DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

PREDICTION OF SOIL RADON GAS USING METEOROLOGICAL PARAMETERS WITH MACHINE LEARNING ALGORITHMS

By
Çağla ÖZTÜRK ZAN

October,2021
İZMİR

PREDICTION OF SOIL RADON GAS USING METEOROLOGICAL PARAMETERS WITH MACHINE LEARNING ALGORITHMS

A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Master of Science in
Statistics, Statistics Program

By
Çağla ÖZTÜRK ZAN

October,2021
İZMİR

M.Sc THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **"PREDICTION OF SOIL RADON GAS USING METEOROLOGICAL PARAMETERS WITH MACHINE LEARNING ALGORITHMS "** completed by **ÇAĞLA ÖZTÜRK ZAN** under supervision of **ASSOC. PROF. DR. NESLİHAN DEMİREL** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

……..

Assoc. Prof. Dr. NESLİHAN DEMİREL

_____

Supervisor

…… ……

Prof.Dr.Müslim Murat SAÇ Asst.Prof.Dr.Engin YILDIZTEPE

_____ _____

(Jury Member) (Jury Member)

_____

Prof.Dr.Özgür ÖZÇELİK

Director

Graduate School of Natural and Applied Sciences

ACKNOWLEDGEMENTS

I am indebted and would like to express my gratitude to my supervisor Associate Professor Neslihan DEMİREL for sparing her precious time for me to progress in this both challenging and teaching path, for never withholding her friendliness and sincerity, and for the precious knowledge she has shared with me, which I will made benefit for all my professional life. I would also like to thank to the Republic of Turkey Ministry of Agriculture and Forestry, General Directorate of Meteorology and Professor Müslim Murat SAÇ for supporting me every time I needed information and data. I would like to express my sincerest gratitude to Assistant Professor Engin YILDIZTEPE for sharing his precious knowledge and experience with me whenever I needed support. I would also like to thank to my dear friend Burak DİLBER, with whom I have discussed the subject, literature and method of my study and experiences of whom I have benefited to the fullest, with the belief that he will have a great run in the future.

I thank my other university teachers one by one for everything they have made me achieve in my academic training during my undergraduate and graduate training and also in real life, and for equipping me with information that will make me have a say in the future. And at last but not the least I would like to express my endless gratitude to my dear father Ahmet Öztürk, who brought me to these days, where I always felt his love, support and pride, and my dear husband Bahadır Zan, who patiently stood by me in this intense and sometimes challenging process, who trusted me and always supported me.

Çağla ÖZTÜRK ZAN

PREDICTION OF SOIL RADON GAS USING METEOROLOGICAL PARAMETERS WITH MACHINE LEARNING ALGORITHMS

ABSTRACT

Radon is the natural radiation source with the highest dose of exposure among all radiation sources found on earth. It consists of the degradation of natural uranium and radium elements in rocks. Factors that shape the movement of radon include meteorological factors such as the rate of decaying of radon isotopes, the fluids that fill the pores(air, water and other gases), atmospheric pressure, soil and air temperature, wind speed, and wind direction. The aim of this study is to evaluate the effects of some meteorological factors on Radon gas using Supervised Learning Algorithms and to estimate the radon gas values according to these factors. For the study, in addition to the radon levels obtained from the Seferihisar region in hourly periods between 30 October 2006 and 04 June 2007, the measurements for the parameters of Hourly Actual Pressure(hPa), Hourly 50 cm Soil Temperature(°C), Hourly Relative Humidity(%), Hourly Temperature(°C), Hourly Wind Degree(°), Hourly Wind Speed(m/sec) and Wind Direction were obtained from the Republic of Turkey Ministry of Agriculture and Forestry, General Directorate of Meteorology. To analyze the relationship between Radon and meteorological factors affecting radon with Supervised Learning Algorithms, Multiple Linear Regression, k-Nearest Neighbor, Support Vector Machines, Regression Trees, Bagging, Random Forests, XGBoost methods have been used. To test the success of applied methods K-Fold Cross-Validation(K=5) and verification tests were performed. Specification coefficient(R^2) for comparing the performance of algorithms, Mean Squared Error(MSE), Root Mean Squared Error(RMSE), Mean Absolute Error(MAE) values were used. The best result was random forests regression when performance criteria were taken into account. This method was followed by the XGBoost and k-Nearest Neighbors algorithms, which gave very close results.

Keywords: Supervised learning algorithms, soil radon gas, meteorological factor.

MAKİNE ÖĞRENMESİ ALGORİTMALARI İLE METEOROLOJİK PARAMETRELERİ KULLANARAK TOPRAK RADON GAZININ TAHMİNİ

ÖZ

Radon, yeryüzünde bulunan tüm radyasyon kaynakları içerisinde en yüksek doza maruz kalınan ve sağlığımıza olumsuz etkisi olan doğal radyasyon kaynağıdır. Kayaçlardaki doğal uranyum ve radyum elementlerinin bozuşmasından oluşur. Radonun hareketini şekillendiren faktörler arasında; Radon izotoplarının bozunma hızı, gözenekleri dolduran akışkanlar (hava, su ve diğer gazlar), atmosferik basınç, toprak ve hava sıcaklığı, rüzgar hızı, rüzgarın yönü gibi meteorolojik faktörler yer alır. Bu çalışmanın amacı bazı meteorolojik faktörlerin, Radon gazı üzerindeki etkilerinin Denetimli Öğrenme Algoritmaları ile değerlendirilmesi ve Radon gazı değerlerinin bu faktörlere göre tahmin edilmesidir. Bu amaçla 30 Ekim 2006 - 04 Haziran 2007 tarihleri arasında saatlik periyodlarla Seferihisar bölgesinden elde edilen Radon gazı seviyesine ek olarak T.C. Tarım ve Orman Bakanlığı Meteoroloji Genel Müdürlüğü'nden Saatlik Aktüel Basınç, Saatlik 50 cm Toprak Sıcaklığı, Saatlik Nispi Nem, Saatlik Sıcaklık, Saatlik Rüzgâr Derecesi, Saatlik Rüzgar Hızı ve Saatlik Rüzgar Yönü parametreleri ölçümleri elde edilmiştir. Radon ve radona etki eden meteorolojik faktörler arasındaki ilişkiyi analiz etmek için Denetimli Öğrenme Algoritmalarından Çoklu Doğrusal Regresyon, k-En Yakın Komşuluk, Destek Vektör Makineleri, Regresyon Ağaçları, Torbalama(Bagging), Rassal Ormanlar, XGBoost yöntemleri kullanılmıştır. Uygulanan yöntemlerin başarılarının sınanması için K çapraz-geçerlilik(K=5) ile doğrulama testleri gerçekleştirilmiştir. Algoritmaların performanslarını karşılaştırmak üzere Belirtme Katsayısı(R^2), Hata Kareler Ortalaması(Mean Square Error-MSE), Kök Hata Kareler Ortalaması(Root Mean Square Error-RMSE), Ortalama Mutlak Hata(Mean Absolute Error-MAE) değerleri kullanılmıştır. Performans kriterleri dikkate alındığında en iyi sonucu Rassal Ormanlar Regresyonu vermiştir. Bu yöntemi, birbirlerine çok yakın sonuçlar veren XGBoost ve k-En Yakın Komşuluk Algoritmaları takip etmiştir.

Anahtar kelimeler: Denetimli öğrenme algoritmaları, toprak radon gazı, meteorolojik faktörler.

CONTENTS

Page

M.Sc THESIS EXAMINATION RESULT FORM ii
ACKNOWLEDGEMENTS iii
ABSTRACT iv
ÖZ v
LIST OF FIGURES ix
LIST OF TABLES xi

CHAPTER 1 - INTRODUCTION 1

1.1 Radon Gas 6
1.2 Data Preprocessing 7
1.2.1 Missing Observation 8
1.2.2 Transformation 11
1.2.3 Outliers 11

CHAPTER 2 - MACHINE LEARNING ALGORITHMS 13

2.1 Regression Analysis 13
2.1.1 Assumptions of Multiple Linear Regression Model 15
2.1.2 Significance Test of Model Parameters 17
2.1.3 Significance Test of The Model 18
2.1.4 Transformations for Non-normal Distributed and Non-constant Variance Error Terms 18
2.1.5 Variable Selection and Model Building 19
2.2 Support Vector Machines 23
2.3 Regression Trees 28
2.4 Bagging 30
2.5 XGBoost (Extreme Gradient Boosting) 33
2.6 Random Forests 36

2.7 k-Nearest Neighbor-kNN 39

CHAPTER 3 - MODEL EVALUATION AND PERFORMANCE CRITERIA 42

3.1 K-Fold Cross-Validation 42
3.2 Model Performance Criteria 42
3.2.1 Coefficient of Determination (R^2) 42
3.2.2 Mean Square Error (MSE) 43
3.2.3 Root Mean Squared Error (RMSE) 43
3.2.4 Mean Absolute Error (MAE) 44

CHAPTER 4 - APPLICATION 45

4.1 Data Preprocessing 45
4.2 Multiple Linear Regression Analysis 53
4.3 Support Vector Regression Analysis 62
4.4 Regression Trees Analysis 65
4.5 Bagging Analysis 66
4.6 XGBoost (Extreme Gradient Boosting) Analysis 67
4.7 Random Forests Analysis 68
4.8 k-Nearest Neighbor (kNN) Analysis 69

CHAPTER 5 - CONCLUSION 71

REFERENCES 73

LIST OF FIGURES

Page

Figure 2.1 General structure of support vector machines 24
Figure 2.2 Regression trees 29
Figure 2.3 Bagging tree algorithm 32
Figure 2.4 Random forests algorithm 37
Figure 4.1 Importing the data set in R software 45
Figure 4.2 Conversion of the data types of the variables 46
Figure 4.3 Removing rows for missing observations of the radon variable corresponding to other variables 48
Figure 4.4 Assigning new values by imputation method instead to missing values .. 49
Figure 4.5 Histogram of radon 50
Figure 4.6 Transformation on radon variable and drawing histogram for radon 50
Figure 4.7 Histogram for radon after transformation 51
Figure 4.8 Anderson-Darling normality test for radon 51
Figure 4.9 Anderson-Darling normality test result for radon 52
Figure 4.10 Box plot of transformed radon 52
Figure 4.11 Removing outliers from the data set 53
Figure 4.12 Correlation chart 54
Figure 4.13 A chart of the degree and direction of the relationship between variable 54
Figure 4.14 VIF values for full model 55
Figure 4.15 Residual charts for the full model obtained after data preprocessing operations 56
Figure 4.16 Anderson-Darling normality test result 57
Figure 4.17 The best possible subset selection 58
Figure 4.18 Creating the final model 60
Figure 4.19 Analysis of model performance criteria with multiple linear regression algorithm 62
Figure 4.20 Analysis of model performance criteria of support vector regression algorithm for linear kernel function 63

Figure 4.21 Analysis of model performance criteria of support vector regression algorithm for polynomial kernel function 64
Figure 4.22 Analysis of model performance criteria of support vector regression algorithm for radial basis kernel function 65
Figure 4.23 Analysis of model performance criteria for regression trees algorithm . 66
Figure 4.24 Analysis of model performance criteria for bagging algorithm 67
Figure 4.25 Analysis of model performance criteria for the xgboost algorithm..... 68
Figure 4.26 Analysis of model performance criteria for random forests algorithm .. 69
Figure 4.27 Analysis of model performance criteria of k-nearest neighbor algorithm 70

LIST OF TABLES

Page

Table 4.1 First 10 observations of the data set 46
Table 4.2 Data types of variables 46
Table 4.3 Corrected data types 47
Table 4.4 Summary statistics of variables 47
Table 4.5 Summary statistics after removed the missing values of Radon 48
Table 4.6 Summary statistics after completing missing values 49
Table 4.7 VIF values for full model 55
Table 4.8 Breush-Pagan test for variance homogeneity 56
Table 4.9 Summary results for the full model 58
Table 4.10 Examples of possible submodels obtained by the best possible subset selection method 59
Table 4.11 Forward selection method summary table 59
Table 4.12 Backward elimination method summary table 60
Table 4.13 Summary results for the final model 61
Table 4.14 Model performance criteria for multiple linear regression algorithm 62
Table 4.15 Support vector regression algorithm model performance criteria for linear kernel function 63
Table 4.16 Support vector regression algorithm model performance criteria for polynomial kernel function 64
Table 4.17 Support vector regression algorithm model performance criteria for radial basis kernel function 65
Table 4.18 Model performance criteria for the regression trees algorithm 66
Table 4.19 Model performance criteria for the bagging algorithm 67
Table 4.20 Model performance criteria for xgboost algorithm 68
Table 4.21 Model performance criteria for random forests algorithm 69
Table 4.22 Model performance criteria for the k-nearest neighbor algorithm 70
Table 5.1 Model performance criteria obtained for all algorithms 72

CHAPTER 1
INTRODUCTION

Radiation is electromagnetic waves that we are exposed to in all aspects of everyday life. Radon gas is a radioactive substance, which is one of the various components of radiation, that is often found in nature and affects our health. Despite being mostly an airborne noble gas, radon is also found in soil, rocks, water and indoor buildings. The main source of radon found in the buildings is soil. The average annual effective dose of radon exposure is estimated to be 1.3 mSv (55% of the total effective dose) (Yücel et al., 2012).

