=1000 |

As it can be understood from Table 4.12, feature selection methods increased the success in all algorithms. The most significant increase in success has been in SVR, DTR and ANN algorithms. For all algorithms, backward elimination and forward feature selection methods have been tested. The success for MLR, SVR and DTR algorithms has increased with the forward feature selection method. The success for the RFR, BRR and ANN algorithms has increased with the backward elimination method.

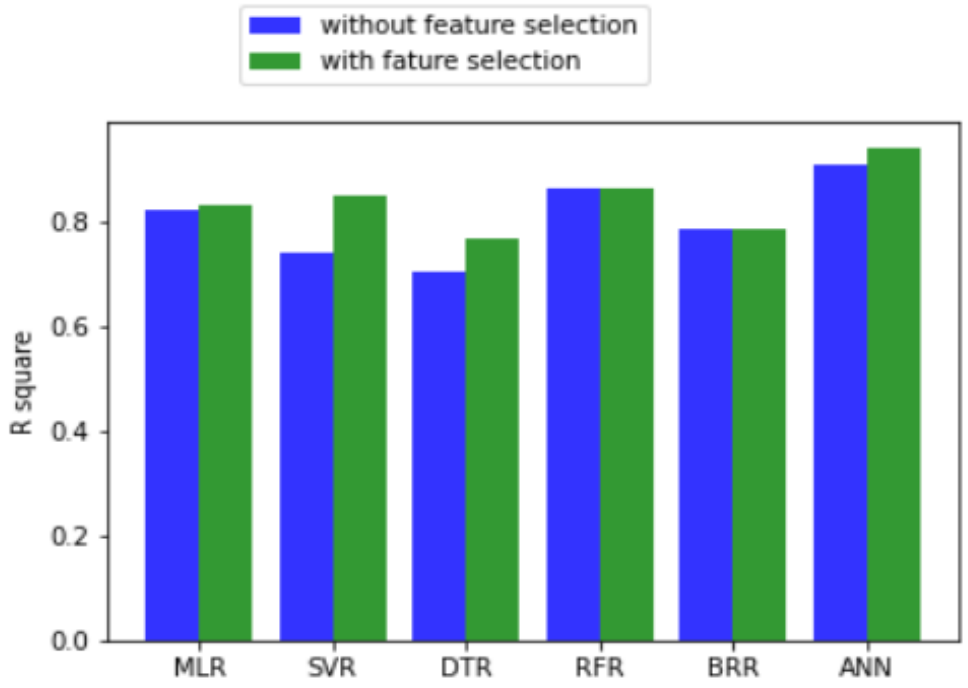



Figure 4.8 Effect of feature selection method to algorithm performances

# 5. CONCLUSION AND SUGGESTIONS

The success of the machine learning algorithms applied in this study was generally high. In order for the success to be high, enough data is used in machine learning models. In order for the data to be of good quality, empty lines were removed during the data preprocessing stage, all the features were collected numerically, and dummy variables were not included in the model. Many tests have been applied to select the parameters in the algorithms appropriately.

When normalization was applied in the data set, the success of the algorithms increased, and the amount of error decreased. MSE also decreased as a result of normalization. According to Table 4.1, MAE was 0.18174 when normalization was not applied in Test_qnfb_1 test, while MAE decreased to 0.01765 in standard normalization, and MAE decreased to 0.00366 in min-max normalization. MSE also decreased as a result of normalization. In addition, it has been observed that different normalization techniques are successful in different machine learning algorithms. While min-max normalization gave more successful results in MLR, DTR, RFR and BRR algorithms, standard scaler normalization technique gave more successful results in SVR and ANN algorithms. For this reason, while investigating the effect of parameter change, machine learning algorithm, and feature selection method on success, the normalization technique, which gives better results for each algorithm, has been applied.

The effect of machine learning algorithm on success was investigated and the most successful results were obtained when ANN was applied according to table 4.5 and figure 4.3. According to Table 4.4, the most successful algorithm, ANN, predicted the QNBFB stock dated 6 January 2022 as 39.63 with a real index value of 39.00. Since more successful results were obtained for unseen data, MLP, which is the ANN type, was applied. When the success was measured for the observed data, results were close to each other for all algorithms. While the SVR was given according to table 4.5 for the data with the most unsuccessful result among the 6 algorithms applied, it gave DTR according to figure 4.3 for the data not seen.

The effect of the parameters in the algorithms on the success of the model has also been investigated. According to Table 4.3, optimum parameters for ANN were epoch=2000, learning rate=0.001, activation function=relu. With the optimum parameters selected in this way, the actual value of the Enkai stock dated 6 January 2022 is estimated as 17,839 in the ANN model, which is 17,770 as in table 4.4. The most successful result for SVR is provided with kernel=linear parameter according to table 3.5. The most successful result for RFR was obtained when the n_estimators parameter was set to 300. For other algorithm types, the most successful results were obtained when the parameters were taken as default.

In order to increase the success of machine learning algorithms in the study, forward selection and backward elimination method, which are feature selection methods, were applied. According to Table 4.12, the feature selection methods increased the success of R square and decreased the error measures in the results obtained from testing the final stock closing price of 2021 for unseen data that is not in the training data set. For unseen data, R square success of MLR algorithm increased from 82.0% to 82.3%, MAE and MSE decreased with forward feature selection method. Similarly, applying the forward feature selection method in the SVR algorithm increased the success of R square from 74.0% to 84.9% and greatly reduced the error measures. Forward feature selection method in DTR algorithm increased the success of R square from 70.2% to 76.5% and decreased the error measures. Since the forward feature selection method did not increase the success in the RFR algorithm, the backward elimination method was applied. Although the backward elimination method did not significantly increase the performance of the RFR algorithm, it increased the success of the R square from 86.2% to 86.5% and slightly decreased the error rates. The forward feature selection method for the BRR algorithm did not increase the success, but the success increased slightly with the backward elimination method. In the MLP algorithm, which is the ANN type, the backward elimination method increased the R square success of the model from 90.8% to 94.2% and was successful by reducing the error rates.

In the study, cross validation technique was applied to prevent overfitting. In order to show that machine learning models learn without overfitting, predictions are made for the 100-days index of 2021, which is not in the training dataset of the

models. As seen in Table 4.6, Table 4.7, Table 4.8, Table 4.9, Table 4.10 and Table 4.11, prediction results close to the actual value were found in the prediction of these 100 unseen data. For this reason, it has been shown that the machine learning models applied in the study can be generalized.

In this study, machine learning models were created by taking 3 internal and 18 external features in the models with the highest number of features. The number of internal features can be increased in future work. The number of features can be increased by adding the features used in this study by performing sentiment analysis with daily data compiled from container data or other financial news sites. In this study, 5 years of data were collected. A machine learning model can be created with data from longer years. A mobile or web-based application can be made by adding an interface to the infrastructure in this study.

## REFERENCES


[1] www.wikipedia.org

[2] https://dataconomy.com/2023/01/stock-prediction-machine-learning/

[3] https://builtin.com/machine-learning/machine-learning-stock-prediction

[4] Hürer, E. Hisse Senedi Fiyatını Etkileyen Faktörler ve
İMKB Üzerine Bir Uygulama, İstanbul Universitesi, Sosyal Bilimler Enstitüsü, Uluslararası İşletmecilik, İstanbul, 1995, 208 s.(Master Thesis)

[5] Molnar C. Interpretable Machine Learning, Lulu.com, 2019, 314 s.

[6] Bi, Q., Goodman, K. E., Kaminsky, J., Lessler, J., What is machine learning? A primer for the epidemiologist, American journal of epidemiology, 2019, 188(12), 2222-2239.

[7] Balı, S., Cinel, M., Günday, A. Hisse Senedi Fiyatlarini Etkileyen Temel Makroekonomik Faktörlerin BIST 100 Endeksi'ne Etkisinin Ölçümlenmesi. Ordu Üniversitesi Sosyal Bilimler Enstitüsü Sosyal Bilimler Araştırmaları Dergisi. 2014

[8] Ghani, M., Awais, M., Muzammul, M. Stock Market Prediction Using Machine Learning(ML) Algorithms. Advances in Distributed Computing and Artificial Intelligence Journal. 2019, 4, 97-116

[9] Sarode, S., Tolani, H., Kak, P., Lifna, C. Stock Price Prediction Using Machine Learning Techniques. International Conference on Intelligent Sustainable Systems (ICISS), 21-22 February, 2019, Palladam, India.