About half of the total dose that people receive from natural radiation sources is the inhalation of radon and its degradation products, and radon ranks second in lung cancer risk, after smoking. In an article published in the Green Building - Sustainaible Construction Technologies Journal in 2017, the negative health effects of radon depending on the level and duration of exposure (Değerli & Umaroğulları, 2017). Therefore, the identification and mapping of risky areas where radon is located has gained a great speed in different countries of the world in order to protect people from exposure to high concentrations of radon. Numerous methods have been developed to identify areas with high radon potential. The most recent and alternative of these methods is the creation of geology-based radon risk maps.

Apart from the harmful effects of radon on health, its effect on earth crust movements has also been observed in various studies. The damages caused by the earthquakes have been felt more and more for many years as a result of urbanization. For this reason, earthquake prediction studies are also increasing. Since an earthquake is caused by movements in the Earth's crust, it contains many parameters. Researchers are trying to understand the effects of these parameters on the earthquake, which occur before, during and after the earthquake. In many studies, it was noted that there were a number of changes in the concentrations of some gases coming out of the soil near the area where the earthquake occurred, before or after the earthquake. One of these gases is Radon (Tabar, 2010).

The relationship between radon concentration and fault lines by taking soil structure and geological features in Afyonkarahisar was conducted by İlhan(2015).

Radon is the most studied gas in the studies conducted to estimate earthquakes before they occur. Since radon is a noble gas that comes out of the soil to the atmosphere, it does not form a chemical compound, and it can dissolve in water and reach the atmosphere. Meteorological factors and seismic activities affect the outflow of radon gas from the soil to the atmosphere. Thus, radon is used in studying the movements of the Earth's crust (Günay, 2016).

Multifarious studies have been conducted in the literature on the effects of meteorological factors, which are other factors affecting the release of Radon into the atmosphere and the concentration of soil radon gas (Kamışlıoğlu, 2016; Gürleyen, 2017; Pişkin, 2017). These studies were usually carried out by examining parameters such as soil depth, soil and air temperature, air pressure, wind speed, wind direction, and the elevation of the place where the measurement was made. In their study in the Western Marmara region, Günay(2016) observed that the concentration of radon varies according to meteorological factors and seismic activity.

In this study, Machine Learning Algorithms were used to analyze the relationship between variables in Radon and various parameters acting on radon. Machine Learning Algorithms are widely used in data mining and big data management today. The term Machine Learning was first described in 1959 by American computer scientist Arthur Samuel as “the ability of a computer to learn without being explicitly programmed”. Machine Learning Algorithms are code snippets that help discover, analyze, and find meaning in complex data sets. Each algorithm is a limited and specific set of step-by-step instructions that a machine can follow to accomplish a specific goal. The goal of the Machine Learning model is to create models or patterns that people can use to make predictions or categorize information.

Machine Learning Algorithms are generally examined under two headings: supervised and unsupervised. There are quite a variety of algorithms under these two

headings. In this study, the Multiple Linear Regression, Support Vector Machines, Regression Trees, Bagging, XGBoost, Random Forests, and k-Nearest Neighbors Algorithm among widely used Supervised Machine Learning Algorithms, were used.

While Simple Linear Regression Analysis examines the causal relation between the dependent (response) variable and one independent (explanatory) variable; Multiple Linear Regression used in this study is a statistical method that models the dependent variable and multiple independent variables. Its purpose is to determine the extent to which the dependent variable is explained by the arguments. Data preprocessing is very important for Multiple Linear Regression Analysis. Distributions of variables in the dataset, have many assumptions based on outlier observations, parameter estimates, and residual estimates. In Regression Analysis, it is important to provide relevant assumptions in order to obtain a healthy model. In general, Regression Analysis it is studied in many areas such as medicine, biology, economics, engineering, agriculture, geology, sociology, etc.

Support Vector Machines are kernel basis Machine Learning Algorithms based on the maximum margin principle used in solving complex model problems. They are used for both classification and regression problems. Although they are used with the most common and first proposed classification problems, in this study, the Support Vector Machines Regression was applied due to the data set used in this study. Support Vector Machines have the ability to be applied to both linear and nonlinear data. In this study, kernel functions were applied because the distribution of the dependent variable is not linear. Although multiple kernel functions are used in the literature, the most commonly used are linear, polynomial, radial basis functions. Comparisons have been made concerning the relationship between radon and meteorological parameters using these three kernel functions. Support Vector Machines are most commonly used in classification applications such as facial recognition systems, and voice analysis.

Regression Trees are another Machine Learning Algorithms, which are tree-based methods used in Data Mining. They are known as Classification and Regression

Trees (CART), based on decision trees. Classification Trees are used for classification problems, Regression Trees are used for regression problems. The Regression Tree comprises of branches and leaves with its tree structure. Since this structure is more understandable algorithmically in solving problems, it is one of the frequently preferred methods in many fields in statistical studies.

In scientific studies, there are recently very popular Machine Learning based Ensemble Learning Algorithms created from basic learners, which have a wide range of uses, and which have higher performance success than the basic trainer. They are used in both classification and regression problems. The reason it is called Ensemble Learning Algorithms is because it uses the Decision Trees Algorithm established in ensemble instead of individual trees. Bagging, Boosting, Random Forests, Stacking and Voting are the most common ones among the Ensemble Learning Algorithms. Among these, Bagging, Boosting based XGBoost and Random Forests Algorithms were used in this study.

Bagging (Bootstrap Aggregating), an ensemble learning method used in predictive analytical studies in solving both classification and regression problems, was developed by Breiman in 1994. The Bagging Algorithm aims at retraining the basic learner by creating new training sets from an existing training set and to generate the training set by random selection.

XGBoost was introduced to the literature by Tianqi Chen and Carlos Guestrin (Chen & Guestrin, 2016). Chen It is the high-performance version of the Gradient Boosting method based on Ensemble Learning Algorithm. Software and hardware optimization techniques were applied to achieve superior results using fewer resources. It is shown that decision tree-based algorithms are best in achieving high predictive power, preventing over-learning, managing empty data and performing these operations ten times faster.

Random Forests is a Machine Learning Algorithm that creates multiple decision trees and combines them to produce a more accurate and stable prediction. Breiman,

who introduced the Bagging method in 1996, teamed up with Adele Cutler in 2003 to advance the Bagging method and developed the Random Forests method. It is very popular in recent years because it can be applied to both regression and classification problems, which also give good results without hyper parameter estimation. It is used in multifarious studies in areas such as genetics and bioinformatics.

k-Nearest Neighbors Algorithm is used to solve both classification and regression problems, but it is mostly used in classification problems in the industry. It was introduced by Fix and Hodges in 1951 for use in pattern (model) recognition and developed by Cover and Hart in 1967. k-Nearest Neighbors Algorithm is a Machine Learning Algorithm that makes a classification or prediction that is determined by the *k* its value of the class or the nearest neighbor of a specified data point.

The effects of some meteorological factors on radon gas were evaluated using the briefly mentioned Machine Learning Algorithms. Coefficient of Determination (R^2), Mean Square Error (MSE), Root Mean Square Error (RMSE), Mean Absolute Error (MAE) values and K-Fold Cross-Validation tests were used for comparing the performance of algorithms.

For the study, in addition to the radon levels obtained from the Seferihisar region in hourly periods between 30 October 2006 and 04 June 2007, the measurements for the parameters of Hourly Actual Pressure(hPa), Hourly 50 cm Soil Temperature(°C), Hourly Relative Humidity(%), Hourly Temperature(°C), Hourly Wind Degree(°), Hourly Wind Speed(m/sn) and Wind Direction were obtained from the Republic of Turkey Ministry of Agriculture and Forestry, General Directorate of Meteorology, corresponding to the same region and the same date range.

In the first chapter of the study, detailed information about Radon gas is given and the operations performed on the data set were described and the data pre-processing steps were performed. In the second chapter, Machine Learning Algorithms are explained in detail. In the third chapter, K-Fold Cross-Validation and Model Performance Criteria are briefly mentioned. In the fifth chapter, the applications of

Machine Learning Algorithms are included, and in the sixth chapter, the study is concluded with the conclusion of the analysis.

1.1 Radon Gas

History of Radon dates back to 1899, to the detection of a radioactive gas released by thorium in a study by Ernest Rutherford and Robert B. Owens. In the same year, a radioactive gas emitted from radium was detected in Pierre and Marie Curie's work on radioactivity. In 1900, a gas deposited in a radium bulb was discovered by Friedrich Ernst Dorn in Halle, Germany. This gas is Radon gas, which is isotope of radium with a long life. In addition, Radon's Toron (220Rn) with a half-life of 56 seconds was detected by Rutherford, and research on this new gas continued. As a result, it has been proven that it is also possible to concentrate this gas into a liquid state. In 1908, it was also determined by William Ramsay and Robert Whytlaw-Gray that Radon gas was the heaviest gas known (Pişkin, 2017).

Radon gas is a colorless, tasteless, odorless, unstable radioactive substance found in a pure state in nature, which can occur naturally. It is represented by the symbol Rn. It's measurement unit is Becquerel (Bq / m^3: is the amount of radon gas in 1 m^3 of air and is expressed by the Becquerel unit). Radon is the natural radiation source with the highest dose of exposer among all radiation sources found on earth. It is produced naturally during the radioactive decay of thorium and uranium, usually found in soil and rocks. It is the heaviest in density compared to other noble gases. Its density is 9.72 G/L at 0°C and is eight times heavier than air. It is the inert gas with the highest boiling point, melting point, freezing point, critical temperature and critical pressure. In addition, it is the only radioactive element of the group of inert gases, but it does not react chemically with any other member of the group. Besides, it is soluble in cold water and its solubility decreases with as the temperature increases. When the temperature reaches its maximum point, it loses its solubility in water (Göker, 2010). Radon, which is gaseous under normal conditions, liquefies at -61.8 ℃ and freezes at -71℃ (Aşık, 2019).

The formation and quantity of radon gas and decay products are related to the geological characteristics of the living environment. In this context, the most important basic sources of radon gas are rocks and soil derived from rocks; however, air and water are among these basic sources. Radon gas occurs when the radioactive radium (226-Ra) contained in the Uranium chain is separated from the molecule to which it is bound by the kinetic energy it receives during its transformation to stable lead (206-Pb). Radon, which breaks off from the molecular bond, moves from the soil to the atmosphere, mixing with the gaseous medium (Göker, 2010). Significant changes in Radon values occur during this transition. Among the factors that shape the movement of radon are meteorological factors such as the rate of decaying of radon isotopes, the fluids that fill the pores(air, water and other gases), and atmospheric pressure. These meteorological factors include soil and air temperature, air pressure, wind speed, and wind direction. There are many studies investigating the effects of these meteorological parameters on radon gas (Arıkpınar, 2010).

1.2 Data Preprocessing

The data in the database must go through a preliminary process before analysis. This is required for preparing for data analysis. The most time-consuming and most laborious process of the total process is data preprocessing. A number of technical errors and errors by the text editor are possible during data collection and registration. Even if there are no errors, the dataset may have extreme values or missing observations. Correction of deficiencies, errors, inaccuracies, inconsistencies and outlier observations in the data set is of great importance. If there are inconsistencies in data, it effect the accuracy of the results, hence to remove the outliers or to impute the missing values make the data consistent. In this case, it is necessary to apply transformation techniques in the data set of the corresponding variable. In order to improve the quality of the data set, such data is identified, deleted from the data set, completed or converted by existing methods (Cebeci, 2020).

1.2.1 Missing Observation

It is possible to find missing values in data sets, especially in large volume data sets. A missing value issue may occur during data collection or during data recording for various reasons. For instance, the failure in recording of an automatic recording device due to a short-term malfunction or power failure causes missing observations to occur in that time frame. Besides, data can be entered incomplete by the staff registering the data. In such cases, the data set must be completed in a healthy way using existing methods (Cebeci, 2020).

Missing values can occur in different ways:

- Missing Completely at Random (MCAR)
- Missing at Random (MAR)
- Missing Not at Random (MNAR)

In MCAR, the missing values occur purely randomly, not due to dependent or independent variables. For instance, in a survey study, the participants may not have answered a question, skipped it, or wanted no to answer a specific question. It is observed in such cases. Since the probability of a missing value is the same for all observations in the dataset, simple methods are sufficient to complete the missing values. Although MAR values are random, they occur depending on different levels of another dependent variable. For instance, in a survey study, senior managers may not answer the question about their salaries. MNAR values are non-randomly occurring values depending on the variable itself or the values of some other variables. Such missing values are quite difficult to detect and complete (Cebeci, 2020).

It is necessary to determine the missing values in the data set and correct this problem with an appropriate method. If the missing values are large enough to undermine the reliability of analysis and statistics, it will be healthier to repeat the

survey, experiment or observation rather than completing it. Numerous methods have been proposed to solve the missing value problem or complete the missing values.

The methods for processing missing values are as follows.

- Deletion Methods
- Imputation Methods
    - Substitution Methods
    - Model-Based Completion Methods

The most commonly used missing value processing methods in data mining applications and statistical analysis are deletion methods. Deletion methods are processed in 3 main headings: Listwise Deletion, Pairwise Deletion and Variable Deletion methods. The most common Method of Deletion is the Listwise Deletion Method, which is the method of completely deleting the line where the missing value is located. Although the Listwise Deletion Method is a simple and practical method in a dataset with missing values, it is at a disadvantage in most cases in terms of creating information loss. This method rarely supports a MCAR values. This, in turn, causes the information of subjects with certain characteristics to be destroyed. Thus, biased parameter estimates are obtained. This is because the MCAR value assumes that the missing value in the dependent variable is unrelated to the observations of both itself and other variables. Therefore, it is recommended that the number of missing values should not be more than 5% of the total number of records in the dataset, in order to implement the Listwise Deletion Method (Cebeci, 2020).

The Pairwise Deletion Method, which is one of the other deletion methods, works by deleting missing values in terms of at least one of the pairs of variables in a bivariate statistic. However, if the calculated statistic belongs to a single variable, a statistical calculation is performed based on observations in the corresponding variable. Since there is only one missing value, instead of deleting the entire data record, pairs of variables that are complete are used, depending on the analysis. Thus, such analyses are called the Analysis of Cases Containing Information. Since it uses

all pairs that are not missing, the power of analysis is higher than the Listwise Deletion Method. However, applying this method in multivariate analyses causes many disadvantages. Due to the calculation of standard errors from the average sample size in many analyses, the standard error can be calculated too high or too low when using the Pairwise Method (Cebeci, 2020).

The process of completely deleting variables that contain more missing values than others in the dataset is called the variable deletion method. The Variable Deletion Method is usually applied if the missing value ratio in a variable is more than 5% and the variable is trivial.

Most of the time, deleting the missing value or the variables with missing value leads to deviations or erroneous results in analysis. Therefore, instead of deleting the corresponding missing values, the appropriate values can be imputed instead of these values. In such cases, Completion Methods called imputation are applied. Imputation Methods are divided into Substitution Methods and Model-Based Completion Methods. Both of these methods have sub-methods (Cebeci, 2020).

Substitution with Measures of Central Tendency is the most commonly used Method of Substitution. This process attempts to replace the measures of central tendency such as average, median, or mode with missing values. Although it is a very simple and fast method, there is also a disadvantage, such as the inability to use correlations between variables. An excessive decrease in the variance of the variable can be observed when used with an excessive number of missing values. In such cases, another Substitution Method, the k-Nearest Neighbors (kNN) Method, can be used. The implementation of this method is the substitution of the missing values with the appropriate values using the k-Nearest Neighbors Method. Other Substitution Methods are Using Dummy Variables and Regression Equation Estimation. Model-Based Completion Methods can also be examined in 3 main headings: Maximum Likelihood, Expectation Maximization and Multiple Imputation. In this study, the Listwise Deletion Method and the Imputation with Measures of Central Tendency were used (Cebeci, 2020).

1.2.2 Transformation

It is the process of transforming the values of variables using a specific function, depending on the aim and case. It is often used in data mining. A data transformation is a displacement process that changes the shape of distribution of the variables or the direction of the relationship between them. Variables can be converted for many reasons. Transformation operations can be performed for basic aims such as normalizing distribution, standardizing, linearizing, reducing sensitivity to outliers, and reducing distortion. In addition, data transformation operations can be applied in order to make variable values comparable if they are measured by different units, to ensure linear relationships between variables, and to produce artificial or derivative variables. Although various tranformation functions are available, they are elaborated in Section 2.1.4 Transformations for non-normal and inequal error variances in Chapter 2 (Cebeci, 2020).

1.2.3 Outliers

Accurate calculation of statistics and model parameters is of great importance in data mining applications and statistical analysis. Outliers must be extracted from the data sets, before starting statistical data analysis, Detecting an outlier is an important data preprocessing step. Outliers are observations that violate statistical assumptions, behaving differently from other observations in the dataset. The purpose of detecting outliers is primarily to clear the data. In addition to clearing data, it is used to detect unexpected situations, unusual events, fraud attempts, rare events, etc. There are several methods and approaches used to determine outliers. These methods and approaches are grouped as Distribution-Based, Depth-Based and Clustering-Based Methods (Cebeci, 2020).

A method that works to find values that do not belong to a distribution, are unlikely to be, and therefore do not meet the assumptions of the relevant distribution is a Distribution-Based Method of determining outliers. This method is based on some statistical tests. They are not efficient in high-volume data sets (Cebeci, 2020).

Outlier value detection can also be done by measuring the proximity, density and distance of observations in the dataset. Depth-Based Methods (Depth-Based Outlier Detection) consider an observation an outlier if it is not close enough to other observations. This detection can also be made according to the intensity. Observations located in a place with a lower density than in the region where the observations are dense are defined as an outlier value. The efficiency of this method varies according to the proximity distance metric it determines, and the fact that it does not need a prior assumption about data distribution is one of the reasons why it is preferred (Cebeci, 2020).

Another method, Clustering-Based Methods, assume that normal data elements belong to large and dense sets, and outliers belong to smaller sets away from these sets, or do not belong to any sets (Cebeci, 2020).

After the outliers are determined, they be removed from the data set in certain proportions by the researcher according to the size of the data set, statistical analysis, assumptions. This process is very sensitive because excessive data loss will negatively affect the results of the analysis.

CHAPTER 2
MACHINE LEARNING ALGORITHMS

Machine Learning is generally studied in two main groups: supervised and unsupervised. Supervised Machine Learning gives the computer a known set of input data and known responses to the data, then creates a model that makes evidence-based reasonable estimates for the response to new data, including uncertainty. In Unsupervised Machine Learning, this process is performed only with input data. In other words, there are no output values corresponding to these inputs (Albalate & Minker, 2013). The k-Nearest Neighbors, Linear Regression, Random Forests, Decision Trees, Support Vector Machines, etc. are given as examples of Supervised Machine Learning Algorithms. Unsupervised Machine Learning Algorithms include clustering, dimension reduction, etc. Supervised Machine Learning Algorithms are given in this study.

2.1 Regression Analysis

Regression Analysis is a statistical method to examine the relationship between the dependent variable and the independent variable(s). Regression Anaylsis can be applied to many different application areas such as engineering, medicine, physics, social sciences, chemistry, economics, etc.

It is one of the best methods to understand the cause-and-effect relationship between at least two variables. The expression of this relationship as a mathematical function is a regression equation. By means of the regression equation, it is possible to determine the extent to which the dependent variable is explained by the independent variable(s) in order to determine the causal relationship more effectively.

The regression equation varies according to the number of independent variables. If the equation contains a single independent variable, it is called Simple Linear Regression Model, and if it contains more than one independent variable it is called

Multiple Linear Regression Model. In this study, Multiple Linear Regression model was used. Therefore, the simple linear regression model will be briefly mentioned and the elaboration will continue through the Multiple Linear Regression model.

A Simple Linear Regression equation that shows the relationship between a dependent variable and a single independent variable is formed as follows:

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i \qquad (2.1)$$

Here; Y is the dependent (response) variable, X is the independent (explanatory) variable, β_0 is the constant term of the equation (y-axis cutting point), β_1 is independent variable coefficient (slope), and ε is the error term.

A Simple Linear Regression equation that shows the relationship between a dependent variable and multiple independent variables is formed as follows:

$$Y_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \ldots + \beta_k X_{ki} + \varepsilon_i \ , \ i = 1,2,\ldots,n \qquad (2.2)$$

In this equation; Y is the dependent variable, X_{1i} , … , X_{ki} are independent variables, $\beta_j \ j = 0,1,\ldots,k$ are parameters of regression coefficients, and ε is the error term. β_j parameter shows the expected change in Y dependent variable corresponding to one unit of change in the X_j independent variable, when all other independent variables contained in the model are kept constant. Therefore, β_j $j = 1,2,\ldots,k$ parameters are called partial regression coefficients.

Expression in matrix format is more appropriate when dealing with multiple linear regression models. Matrix representation of the model shown in equality (2):

$$\boldsymbol{Y = X\beta + \varepsilon} \qquad (2.3)$$

Here;

$$Y = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix}_{n \times 1} \quad X = \begin{bmatrix} 1 & X_{11} & X_{21} & \dots & X_{k1} \\ 1 & X_{12} & X_{22} & \dots & X_{k2} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & X_{1n} & X_{2n} & \dots & X_{kn} \end{bmatrix}_{n \times (k+1)} \quad \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_k \end{bmatrix}_{(k+1) \times 1}$$

$$\varepsilon = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}_{n \times 1} \qquad (2.4)$$

$\boldsymbol{Y}$ is the $(n \times 1)$ dimensional dependent variable vector, $\boldsymbol{X}$ is the $n \times (k+1)$ dimensional independent variables matrix, $\boldsymbol{\beta}$ $(k+1) \times 1$ is the dimensional regression coefficients vector, and $\boldsymbol{\varepsilon}$ is the error terms vector sized $(n \times 1)$.