[10] Usmani, M., Adil, S., Raza, K., Ali, S. Stock Price Prediction Using Machine Learning Techniques. 3rd International Conference On Computer And Information Sciences (ICCOINS), 15-17 August, 2016, Kuala Lumpur, Malaysia.

[11] Tipirisetty, A. Stock Price Prediction using Deep Learning. San Jose State University, Department of Computer Science, California, 2018, 54s. (Master Thesis)

[12] Singh, S. Stock Prediction using Machine Learning, California State University, Computer Science, California, 2021, 16s. (Master Thesis)

[13] Guo, Y. Stock Price Prediction Using Machine Learning, Sodertorn University, School of Social ScienceMaster, Economics, Stockolm, 2022, 34. (Master Thesis).

[14] Maini, V., Sabri S Machine Learning for Humans. (Ed: Sachin Maini), 2017, 97 s.

[15] Skinea, S. Data Science Design Manual, New York, USA, 2017, 453 s.

[16] Aggarwal, C., Data Mining, Springer International Publishing, New York, USA, 734 s.

[17] Chicco D, Warrens MJ, Jurman G. The Coefficient of Determination R-Squared Is More Informative Than SMAPE, MAE, MAPE, MSE and RMSE in Regression Analysis Evaluation. PeerJ Computer Science. 2021, 7, 1-24

[18] https://finans.mynet.com

[19] https://www.analyticsvidhya.com/

[20] www.python.org

[21] Stancin, I., Jovic, A. An Overview and Comparison of Free Python Libraries for Data Mining and Big Data Analysis. 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 20-24 May, 2019, Opatija Croatia

[22] McKinney, W. Pandas: A Foundational Python Library for Data Analysis and Statistics. Python for High Performance and Scientific Computing. 2011, 14.9, 1-9.

[23] Bernard, J. Python Recipes Handbook, Steve Anglin, Springer Science+Business Media New York, Fredericton, Canada, 2016, 123

[24] Lemenkova, P. Processing Oceanographic Data by Python Libraries Numpy, Scipy and Pandas. Aquatic Research. 2019, 2(2), 73-91.

[25] Shopbell, P., Britton, M., Erbert, R. matplotlib – A Portable Python Plotting Package. Astronomical Data Analysis Software And Systems XIV ASP Conference Series, 2005, California(Astronomical Data Analysis Software and Systems XIV, 91-95.)

[26] Ari, N., Ustazhanov, M. Matplotlib in Python. 11th International Conference on Electronics, Computer and Computation (ICECCO), 29 September, 2014, Abuja, Nigeria, 1-6.

[27] https://www.isyatirim.com.tr/

[28] Aslan, B. Derin Öğrenme ile Borsa Verileri Üzerinde Tahminleme Yapılması, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı, İzmir, 2020, 61s. (Yüksek Lisans Tezi)

[29] https://www.borsaistanbul.com/

[30] https://www.alnusyatirim.com/

[31] https://bigpara.hurriyet.com.tr/

[32] Karagöz, S. Payların Kapanış Fiyatlarının Makine Öğrenmesi Yöntemleri ile Tahmin Edilmesi, Lisansüstü Eğitim Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı, İstanbul, 2020, 118s. (Yüksek Lisans Tezi)

[33] www.tcmb.gov.tr

[34] Kotsiantis, S.B., Kanellopoulos, D., Pintelas P.E., Data Preprocessing for Supervised Leaning. International Journal of Computer Science Volume 1 Number 1, 2006, 1306-4428, 111-117

[35] Alexandropoulos, S.N., Kotsiantis S.B., Vrahatis M.N., Data Preprocessing in Predictive Data Mining, Cambridge University Press, 2019, 34 E1

ynet.com/

[36] Ferreira, P., Le. D., Zincir-Heywood N., Exploring Feature Normalization and Temporal Information for Machine Learning Based Insider Threat Detection. 15th International Conference on Network and Service Management (CNSM), 21-25 October, 2019, Halifax, NS, Canada

[37] Raju, G., Lakshmi, K., Jain, V., Kalidindi, A., Padma V., Study the Influence of Normalization/Transformation process on the Accuracy of Supervised Classification. Third International Conference on Smart Systems and Inventive Technology (ICSSIT), 20-22 August, 2020, Tirunelveli, India.