2.1.1 Assumptions of Multiple Linear Regression Model

1. The regression model is linear. β coefficients must have the power one in Equation 2.2.
2. X values do not change in iterative samples. It is assumed that X is not probabilistic. The values received by X should not change depending on the probability. (X is nonstochastic.)
3. The mean of the error term is zero. The mean of $E(\varepsilon_i \mid X_1, \dots, X_k) = 0$.
4. ε_i is homogeneous. (Homoscedasticity) $V(\varepsilon_i) = \sigma^2$
5. Error terms are independent of each other. $Cov(\varepsilon_i, \varepsilon_j) = 0$ $(i \neq j)$
6. The covariance of ε_i and X_i is zero. $Cov(\varepsilon_i, X_{ji}) = 0 \quad j = 1,2, \dots, k$
7. Number of observations n, must be more than the number of parameters to be estimated. In the literature, it is generally recommended to have an observation number at least ten times the number of independent variables.
8. Independent variables must have variance. In a certain sample not all values X_i should be the same.

9. The regression model must be established correctly. No errors should be made in setting up the model.

10. There should be no multicollinearity.

Multicollinearity is a lineer relationship between independents variables in Multiple Lineer Regression. The problem of multicollinearity negatively affects the model that will be established in Regression Analysis. In addition, the efficiency of the Least Squares Method used to estimate parameters in the regression model will be reduced in the presence of multicollinearity. In statistical analysis, researchers aim at ensuring that the estimators they obtain are unbiased, minimum variance, and consistent. In the case of multicollinearity, estimators will not be able to have these properties, and the results obtained will be insufficient and erroneous. Other problems that will arise in the case of multicollinearity are the uncertainty of the values of regression coefficients, growth of variance of regression coefficients, a decrease of t statistics, growth of confidence intervals, and R^2 resulting greater than it is.

Varıiance Inflation Factor and Correlation Coefficient are used to detect the presence of multicollinearity in this study.

The level of correlation between the arguments is determined by the correlation coefficient. The correlation matrix R = X'X is used, which shows the level of this relationship. Here, X is the data matrix for $n \times p$ dimensional arguments. ($p = k + 1$, p is the number of parameters.) Correlation coefficient contained in the correlation matrix, value of (r) close to 1 indicates that there are multicollinearity.

Another method used to detect the problem of multicollinearity is the Variance Inflation Factor. This factor is determined using the the inverse of the standardized correlation matrix $C = R^{-1} = (\mathrm{X'X})^{-1}$, and denoted by VIF. The variance inflation factor of variable X_j is shown in Equation 2.5 (Chatterjee & Price, 1997).

$$VIF_j = C_{jj} = \frac{1}{1 - R_j^2} \quad , \quad j = 1,2, \dots , k \qquad (2.5)$$

If there is no relationship between independent variable X_j and other independent variables, then VIF_j = 1, since R_j^2 = 0. Thus, there is no problem of multicollinearity. In the case of the opposite situation, i.e. if there is a complete relationship between the independent variable X_j and other independent variables, VIF_j will be infinite since R_j^2 = 1, which means a full multi-linear connection. In different sources in literature, it is stated that there is a problem with multicollinearity if the value of VIF_j is greater than 5 or 10 (Montgomery et al., 2013).

2.1.2 Significance Test of Model Parameters

Hypotheses for the significance test of the parameter when $j = 0,1, \dots , k$ is β_j in the Multiple Linear Regression model denoted by the Equation 2.2 is established as:

$$H_0 : \beta_j = 0$$
$$H_1 : \beta_j \neq 0 \qquad (2.6)$$

This test, also known as the significance test of partial regression coefficients, tests the contribution of X_j when there are other independent variables in the model. The test statistic used for this hypothesis is as below (Montgomery et al., 2013);

$$t_0 = \frac{\hat{\beta}_j}{\sqrt{\hat{\sigma}^2 C_{jj}}} = \frac{\hat{\beta}_j}{se(\hat{\beta}_j)} \qquad (2.7)$$

Here C_{jj}, is the diagonal element j. in matrix $(X'X)^{-1}$. If $|t_0| > t_{a/2,n-k-1}$ then $H_0 : \beta_j$ = 0 hypothesis is rejected. In this case, the coefficient β_j is significant at

confidence level $\%(1-\alpha)$, independent variable X_j can remain in the model. If hypothesis H_0 cannot be rejected, independent variable X_j can be removed from the model.

2.1.3 Significance Test of The Model

In other words, the regression significance test is a test used to decisively determine whether there is a linear relationship between the dependent variable and the independent variables. This process is often a test of the significance of the whole model. The appropriate hypothesis test is as follows:

$$H_0: \ \beta_1 = \beta_2 = \ldots = \beta_k = 0$$
$$H_1: \text{ At least one } \beta_j \neq 0 \quad j = 1,2,\ldots,k \qquad (2.8)$$

The test statistic used for this hypothesis is as below;

$$F = \frac{SSR/k}{SSE/n-p} \qquad (2.9)$$

Here SSR is the Sum of Squared Regression, SSE is the Sum of Squared Errors, k is the number of variables, p is the number of parameters. H_0 is rejected at confidence level $\%(1-\alpha)$ when test statistics F is greater than the table value of $F_{\alpha,k,n-p}$. The rejection of H_0 means that at least one of the independent variables makes a significant contribution to the model. It is necessary to decide which variables should remain in the model using variable selection methods.

2.1.4 Transformations for Non-normal Distributed and Non-constant Variance Error Terms

Unequal error variances and non-normal distribution of error terms are problems that often appear together. Constant variance assumption is one of the basic

assumptions of Regression Analysis. Detection and correction of non-constant error variance is important. If this problem is not eliminated, the least squares estimators will again be unbiased, but they will not carry the property of minimum variance. This, in turn, means that the standard errors of regression coefficients will be greater than they should be. In order to correct these deviations that occur in the regression model, the shape and spread of the distribution of the *Y*-dependent variable must be changed (Neter et al., 1996). This problem can be eliminated with appropriate transformations on the dependent variable. Depending on the distribution of the dependent variable, types of transformations such as Square Root Transformation, Logarithmic Transformation and Inverse Transformations can be applied on the dependent variable (Montgomery et al., 2013). The transformation that will be performed on the dependent variable can also contribute to the linearization of the curvilinear regression relationship. By assumption, there must also be a linear or almost linear relationship between the dependent variable and the independent variables. A simultaneous transformation on *X* may also be required to obtain or maintain a linear regression relationship (Neter et al., 1996).

2.1.5 Variable Selection and Model Building

One of the most important problems of Regression Analysis is the selection of variables in the model. A set of independent variables is often created based on the researcher's previous experience or theoretical views. All independent variables must be taken into the model to explain the dependent variable, but not all of these candidate variables are really useful for the model.

Important variable selection methods to be used in variable determination are as follows:

1. Best Subset
2. Stepwise Regression
    a. Forward Selection
    b. Backward Elimination

2.1.5.1 Best Subset

In this approach, all possible models are established, first as one-variable models, then two-variable, and then three-variable models. 2^k equality is obtained for the k candidate variable. (The first model is only the model in which the fixed term exists.) The "best" model is decided taking various criteria for each model (R_p^2 , R_{adj}^2 , MSE, Mallow C_p , $Press_p$) into account.

The criteria used to decide the "best" model are summarized below.

R_p^2 or SSE_p Criteria: Let R_p^2 be a multiple specification coefficient for a subset regression model that contains p number of terms i.e. k argument and a cut point for β_0. This criterion is calculated using the equation below:

$$R_p^2 = \frac{SSR_p}{SST} = 1 - \frac{SSE_p}{SST} \qquad (2.10)$$

Here SSR_p is the Sum of Squared Regression for the termed subset model p, SSE_p is the Sum of Squared Error for the term subset model p. Since 1 and SST is constant in this Equation 2.10, R_p^2 and SSE_p are inversely proportional. R_p^2 will increase as SSE_p decreases. The value SSE_p will never increase with each *X* variable added to the model, it will decrease or it will not change. In this case R_p^2 will reach its maximum value while all variables are in the model. This can be defined as the disadvantage of this criterion.

R_{adj}^2 or MSE_p Criteria: Since R_p^2 does not take into account the number of parameters in the model and will never decrease, it is presented as a different criterion R_{adj}^2 (corrected R^2).

$$R_{adj}^2 = 1 - \frac{SSE/(n-p)}{SST/(n-1)} = 1 - \frac{MSE}{SST/(n-1)} \qquad (2.11)$$

This criterion takes into account the number of parameters in the model based on the degrees of freedom. R^2_{adj} will increase if and only if MSE decreases.

C_p Criteria :

$$C_p = \frac{SSE_p}{MSE(X_1, X_2, \dots, X_{p-1})} - (n - 2p) \quad (2.12)$$

When evaluating the criterion C_p in general, a small value of C_p is preferred. C_p being small means a model with a lower total error. Here it is desired that C_p is equal to the number of parameters.

- If $C_p = p$, there is no bias in the model.
- If $C_p > p$, the model is biased.
- If $C_p < p$, there is no bias, but a sampling error has been made.

Press_p Criteria;

$$Press_p = \sum(Y_i - \hat{Y}_{(i)})^2 \quad (2.13)$$

Here Y_i is the value of the i observation in the dependent variable, $\hat{Y}_{(i)}$ is the estimate value after the observation i is removed. A small value of $Press_p$ indicates that the model is well established.

2.1.5.2 Stepwise Regression

It is the most commonly used variable selection method. A variable is added or subtracted from the model at each step. The criterion used for addition or subtraction is the partial F test.

F_{in} : The F value used to add variables to the model.

F_{out}: The F value used to extract variables from the model.

It should be $F_{in} \geq F_{out}$ (usually $F_{in} = F_{out}$). In Stepwise Regression, the entry of the first variable into the model is determined by having the highest correlation with the dependent variable. At the same time it will have the largest F statistics. The variable that has a value the greatest F value is taken into the model (if $F > F_{in}$). Partial F values are recalculated with all other variables, while the corresponding variable is in the model. If the variable with the greatest partial F value is greater than F_{in}, it is taken into the model. At this stage, if the partial F value of the variable previously added into the model is smaller than F_{out}, then the variable is subtracted. In this way, with each variable taken into the model, control of the variables previously entered into the model is performed. The process continues until there are no new variables left to enter or exit the model.

a. Forward Selection

It is a simpler form of Stepwise Regression. In this method, only a variable is added to the model (if $F > F_{in}$). The added variable is not subsequently removed from the model. It is a weaker method than Stepwise Regression since it does not check whether the variable added in the previous step will be removed in the next step.

b. Backward Elimination

The model starts by including all variables in the model. The variable with the smallest partial F value is eliminated ($F < F_{out}$). The model is re-established with k-1 variables. The partial F values are recalculated. Again if $F < F_{out}$ has the smallest partial F value, then it is eliminated. The process continues until there are no variables removed in the model. It is possible that there is a problem of multicollinearity, since it takes all variables into the model at the same time. In most

cases, the balance between the number of variables and the number of observations is not maintained.

2.2 Support Vector Machines

Support Vector Machines, introduced by Vapnik in 1963, is a statistics-based Machine Learning method proposed for the solution of the classification and regression problems, that divides the multi-dimensional data into two completely separate classes using a hyper-plane, and based on statistical learning theory. Adaptation of Support Vector Machines for regression was proposed by Smola and Schölkopf (2004), and this method has been called Support Vector Regression (Smola & Schölkopf, 2004). The fact that it is robust and it performs well has made this algorithm popular. It is suitable for small and medium-sized data sets. There are no prior information or assumptions about the distribution. Support Vector Machines are quite advantageous as they can be applied to both linear and nonlinear data, have a high accuracy rate, model complex decision boundaries, work with a large number of variables, and lack overfitting problems. However, there are disadvantages such as the inability to generate probabilistic estimates and the need for kernel functions to be positive defined continuous functions.

The goal of Support Vector Machines is to find a hyperplane in an *n*-dimensional space that distinctly classifies data points. Data points on both sides of the hyperplane that are closest to the hyperplane are referred to as Support Vectors. These affect the position and direction of the hyperplane and thus help build support vector machines.

The following Figure 2.1 is an example of hard margin support vectors. The correct maximum margin delimiter, which is in the form of a straight line in the middle, is called the margin, while the dashed lines are called the margin hyperplanes (decision boundary planes).

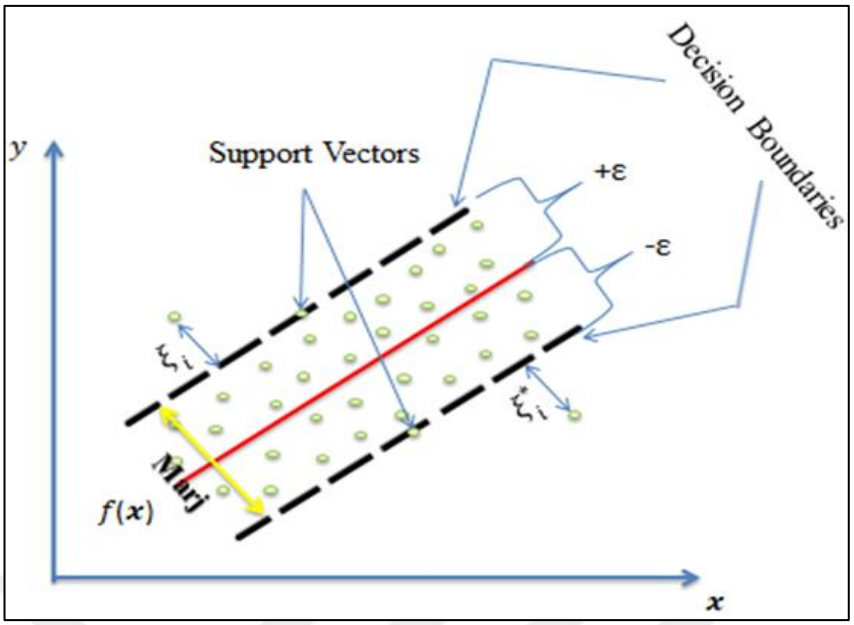

Figure 2.1 General structure of support vector machines

The important thing here is to determine which plane is better. Support Vector Regression tries to find the flatest possible function $f(\boldsymbol{x})$ with the maximum ε deviation among for actual response variable y_i defined as $y_i \in R$ for all training data. In other words, errors will not be ignored as long as they are smaller than ε, but a deviation greater than this value will not be accepted (Burges, 1998).

First, the $f(\boldsymbol{x})$ function for Support Vector Regression is expressed using the following equation.

$$f(\boldsymbol{x}) = \boldsymbol{w} \,.x + \, b \qquad (2.14)$$

Here $\boldsymbol{w}$ is the normal of multiple plane (weight vector), x is the input vector, and b is deviation (constant number). The flatness of the given equation can be achieved with $\boldsymbol{w}$ being small. For this, the norm $\|\boldsymbol{w}\|^2$ should be minimized.

Later the margin hyperplanes (decision boundary) equations are defined as below;

$$\boldsymbol{w}.x + b = +\varepsilon \quad (2.15)$$

$$\boldsymbol{w}.x + b = -\varepsilon \quad (2.16)$$

(Smola & Vishwanathan 2008).

The decision function for Linear Support Vector Regression is as follows.

$$f(\boldsymbol{x}) = \sum_{i=1}^{N}(a_i - a_i^*).\langle x_i, x\rangle + b \quad (2.17)$$

As seen, $\boldsymbol{w}$ weights can be written as as a linear combination of examples x_i. Thus, the complexity of the function represented by the support vectors will be independent of the size of the input samples and will depend only on the number of support vectors (Friedman et al., 2001).

As with many problems, it is not always possible to separate data linearly. In such cases, the problem caused by the fact that part of the training data remains on the other side of the optimal hyperplane is solved by identification of the artificial variable (ξ). The balance between maximizing the margin and minimising false function (hyperplane) errors is achieved by an arrangement parameter that takes positive values and is denoted by C (Cortes & Vapnik, 1995). The optimization problem for data that cannot be linearly distinguished using an artificial variable and an tessellation parameter is formulated below.

$$\frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*) \quad (2.18)$$

The artificial variable ξ here refers to the free variable, i.e. deviations from the hyperplanes of margin. ξ_i is the positive directional deviation, and ξ_i^* indicates negative directional deviation. C the parameter is called the regularization parameter (Smola & Vishwanathan 2008).

The maximum value of the margin between the decisional boundaries, called the remainder margin, plays an important role in determining the optimal hyperplane. In cases where there is a soft margin, data is allowed to enter the margin zone. Cases, where data is not allowed to enter the margin region, are called hard margin. Hard margin is very sensitive to outliers. For this reason, it causes wrong decision limits. This is controlled by parameter C. If C is selected small, flexibility increases, and data is found in the margin region. If C is chosen greater, the margin narrows as much as possible. The Optimum C parameter can be decided by cross-validation. In cases where the model is overfit, the number of C should be decreased.

Functions that enable the transition from linear support vector machines to nonlinear support vector machines are called kernel functions. It is one of the methods used for data sets with complex structure that cannot be separated linearly. Kernel functions convert data into a higher-dimensional feature space to enable linear separation. Although it has multiple types, the most commonly used kernel functions are: linear kernel function, polynomial kernel function, and Radial Basis Kernel Function. In this study, the three aforementioned important methods mentioned were used.

Function shown as the inner product of two observations is the general form of the kernel function;

$$K(x_i, x_{i'}). \qquad (2.19)$$

This function measures the similarity of the two observations (James et al., 2013).

For Linear Kernel quantifying the similarity of a pair of observations using the Pearson (standard) correlation, the following function is used:

$$K(x_i, x_{i'}) = \sum_{j=1}^{p} x_{ij}\, x_{i'} \qquad (2.20)$$

(James et al., 2013).

The function for Polynomial Kernel with d kernel parameters, which leads to a much more flexible decision boundary among kernel functions is as follows:

$$K(x_i, x_{i'}) = (1 + \sum_{j=1}^{p} x_{ij}\, x_{i'j})^d. \qquad (2.21)$$

Linear Kernel Function is used when $d = 1$, and the Polynomial Kernel Function is used when $d > 1$. As the degree of the polynomial increases, there is a high probability that the problem of overfitting will be encountered (James et al., 2013).

Another most commonly used kernel method is the Radial Basis Kernel Function with the following function;

$$K(x_i, x_{i'}) = \exp(-\gamma \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2) \qquad (2.22)$$

The greater the Euclidean distance of a given test observation from a training observation, the smaller the function will be. This is the normal behavior of the Radial Basis Kernel. Namely, only nearby training observations have an effect on a test observation (James et al., 2013). What the Radial Basis Kernel is trying to do is to calculate how similar each point is to a given point with the normal distribution and to estimate to which group it will be included. The width of the distribution is controlled by the gamma hyperparameter. The smaller the *Gamma* (γ), the wider the distribution. If the model has overfit, it is necessary to reduce the γ, and if the model has underfit, it is necessary to increase the γ.

The predictive performance of the regression-based model in Support Vector Machines directly affects the Support Vector Regression parameters. The epsilon value (ε), the regularization parameter (C), the kernel function type, and the parameter of the kernel function, if any, must be carefully determined. In addition, it is also very important to select the kernel function type and determine the most

appropriate parameter values of this function. Support Vector Machines do not have a rule or formula based on a mathematical model to obtain the desired values of parameters that affect prediction accuracy. To this end, an efficient search algorithm should be used to find the optimal values of the parameters used in Support Vector Machines based models, so that the highest level of model performance can be achieved (Açıkkar & Sivrikaya, 2020).

2.3 Regression Trees

Although the development of Regression Trees is based on the Automatic Interaction Detection (AID) program proposed by Morgan and Sonquist in 1963.

Regression Trees are one of the decision tree methods used to model when the dependent variable is continuous. The purpose of Regression Trees is to divide the data into homogeneous groups in an ordered format in order to maximize the differences in the dependent variable. Understandability of the decision rules used in creating a tree structure is one of the most important advantages of choosing Regression Trees.

In statistical methods, some functions can be difficult and complex to understand. Regression Trees, with their root and leaf structures, describe these complex functions as a set of rules from root to leaf after the tree is created. It simplifies even the most complex structure, making it more understandable. It is simpler than the classification tree method (Breiman et al., 2017).

Regression trees are a prediction technique in a tree view consisting of decision nodes, branches, and leaves. Tree formation starts with the entire data set. The observations that provide the condition at each node are assigned to the left branch, while the others are assigned to the right branch (Friedman et al., 2001).

When there is a sample space divided into terminal nodes by binary divisions with the response variable $Y(t)$ predicted at each terminal t node in the measurement space X, we obtain the following Figure 2.2:

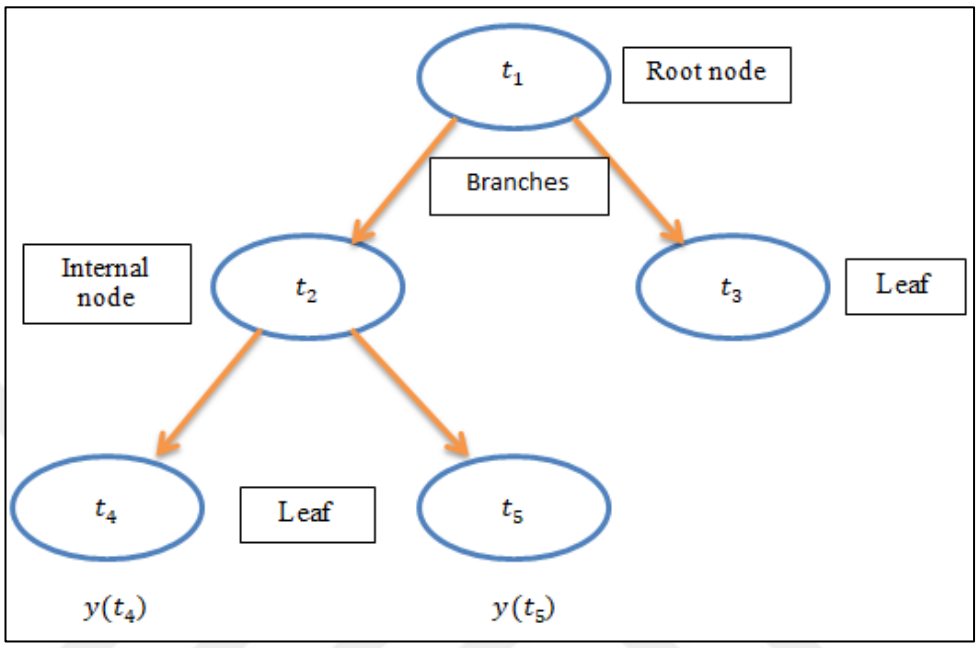

Figure 2.2 Regression trees (Breiman et al., 2017)

When creating a tree structure, decision rules are created by asking a series of questions to the data. First, the root node begins by asking a question. This process continues until it reaches the leaves, that is, until the tree branches and grows. The most important stage here is to determine which criterion or value the branching will be based on. In the Regression Tree, the criterion for minimizing the error sum of squares is used to determine the division value of the variable in which the division will be made. However, finding this value is often computationally impossible. Therefore, there must be an algorithm that automatically decides to find the best division.