[38] King, R., Orhobor, O., Taylor, C. Cross-Validation is Safe to Use, Nature Machine Intelligence. 2021, 3, 276-276

[39] B, Daniel. Cross-Validation, Data Science Laboratory. 2019, 2, 542-545

[40] Chandrashekar, G., Sahin, F., A Survey on Feature Selection Methods. Computers & Electrical Engineering, 2014, 40, 16-28.

[41] Jović, A. and Brkić, K., Bogunović, N. A Review of Feature Selection Methods with Applications. 38th International Convention on Information and Communication Technology, 25-29 May, 2015, Opatija, Croatia.

[42] https://towardsdatascience.com/

[43] Zhang, F., O'Donnel, L. Support Vector Regression, Machine Learning Methods and Applications to Brain Disorders. 2020, 7, 123-140

[44] Liu, Y., Wang, Y., Zhang, J. New Machine Learning Algorithm: Random Forest. International Conference on Information Computing and Applications, 2012, (Information Computing and Applications, 246–252)

[45] Bonaccorso G. Machine Learning Algorithms., Packt Publishing, Birmingham, UK, 2017, 337s.

[46] Kubat, M. An Introduction to Machine Learning. Springer International Publishing, FL., USA, 2017, 348 s.

# APPENDIX

## APPENDIX A (Algorithm Results for Ford Oto Stock)

Table Appendix A.1 Algorithms evoluation for unseen 2022 datas

| Test Name | Algorithm | R Square | MAE | MSE | Feature Number | Methods | Parameters |
|---|---|---|---|---|---|---|---|
| Froto_test2 | MLR | 0.9766 | 3.0904 | 17.819 | 22 | Min-max,cv | default |
| Froto_test4 | SVR | 0.9070 | 6.2627 | 70.992 | 22 | ss, cv | kernel=linear |
| Froto_test6 | DTR | 0.8953 | 7.1477 | 79.900 | 22 | Min-max,cv | default |
| Froto_test8 | RFR | 0.9293 | 5.8875 | 53.926 | 22 | Min-Max,cv | n_estimators=300 |
| Froto_test10 | BRR | 0.9768 | 3.0946 | 17.654 | 22 | Min-Max,cv | default |
| Froto_test12 | ANN | 0.9497 | 4.6983 | 38.391 | 22 | Ss, cv | epoch=1000 lr=0.0001 |

Table Appendix A.2 actual and prediction MLR test results for Froto closing stock price.

```
y_test(actual) vs y_prediction
          0           0
0     43.58   43.051500
1     51.05   50.277017
2    527.00  531.070777
3     50.68   52.308905
4     23.20   22.734684
..      ...         ...
231   43.76   44.507911
232   24.44   24.143107
233   37.76   37.876572
234   51.79   51.488416
235   72.18   71.751861

[236 rows x 2 columns]
```

Table Appendix A.3 actual and prediction MLR results for Froto closing stock price (unseen 104 datas for the year 2022)

```
y_test(actual) vs y_prediction
     Froto Endeks           0
0          232.81  231.954833
1          235.40  234.717023
2          241.22  237.486268
3          239.28  240.206046
4          237.53  236.243460
..            ...         ...
99         272.88  279.914945
100        274.91  281.699929
101        290.24  297.849784
102        298.53  306.117892
103        295.06  302.138382

[104 rows x 2 columns]
```

Table Appendix A.4 actual and prediction SVR test results for Froto closing stock price.

| | y_test(actual) | y_prediction |
|---|---|---|
| | 0 | 0 |
| 0 | 43.58 | 51.640316 |
| 1 | 51.05 | 58.532137 |
| 2 | 527.00 | 496.391688 |
| 3 | 50.68 | 52.241457 |
| 4 | 23.20 | 22.945987 |
| .. | ... | ... |
| 231 | 43.76 | 50.621517 |
| 232 | 24.44 | 26.882427 |
| 233 | 37.76 | 44.860387 |
| 234 | 51.79 | 60.970441 |
| 235 | 72.18 | 80.712073 |