Let y_i denotes the response variable $(i = 1,2,\dots,N)$ and x_i = (x_{i1} , x_{i2} , … , x_{ip}) denotes the predictors. The response variable is modelled as a constant c_m when m is the number of regions, R represents the region for each region allocated as

R_1, R_2, R_3, … , R_m. When the sum of squares is minimized c_m is obtained as the average y_i in region R_m.

$$\hat{c}_m = \text{ave}(y_i \mid x_i \in R_m) \qquad (2.23)$$

The division variable j defined for the tree that originally began to be created by taking the entire dataset and division point s are defined with the following half-plane:

$$R_1(j,s) = \{ X \mid X_J \leq s \} \text{ and } R_2(j,s) = \{ X \mid X_J > s \} \qquad (2.24)$$

Later,

$$\min_{j,s}[\min_{c_1} \sum_{x_1 \in R_1(j,s)} (y_i - c_1)^2 + \min_{c_2} \sum_{x_1 \in R_2(j,s)} (y_i - c_2)^2] \qquad (2.25)$$

division variable j and division point s are sought and internal minimization for j and s is found as follows:

$$\hat{c}_1 = \text{ave}(y_i \mid x_i \in R_1 (j,s)) \text{ and } \hat{c}_2 = \text{ave}(y_i \mid x_i \in R_2(j,s)) \qquad (2.26)$$

These operations are performed quickly and the best (j, s) pair is determined. After the best division is found, the data set is divided into two regions, and the division process is repeated in these two regions. This process continues in the same way until the process is completed (Friedman et al., 2001).

2.4 Bagging

Developed by Breiman in 1994, it is the most successful Machine Learning based Ensemble Learning Algorithm in predictive analytical studies used in solving both classification and regression problems (Breiman, 2001). This algorithm consists of a set of models combined with the resampling method. Its purpose is to increase

accuracy by combining the estimates produced by the set of models. However, in the model established, overfitting is observed when the training data set overfits to the observation data. Ensuring randomness in observation and variables, reduce overfitting and improve forecasting success are effective in the preference of this method.

Ensemble Algorithms are categorized into three as Bagging (Bootstrap Aggregating), Boosting, and Stacking (*stacked generalization*). In the Bagging method, new trees are created using the resampling method, replacing them from the data set, and a collection of trees emerges from these trees. For this reason, they are called the Ensemble Learning algorithms. It gives more efficient results than single trees.

This method uses the entire data set (N) to divide the priority data set into two parts as test and training data with simple random sampling without repetition. This ratio is determined by taking into account the size of the audience according to the literature. With simple repeated random sampling after determining the training data set, sample units are assigned to bags created from 1 to n. The model is established with the sample units in each bag. If the problem solved at the stage of combining the results obtained from the established models is a regression problem, the dependent variable is determined by averaging the results obtained. If the classification problem is solved, the dependent variable is estimated according to the majority voting results obtained from the established models (James et al., 2013). Figure 2.3 is an example of the Bagging Algorithm process.

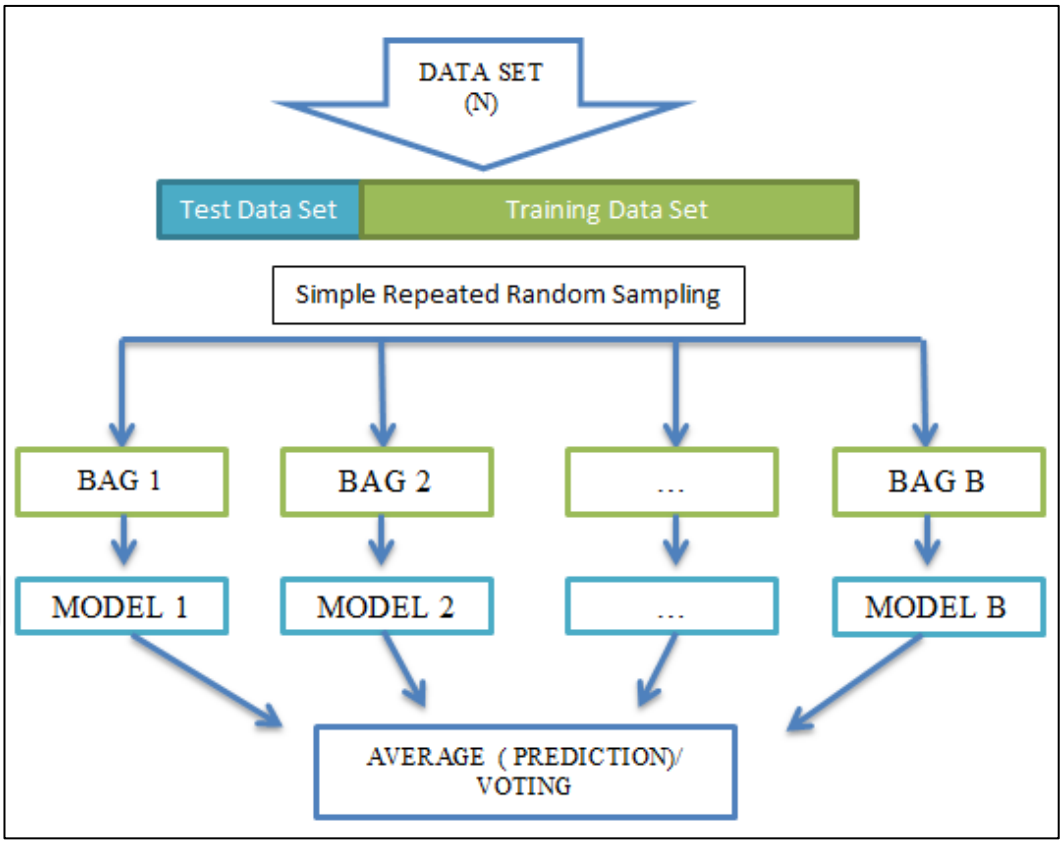

Figure 2.3 Bagging tree algorithm

Suppose estimate defined for input x is $\hat{f}(x)$. When Z = {(X_1, Y_1), (X_2, Y_2), . . . , (X_N, Y_N)} is defined as estimate $\hat{f}^{*b}(x)$ for each subset created for $b = 1,2,\dots,B$, in the defined training dataset;

$$\hat{f}_{bag}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^{*b}(x) \qquad (2.27)$$

the average of estimates is obtained in the regression problem using Equation 2.27 (Friedman et al., 2001).

The point to be considered when creating a model is to establish a balance of bias and variance. When applying bagging to regression trees, B regression tree is created using boot training kits b. Since these trees grow deep and are not pruned, the variance of each tree is high. But the bias is low. In other words, the problem of overfitting occurs. In Equation 2.27 the variance is reduced by averaging the

estimates obtained from the B regression tree. If the balance of bias and variance is achieved, a good-fit is obtained (James et al., 2013).

2.5 XGBoost (Extreme Gradient Boosting)

XGBoost, which first appeared in the article "A Scalable Tree Boosting System" published by Tianqi Chen and Carlos Guestrin in 2016, is actually a high-performance state of Gradient Boosting algorithm optimized with various regulations (Muratlar, 2020). The main reason for its high performance is that it prevents overfitting and provides bias-variance trade-offs by combining Gradient Boosting, bagging- bootstrap aggregation, and feature randomness (Noyan, 2020). In addition, the fact that it can get high predictive power, prevent over-learning, manage empty data and do it ten times faster than other popular algorithms are the reasons why it is the most preferred algorithm. One of the benefits of XGBoost is the better prediction of test error in machine learning, and that it allows the cross-validation used in model selection to be used each time the upgrade process is performed. Therefore it is being used in many studies such as store sales forecast, high energy physics event classification, web text classification, predict customer behavior, motion detection, malware classification, estimated ad clickthrough rate, and the estimation of risk in terms of product categorization (Chen & Guestrin, 2016).

The factor in XGBoost creating consecutive decision trees faster is parallelization. In addition, when the XGBoost algorithm prunes the tree, it makes splits up to the specified *max_depth* parameter value, and then starts pruning the tree backwards and removes sections that do not have any positive contribution. Thus, there is no need to go to the starting point to train the model from scratch. Assume that the number of trees was over-entered and the model was overfitting, limiting the number of trees when estimating and overfitting can be prevented.

XGBoost, and Gradient Boosting, which forms the basis of XGBoost, are also based on Boosting Algorithms. Therefore it is necessary to first explain the logic

behind the Boosting Algorithms and continue with Gradient Boosting to elaborate the XGBoost Algorithm.

Boosting, like Bagging, is one of the statistical learning methods used for both regression and classification. Boosting methods now attempt to improve optimization-based performance. A model in a series is created by building on the estimated residuals of the previous model in the series. There is a cumulative error assessment. Each tree is grown using information from previously grown trees. Thus, trees are dependent, not independent (James et al., 2013). Although Bagging and Boosting work in the same way, the difference between them is whether the trees that will be created have dependencies with the previous trees, and whether error optimization is cumulative over residuals. Examples of Boosting Algorithms are AdaBoost, Gradient Boosting Model, XGBoost, light GBM, CatBoost methods (Keskin, 2019).

Gradient Boosting primarily determines the first leaf when creating the tree. Later, the error between the prediction and the output is decoded. Considering these prediction errors, new trees are created and the tree is trained to reproduce the error created by the previous tree. This situation continues until the error between the previous output and the current estimate reaches the desired level or until no further development can be made from the model (Muratlar, 2020).

For a given example, to classify the decision rules in trees (q : $\mathbb{R}^m \rightarrow T$, $w \in \mathbb{R}^T$) given by q using leaves and to calculate the final estimate by adding scores at the relevant news in ($F = \{ f(x) = w_{q(x)} \}$ given by w, the following equation is used;

$$L(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k) \quad \text{where} \quad \Omega(f) = \gamma T + \frac{1}{2}\lambda \, || w ||^2 \qquad (2.28)$$

Here T is the number of leaves in the tree, l is the varying convex loss function measuring the difference between estimation $\hat{y}_i$ and target y_i and it measures the model's prediction on training data. Ω, on the other hand, refers to the term

regularization, which punishes the complexity of the model. This term corrects recently learned weights to prevent overetraining, i.e. eliminates complexity (Chen & Guestrin, 2016).

XGBoost starts with the first leaf and makes the base score about the weights of the samples. In a regression problem, the first estimate is the average of all data. As the algorithm works by giving weights to trees, it creates the next tree based on the errors of the previous tree the same way it did the previous tree. The performance of the estimation is examined with the model's residual. XGBoost draws a tree that predicts these errors. The number of trees and number of leaves will be determined by the researcher according to the data set and model. If the errors obtained by the first determined estimate are collected and written to the corresponding leaves, the estimate is correct, except for the average leaves; however, this leads to overfitting, that is, the established model completely memorizes the problem. Calculations to prevent overlearning, a number, defined as learning rate, is added between 0 and 1. In this way, both overlearning is prevented and the variance is reduced. If the error decreases, the tree is drawn again. This process repeats up to the specified number of trees (Noyan, 2020).

Unlike other algorithms, there are multiple parameters used for XGBoost. These parameter values were also used automatically with the R package in this study. *Max_depth*: depth of tree, *base_score*: first estimate of the first round of gradient upgrade in regression problems, *Eta*: learning rate with a value between 0 and 1, *Gamma*: parameter that prevents over-learning, *Colsample_bytree*: decoded sample rate randomly selected for each tree created, *Colsample_bynode*: sub-sample ratio of columns for each node, *min_child_weight*: the minimum sum of the sample weight (hessian) needed in a node (child) (It is the parameter that has an important role in preventing overfitting and reducing the complexity of the model. This parameter also affects the depth of the tree. Using cross-validation, the appropriate *min_child_weight* can be selected for the model.), *Subsample*: row ratio taken to create each tree, *Nrounds*: parameters that specify the number of trees in the model (Abar, 2020).

When creating a model with XGBoost, it is aimed to achieve the best results of performance criteria by selecting the optimal values of parameters.

2.6 Random Forests

Random Forests is an Ensemble Learning Method proposed by Ho in 1998, and created by combining the Bagging method developed by Breiman in 1996 and the Random Subspace method used to randomly select a subset of properties to be used to grow each tree. A paper on written character recognition, published by Amit and Geman in 1997, describing a large number of geometric features and looking for a random selection of them for the best cleavage at each node, was also instrumental in developing this method (Breiman, 2001).

Random Forests is a versatile and intelligent Machine Learning Method based on Decision Tree Algorithms that can be used for both classification and regression problems. Random Forests creates a forest by combining multiple singular decision trees to produce a more accurate and stable prediction. It can handle large amounts of data sets of high volumes. It is known as an important method of dimensionality reduction, as it can handle more than a few thousand variables and well identify the important ones among them. Outlier value is an expert solution for addressing performing other data discovery steps while removing missing value issues and easily maintaining accuracy. It is an advanced version of the Bagging Method. In addition to its advantages, the problem of overfitting can be observed when there is too much noise in the sample data. Since the performance of the model cannot be controlled, it can act as a black box approach for statistical models (Sullivan, 2017). Figure 2.4 shows an example of Random Forests Algorithm.

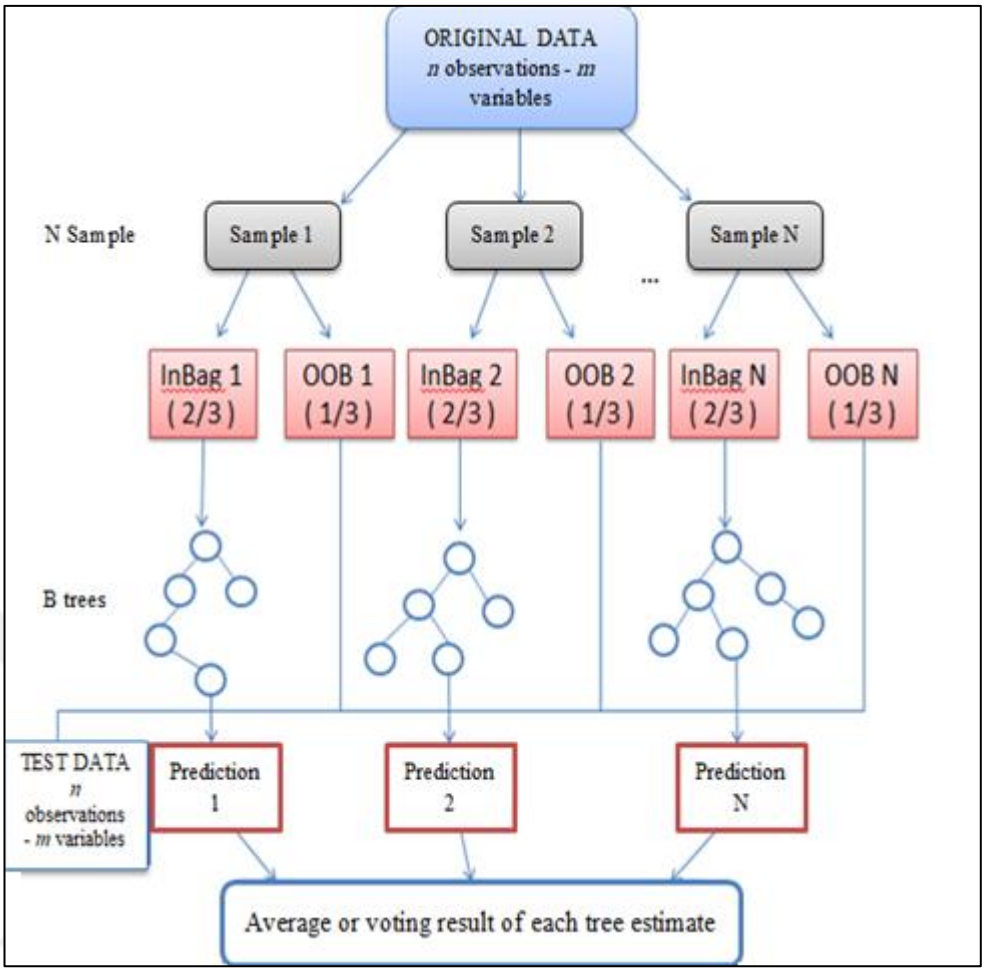

Figure 2.4 Random forests algorithm

The Random Forests Algorithm is based on two parameters. These parameters; the number of trees to be developed (B) and the number of variables to use on each node (m). If the original dataset is not its own test set when creating each tree, the same number of trees (B) are created using the bootstrap method with the sample count (N). 2/3 of the training data set separated from the original data set when creating the decision forest is the inbag data set used to create the tree, and 1/3 is the of the Out of Bag (OOB) data set used to calculate the error rate of the established model (James et al., 2013). For each tree in the decision forest, each dataset from the original dataset is different from each other. The inbag and out of bag datasets of each tree are also different. Thus, in the process of growing trees in the forest, each tree is grown independently of each other without being affected by the others (Ercire, 2019).

After this stage, the other parameter that needs to be determined is the one that performs the best division, giving the best information among the variables randomly selected from the learning dataset to create the tree (Korkem, 2013). m estimator variables among p variables are selected randomly for the division of each node, when creating a tree. This value is proposed as $m = p/3$ for regression trees, as $m = \sqrt{p}$ for classification trees. p refers to the number of variables. Here the condition $m < p$ must be met. Because overgrowth of the tree and overrfitting of the tree are not desirable. The tree is divided into branches from the one with the highest gain of knowledge from this m predictor and the one with the best division (Friedman et al., 2001). The criteria of the Classification and Regression Tree (CART) Algorithm, a classical decision tree method, are used to determine depending on which value of the determined variable the tree should branch (Breiman, 2001). This criterion is the gini index for classification trees. However the distinctions in the Regression Tree are not carried out according to the gini index criterion, but according to the "algorithm for reducing squared residuals", which means that the total variance estimated for the resulting two nodes must be minimized (Özkan, 2012). Instead of searching for the variable that will provide the best branching in each node, the variable that will provide the best branching is randomly selected. The purpose of randomness is to minimize correlation between trees and decrease error rate. A decrease in the error rate improves the performance of the algorithm and makes the algorithm strong against over-learning. Also not pruning the tree makes random forests more advantageous than other decision tree methods (Ekelik & Altas, 2019).

The above operations are repeated at each node (B) times, until the leaf node is obtained i.e. (Sullivan, 2017). For a new data class estimate, the estimates made by the B tree separately are combined and the final estimate is predicted by taking the average of the estimates made for the Regression Trees and the class with the most votes for the Classification Trees. This estimate is determined by the Out of Bag error rate (Friedman et al., 2001).

All trees created with the Inbag data set are tested with the resulting Out of Bag data and the error rate is calculated by averaging these error rates and the Out of Bag error of the decision forest is calculated (James et al., 2013). This error depends on the individual strengths of each tree (each tree's own error rate) and decency between them. The weights given to every tree in inverse proportion with the Out of Bag error rate are called estimates. The average of these estimates is effective in determining the class. One of the important factors in choosing Random Forests is that it can calculate its own error rate (Breiman, 2001).

2.7 k-Nearest Neighbor-kNN

The k-Nearest Neighbors Algorithm was first introduced by Fix and Hodges in 1951 as a nonparametric method for use in pattern (model) recognition, and was later developed by Cover and Hart in 1967 (Elasan, 2019). This method is a nonparametric Supervised Machine Learning Algorithm used in both classification and regression problems, which determines the class in which the observations will take place and the nearest neighbor by the *k* neighbors value. Unlike other Supervised Learning Algorithms, it does not have a training stage. It does not learn training data, but memorizes it. This feature also shows that it is a lazy type of learning. It learns complex target functions quickly without losing information. For this reason, it is one of the simplest and most used methods (Goyal et al., 2014). Modern applications of the k-Nearest Neighbors Algorithm include suggestion systems, text categorization, heart disease classification, and financial market forecasting (Anava & Levy, 2016).

The goal of the k-Nearest Neighbors Algorithm is to predict which class the new example will be included in according to the proximity relationship with the earlier data on the existing learning data when a new sample arrives. The algorithm decides this by looking at the k-Nearest Neighbors for the new example. In classification problems, this estimate is determined by a majority vote among the nearest neighbors, while in regression problems, it is the average of the numerical values of

the nearest neighbors decisively. In this way, the new instance is labeled (Friedman et al., 2001).

In k-Nearest Neighbors Regression, first the k neighbors, which are closest to x_0 is determined, and these neighbors are defined as N_i. The successful implementation of this algorithm depends on careful determination of the three parameters. These parameters are the number of nearest neighbors k, the weight vector α, and the distance metric (Anava & Levy, 2016).

First of the algorithm steps is to determine parameter k. The optimal value for k is determined by the equilibrium of bias and variance. Approaches used to estimate test rates can be used to define the most optimal k value in k-Nearest Neighbors Regression (James et al., 2013). Cross-validation can also be proposed as another method. Cross-validation determines a good value of k retrospectively using an independent dataset. In general, a k value between 3 and 10 is optimal (Wettschereck & Dietterich, 1994; Sun & Huang, 2010).

Second step is the determination of new observation x_0 to be added in the sample data set and calculate all the distances between the available data and the new observation. There are several functions used to determine the distances. The k-Nearest Neighbors Regression uses the same distance functions as the k-Nearest Neighbors Classification. These funcions include the commonly used Euclidean function and Manhattan**,** Minowski, and Hamming functions. Hamming is used when there are categorical variables, and the other three functions are used when there are continuous variables. In addition, when there is a mixture of numeric and categoric variables in the dataset, standardization of numeric variables between 0 and 1 is required (Alkış, 2017).

Another approach to be used in distance calculation is the application of the weight vector. It is one of the methods that is effective in the performance of the algorithm. Conducting experiments on the data set with different k values, which are used most frequently for determining the number of k neighbors as mentioned above**,**

and selecting the value that is observed to be the most successful causes a time loss. In order to determine the optimal k value by preventing time loss, a method of weighting distances has been developed. This operation is performed by weighing k instances, nearest to the test data, regarding their distances, to determine the the most successful k value in the algorithm (Alkış, 2017). These weights, which use a density function, are called "Kernel methods" (Goyal et al., 2014).

After the parameters are determined, using the average of all training responses in N_i nearest to x_0, which has the number of k neighbors and the function $f(x_0)$ in Equation 2.29 below, labelling is done by estimating to which set the new observation x_0 belongs to (James et al., 2013)

$$f(x_0) = \frac{1}{k} \sum_i N_i. \qquad (2.29)$$

Finally, although it is a simple algorithm, it is a distance-dependent machine learning method and it is not clear which distance criterion to use (Alkış, 2017).

CHAPTER 3
MODEL EVALUATION AND PERFORMANCE CRITERIA

3.1 K-Fold Cross-Validation

In data mining studies, K-Fold Cross-Validation is used to test the success of the applied method. It is one of the methods of disassembling the dataset to evaluate prediction models and train the model. In cross-validation, the data set is divided into two parts in a certain proportion as training and test, and after the system is trained with the training set, its success is tested with the test set.

In the folding cross-validation method, first of all, a K value is decided. K value is usually preferred as 5 or 10. After determining K value the data is divided into equal parts depending to the number of K. K-1 groups are used to train model and the remaining group is used for testing, and the sum of squares predicted for the error is obtained. This process is repeated for K times. The K estimates of the prediction error are combined to produce a K-Fold Cross-Validation estimate. The average of the combined values is calculated (James et al., 2013).

$$CV_{(K)} = \frac{1}{K} \sum_{1}^{K} MSE_i \qquad (3.1)$$

3.2 Model Performance Criteria

3.2.1 Coefficient of Determination (R^2)

The coefficient of determination is the value that indicates how much change in the dependent variable is explained by the independent variables. In other words, it measures how much of the change in the dependent variable is explained by the model.

$$R^2 = \frac{SSR}{SST} = 1 - \frac{SSE}{SST} \qquad (3.2)$$

Here SST(Sum Squares of Total) refers to change in the dependent variable without taking into account independent variables, SSR(Sum Squares of Regression) to the change described by the model, SSE(Sum Squares of Error) to the change that cannot be explained by the model. Coefficient of determinations receieves a value between 0 and 1. $0 \leq R^2 \leq 1$. R^2 being near 1 means that most of the variability in the dependent variable is explained by the regression model.