[236 rows x 2 columns]

Table Appendix A.5 actual and prediction SVR results for Froto closing stock price (unseen 104 datas for the year 2022)

| | y_test(actual) vs | y_prediction |
|---|---|---|
| | Froto Endeks | 0 |
| 0 | 232.81 | 232.238272 |
| 1 | 235.40 | 238.450959 |
| 2 | 241.22 | 240.890226 |
| 3 | 239.28 | 241.415209 |
| 4 | 237.53 | 239.401361 |
| .. | ... | ... |
| 99 | 272.88 | 280.552562 |
| 100 | 274.91 | 285.192254 |
| 101 | 290.24 | 295.428172 |
| 102 | 298.53 | 301.772666 |
| 103 | 295.06 | 298.826785 |

[104 rows x 2 columns]

Table Appendix A.6 actual and prediction DTR test results for Froto closing stock price.

```
y_test(actual) vs y_prediction
          0       0
0     43.58   43.08
1     51.05   49.78
2    527.00  514.20
3     50.68   51.01
4     23.20   23.05
..      ...     ...
231   43.76   43.58
232   24.44   24.85
233   37.76   36.63
234   51.79   52.40
235   72.18   70.89

[236 rows x 2 columns]
```

Table Appendix A.7 actual and prediction DTR results for Froto closing stock price (unseen 104 datas for the year 2022)

```
y_test(actual) vs y_prediction
     Froto Endeks       0
0          232.81  233.37
1          235.40  231.05
2          241.22  242.15
3          239.28  242.15
4          237.53  242.15
..            ...     ...
99         272.88  291.49
100        274.91  290.92
101        290.24  294.58
102        298.53  294.00
103        295.06  298.63

[104 rows x 2 columns]
```

Table Appendix A.8 actual and prediction RFR test results for Froto closing stock price.

```
y_test(actual) vs y_prediction
          0           0
0     43.58   42.916433
1     51.05   49.859133
2    527.00  478.846267
3     50.68   51.065933
4     23.20   23.138033
..      ...         ...
231   43.76   43.891333
232   24.44   24.753600
233   37.76   35.655267
234   51.79   52.096900
235   72.18   69.136967

[236 rows x 2 columns]
```

Table Appendix A.9 actual and prediction RFR results for Froto closing stock price (unseen 104 datas for the year 2022)

```
y_test(actual) vs y_prediction
     Froto Endeks           0
0          232.81  228.125867
1          235.40  232.014133
2          241.22  240.051700
3          239.28  241.339400
4          237.53  241.277467
..            ...         ...
99         272.88  287.242300
100        274.91  287.228567
101        290.24  290.072800
102        298.53  293.636633
103        295.06  298.585100

[104 rows x 2 columns]
```

Table Appendix A.10 actual and prediction BRR test results for Froto closing stock price.

```
y_test(actual) vs y_prediction
          0           0
0     43.58   43.117734
1     51.05   50.274119
2    527.00  530.188040
3     50.68   52.141752
4     23.20   22.730894
..      ...         ...
231   43.76   44.514866
232   24.44   24.132079
233   37.76   37.851245
234   51.79   51.463752
235   72.18   71.663207

[236 rows x 2 columns]
```

Table Appendix A.11 actual and prediction BRR results for Froto closing stock price (unseen 104 datas for the year 2022)

```
y_test(actual) vs y_prediction
     Froto Endeks           0
0          232.81  231.465599
1          235.40  235.131486
2          241.22  237.858298
3          239.28  240.514841
4          237.53  237.225749
..            ...         ...
99         272.88  279.659157
100        274.91  281.843406
101        290.24  297.702106
102        298.53  305.596567
103        295.06  302.175572

[104 rows x 2 columns]
```

Table Appendix A.12 actual and prediction ANN test results for Froto closing stock price.

```
y_test(actual) vs y_prediction
           0           0
0      43.58   42.093979
1      51.05   49.526359
2     527.00  510.040949
3      50.68   48.613004
4      23.20   23.657225
..       ...         ...
231    43.76   49.062650
232    24.44   23.897290
233    37.76   36.726710
234    51.79   52.035204
235    72.18   73.415220

[236 rows x 2 columns]
```

Table Appendix A.13 actual and prediction ANN results for Froto closing stock price (unseen 104 datas for the year 2022)

```
y_test(actual) vs y_prediction
     Froto Endeks           0
0          232.81  217.053739
1          235.40  225.245564
2          241.22  227.683706
3          239.28  234.551708
4          237.53  234.329582
..            ...         ...
99         272.88  283.555532
100        274.91  285.847410
101        290.24  296.480093
102        298.53  301.375669
103        295.06  300.599140

[104 rows x 2 columns]
```

**APPENDIX B (Algorithm Results for Koc Holding Stock)**

Table Appendix B.1 Algorithms evoluation for unseen 2022 datas.

| Test Name | Algorithm | R Square | MAE | MSE | Feature Number | Methods | Parameters |
|---|---|---|---|---|---|---|---|
| Kchol_test 2 | MLR | 0.9858 | 0.3350 | 0.2020 | 22 | Min-max,cv | default |
| Kchol_test 4 | SVR | 0.9184 | 0.7956 | 1.1599 | 22 | ss, cv | kernel=linear |
| Kchol_test 6 | DTR | 0.9208 | 0.8538 | 1.1266 | 22 | Min-max,cv | default |
| Kchol_test 8 | RFR | 0.9509 | 0.6299 | 0.6985 | 22 | Min-Max,cv | n_estimators=300 |
| Kchol_test 10 | BRR | 0.9863 | 0.3475 | 0.1947 | 22 | Min-Max,cv | default |
| Kchol_test 12 | ANN | 0.9544 | 0.6564 | 0.6486 | 22 | Ss, cv | epoch=1000 lr=0.0001 |

Table Appendix B.2 actual and prediction MLR test results for Kchol closing stock price.

```
y_test(actual) vs y_prediction
         0          0
0    15.06  14.958780
1    17.23  17.353171
2    82.64  81.809716
3    15.29  15.065392
4    11.75  11.529820
..     ...        ...
231  15.69  15.848348
232  11.41  11.385973
233  15.95  15.920260
234  17.39  17.298249
235  19.03  18.950900

[236 rows x 2 columns]
```

Table Appendix B.3 actual and prediction MLR results for Kchol closing stock price (unseen 104 datas for the year 2022)

```
y_test(actual) vs y_prediction
     Kchol Endeks          0
0           29.59  29.415391
1           29.89  30.202578
2           30.73  30.614062
3           31.37  31.633828
4           32.06  31.562266
..            ...        ...
99          39.14  39.123291
100         38.51  38.995820
101         39.84  40.335381
102         40.08  41.039824
103         40.20  41.308184

[104 rows x 2 columns]
```

Table Appendix B.4 actual and prediction SVR test results for Kchol closing stock price.

| | y_test(actual) | y_prediction |
|---|---|---|
| | 0 | 0 |
| 0 | 15.06 | 15.436165 |
| 1 | 17.23 | 17.139599 |
| 2 | 82.64 | 80.126673 |
| 3 | 15.29 | 15.094289 |
| 4 | 11.75 | 12.259839 |
| .. | ... | ... |
| 231 | 15.69 | 15.273635 |
| 232 | 11.41 | 11.905561 |
| 233 | 15.95 | 15.880005 |
| 234 | 17.39 | 16.866960 |
| 235 | 19.03 | 18.559875 |

[236 rows x 2 columns]

Table Appendix B.5 actual and prediction SVR results for Kchol closing stock price (unseen 104 datas for the year 2022)

| | y_test(actual) vs | y_prediction |
|---|---|---|
| | Kchol Endeks | 0 |
| 0 | 29.59 | 30.850903 |
| 1 | 29.89 | 32.215977 |
| 2 | 30.73 | 31.876896 |
| 3 | 31.37 | 32.426209 |
| 4 | 32.06 | 32.879580 |
| .. | ... | ... |
| 99 | 39.14 | 38.923131 |
| 100 | 38.51 | 39.352341 |
| 101 | 39.84 | 40.015006 |
| 102 | 40.08 | 40.784260 |
| 103 | 40.20 | 40.311753 |

[104 rows x 2 columns]

Table Appendix B.6 actual and prediction DTR test results for Kchol closing stock price.

```
y_test(actual) vs y_prediction
         0      0
0    15.06  14.68
1    17.23  17.83
2    82.64  70.66
3    15.29  15.11
4    11.75  11.63
..     ...    ...
231  15.69  15.55
232  11.41  11.49
233  15.95  15.64
234  17.39  17.23
235  19.03  18.79

[236 rows x 2 columns]
```

Table Appendix B.7 actual and prediction DTR results for Kchol closing stock price (unseen 104 datas for the year 2022)

```
y_test(actual) vs y_prediction
     Kchol Endeks      0
0           29.59  30.59
1           29.89  29.75
2           30.73  30.59
3           31.37  30.94
4           32.06  31.84
..            ...    ...
99          39.14  39.32
100         38.51  39.20
101         39.84  39.32
102         40.08  39.32
103         40.20  39.26

[104 rows x 2 columns]
```

Table Appendix B.8 actual and prediction RFR test results for Kchol closing stock price.

```
y_test(actual) vs y_prediction
         0          0
0    15.06  14.968433
1    17.23  17.351967
2    82.64  76.401500
3    15.29  15.250267
4    11.75  11.684300
..     ...        ...
231  15.69  15.745833
232  11.41  11.433200
233  15.95  15.743433
234  17.39  17.268067
235  19.03  18.853633

[236 rows x 2 columns]
```

Table Appendix B.9 actual and prediction RFR results for Kchol closing stock price (unseen 104 datas for the year 2022)

```
y_test(actual) vs y_prediction
     Kchol Endeks          0
0           29.59  30.290267
1           29.89  30.175600
2           30.73  31.007500
3           31.37  31.204800
4           32.06  31.478300
..            ...        ...
99          39.14  38.661533
100         38.51  38.822467
101         39.84  39.290600
102         40.08  39.932733
103         40.20  39.798667

[104 rows x 2 columns]
```

Table Appendix B.10 actual and prediction BRR test results for Kchol closing stock price.

```
y_test(actual) vs y_prediction
         0          0
0    15.06  14.960658
1    17.23  17.345474
2    82.64  81.828539
3    15.29  15.071922
4    11.75  11.527965
..     ...        ...
231  15.69  15.809573
232  11.41  11.393600
233  15.95  15.862215
234  17.39  17.301061
235  19.03  18.941127

[236 rows x 2 columns]
```

Table Appendix B.11 actual and prediction BRR results for Kchol closing stock price (unseen 104 datas for the year 2022)

```
y_test(actual) vs y_prediction
     Kchol Endeks          0
0           29.59  29.483514
1           29.89  30.369418
2           30.73  30.738593
3           31.37  31.700506
4           32.06  31.745208
..            ...        ...
99          39.14  39.216459
100         38.51  39.107162
101         39.84  40.344089
102         40.08  41.067874
103         40.20  41.298261

[104 rows x 2 columns]
```

Table Appendix B.12 actual and prediction ANN test results for Kchol closing stock price.

```
y_test(actual) vs y_prediction
         0          0
0    15.06  14.906495
1    17.23  17.223084
2    82.64  81.703110
3    15.29  15.171661
4    11.75  11.676911
..     ...        ...
231  15.69  15.725376
232  11.41  11.422579
233  15.95  15.749786
234  17.39  17.258541
235  19.03  18.804615

[236 rows x 2 columns]
```

Table Appendix B.13 actual and prediction ANN results for Kchol closing stock price (unseen 104 datas for the year 2022)

```
y_test(actual) vs y_prediction
     Kchol Endeks          0
0           29.59  28.533310
1           29.89  30.535785
2           30.73  30.500633
3           31.37  31.883740
4           32.06  32.522657
..            ...        ...
99          39.14  37.722863
100         38.51  38.129102
101         39.84  39.021482
102         40.08  39.403005
103         40.20  39.435347

[104 rows x 2 columns]
```