3.2.2 Mean Square Error (*MSE*)

Mean Square Error is obtained by dividing the sum of squares of the differences between each real value and the estimated value corresponding to that value to the sample size. Although there is no power of interpretation alone, it helps to choose the best model by comparing several studied models. It is always a positive number. When it is close to zero, it means that estimators perform better.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 = \frac{1}{N}\sum_{i=1}^{N} {e_i}^2 \qquad (3.3)$$

Here; N is the sample size, y_i: is the true value, $\hat{y}_i$ is the estimated value and e_i is defined as the difference between the actual value and the estimated value.

3.2.3 Root Mean Squared Error (*RMSE*)

Root Mean Squared Error, is the square root of the Mean Square Error. The reason for frequent preference of this criterion is that Mean Square Error is sometimes too large to be easily compared and is hypersensitive to outliers. Root Mean Squared Error always gets a positive value. A value close to zero indicates that it performs better.

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} {e_i}^2} \qquad (3.4)$$

3.2.4 Mean Absolute Error (*MAE*)

Mean Absolute Error is the average of the absolute values of the differences between each real value and the estimated value corresponding to that value. Mean Absolute Error also gets a value greater than zero. A value close to zero indicates that it performs better. The mean absolute error is more robust against outliers because it does not use the squared expression.

$$MAE = \frac{1}{N}\sum_{i=1}^{N} |y_i - \hat{y}_i| \quad (3.5)$$

CHAPTER 4
APPLICATION

In this study, Hourly Radon Levels (bq) was obtained as response variable from the Seferihisar region in hourly periods between 30 October 2006 and 04 June 2007. Corresponding to the Radon levels, the measurements for the parameters of Hourly Actual Pressure (hPa), Hourly 50 cm Soil Temperature (°C), Hourly Relative Humidity (%), Hourly Temperature(°C), Hourly Wind Degree (°), Hourly Wind Speed (m/sec) and Wind Direction were obtained as predictors from the Republic of Turkey Ministry of Agriculture and Forestry, General Directorate of Meteorology. The obtained data set consists of 5232 observations and 8 variables. R statistical programming language and required packages were used for analysis.

4.1 Data Preprocessing

The data set was gathered and created in Excel. It was imported from Excel to R with the readxl package, as in Figure 4.1. In terms of reproducibility `set.seed()` function was used of the beginning of the study.

```
set.seed(80)

library(readxl)

setwd("C:/Users/Çağla/Desktop")

radon_data <- read_excel("radon_data_set.xlsx")
```

Figure 4.1 Importing the data set in R software

The following Table 4.1 shows the first ten observations of the dataset. The Radon variable contained in the data set is a dependent (response) variable, while the actual pressure, relative humidity, temperature, soil temperature, wind degree, wind speed and wind direction variables are independent variables (predictors).

Table 4.1 First 10 observations of the data set

	radon	actpress	r.humidity	temp	soiltemp	w.degree	w.speed	w.direction
1	0	1005.2	96	12.2	19.4	29	1.2	N
2	0	1004.8	96	12.6	19.4	36	1.3	N
3	0	1004.4	96	12.1	19.4	36	1.3	N
4	0	1004.3	96	12.2	19.4	14	0.5	N
5	0	1004.2	96	12.7	19.4	0	0	C
6	0	1004.2	96	12.8	19.4	0	0	C
7	0	1004.3	96	14.1	19.4	0	0	C
8	0	1004.6	96	17.0	19.4	0	0	C
9	0	1004.1	96	17.2	19.4	0	0	C
10	0	1004.0	96	17.9	19.4	122	2.1	E

Checking the structure of the variables shows that the data types of the wind speed and wind direction were misread. Wind speed must be numerical, and wind direction must be of factor type.

Table 4.2 Data types of variables

```
radon       : chr [1:5232] "0" "0" "0" "0" ...
actpress    : num [1:5232] 1005 1005 1004 1004 1004 ...
r.humidity  : num [1:5232] 96 96 96 96 96 96 96 96 96 96 ...
temp        : num [1:5232] 12.2 12.6 12.1 12.2 12.7 12.8 14.1 17 17.2 17.9 ...
soiltemp    : num [1:5232] 19.4 19.4 19.4 19.4 19.4 19.4 19.4 19.4 19.4 19.4 ...
w.degree    : chr [1:5232] "29" "36" "36" "14" ...
w.speed     : chr [1:5232] "1.2" "1.3" "1.3" "0.5" ...
w.direction : chr [1:5232] "N" "N" "N" "N" ...
```

Figure 4.2 shows the coercion functions to set the correct data types.

```
radon_data$radon <- as.numeric(radon_data$radon)
radon_data $ w.degree <- as.numeric(radon_data$w.degree)
radon_data$ w.speed <- as.numeric(radon_data$w.speed)
radon_data$ w.direction <- as.factor(radon_data$w.direction)
str(radon_data)
```

Figure 4.2 Conversion of the data types of the variables

Table 4.3 shows the corrected version of the data types.

Table 4.3 Corrected data types

```
radon       : num [1:5232] 0 0 0 0 0 0 0 0 0 0 ...
actpress    : num [1:5232] 1005 1005 1004 1004 1004 ...
r.humidity  : num [1:5232] 96 96 96 96 96 96 96 96 96 96 ...
temp        : num [1:5232] 12.2 12.6 12.1 12.2 12.7 12.8 14.1 17 17.2 17.9 ...
soiltemp    : num [1:5232] 19.4 19.4 19.4 19.4 19.4 19.4 19.4 19.4 19.4 19.4 ...
w.degree    : num [1:5232] 29 36 36 14 0 0 0 0 0 122 ...
w.speed     : num [1:5232] 1.2 1.3 1.3 0.5 0 0 0 0 0 2.1 ...
w.direction : Factor w/ 5 levels "C","E","N","S",..: 3 3 3 3 1 1 1 1 1 2 ...
```

Table 4.4 shows the summary statistics of the variables.

Table 4.4 Summary statistics of variables

```
     radon          actpress      r.humidity         temp       
 Min.   :  0.0   Min.   : 997   Min.   :14.00   Min.   :-3.70  
 1st Qu.: 73.0   1st Qu.:1010   1st Qu.:54.00   1st Qu.: 8.90  
 Median :116.0   Median :1015   Median :68.00   Median :13.10  
 Mean   :138.4   Mean   :1016   Mean   :66.91   Mean   :13.05  
 3rd Qu.:185.0   3rd Qu.:1020   3rd Qu.:82.00   3rd Qu.:16.70  
 Max.   :436.0   Max.   :1034   Max.   :96.00   Max.   :33.60  
 NA's   :949     NA's   :872    NA's   :178     NA's   :184    

    soiltemp        w.degree       w.speed      w.direction 
 Min.   : 8.10   Min.   :  0.0   Min.   :0.000   C   :  68  
 1st Qu.:12.00   1st Qu.: 28.0   1st Qu.:1.000   E   : 807  
 Median :13.60   Median :150.0   Median :1.700   N   :2806  
 Mean   :14.46   Mean   :168.8   Mean   :2.012   S   : 992  
 3rd Qu.:16.40   3rd Qu.:319.0   3rd Qu.:2.800   W   : 490  
 Max.   :23.70   Max.   :360.0   Max.   :9.300   NA's:  69  
 NA's   :210     NA's   :263     NA's   :76                 
```

Each variable in the dataset includes missing values. 949 observations for the radon variable, 872 for the current pressure, 178 for the relative humidity, 184 for the temperature variable, 210 for the soil temperature, 263 for the wind degree, 76 for the wind speed, and finally 69 for the wind direction were NA.

As stated in Section 1.2, detecting loss missing observations is quite important, and these missing observations can be corrected by appropriate methods. Since the Radon is a dependent variable and the main goal is to estimate the values of this variable in this study, the missing observation row of the Radon variable was removed from all the data in order not to obtain biased results in estimation. After this process, the current number of observations decreased to 4283.

```
radon_new <- na.omit(radon_data$radon)
actpress_new <- radon_data$actpress[!is.na(radon_data$radon)]
r.humidity_new <- radon_data$r.humidity[!is.na(radon_data$radon)]
temp_new <- radon_data$temp[!is.na(radon_data$radon)]
soiltemp_new <- radon_data$soiltemp[!is.na(radon_data$radon)]
w.degree_new <- radon_data$w.degree[!is.na(radon_data$radon)]
w.speed_new <- radon_data$w.speed[!is.na(radon_data$radon)]
w.direction_new <- radon_data$w.direction[!is.na(radon_data$radon)]
radon_data <- data.frame(radon = radon_new, actpress = actpress_new,
                         r.humidity = r.humidity_new, temp = temp_new,
                         soiltemp = soiltemp_new, w.degree = w.degree_new,
                         w.speed = w.speed_new, w.direction = w.direction_new)

summary(radon_data)
dim(radon_data)
```

Figure 4.3 Removing rows for missing observations of the radon variable corresponding to other variables

In Figure 4.3, the missing values of Radon were removed from the data set. Table 4.5 gives the summary statistics after cleaning the missing values.

Table 4.5 Summary statistics after removed the missing values of Radon

radon	actpress	r.humidity	temp
Min. : 0.0	Min. : 997	Min. :14.00	Min. :-3.70
1st Qu.: 73.0	1st Qu.:1010	1st Qu.:52.00	1st Qu.: 9.20
Median :116.0	Median :1015	Median :67.00	Median :13.50
Mean :138.4	Mean :1016	Mean :66.49	Mean :13.54
3rd Qu.:185.0	3rd Qu.:1020	3rd Qu.:82.00	3rd Qu.:17.30
Max. :436.0	Max. :1034	Max. :96.00	Max. :33.60
	NA's :576	NA's :164	NA's :153

soiltemp	w.degree	w.speed	w.direction
Min. : 8.10	Min. : 0.0	Min. :0.000	C : 67
1st Qu.:12.50	1st Qu.: 26.0	1st Qu.:0.900	E : 595
Median :14.20	Median :157.0	Median :1.700	N :2410
Mean :15.19	Mean :172.5	Mean :1.943	S : 727
3rd Qu.:17.98	3rd Qu.:331.0	3rd Qu.:2.700	W : 416
Max. :23.70	Max. :360.0	Max. :9.300	NA's: 68
NA's :169	NA's :238	NA's :75	

`mice` package was used for solving the missing data, as shown in Figure 4.4. This package estimates missing data by iteration using multiple imputation methods. The default imputation method in this package is PMM (Predictive Mean Matching).

Since it is not possible to determine the average for a categorical variable, it assigns the best instead of the missing value, in the same way, by generating *m* values. Here it has assigned five categories in the wind direction with regard to their number.

```
Library(mice)
imp <- mice(radon_data)
imp$imp
complete(imp)
radon_data <- complete(imp)
```

Figure 4.4 Assigning new values by imputation method instead to missing values

Table 4.6 presents summary statistics after completing missing values.

Table 4.6 Summary statistics after completing missing values

radon	actpress	r.humidity	temp
Min. : 0.0	Min. : 997	Min. :14.00	Min. :-3.70
1st Qu.: 73.0	1st Qu.:1010	1st Qu.:53.00	1st Qu.: 9.10
Median :116.0	Median :1015	Median :67.00	Median :13.50
Mean :138.4	Mean :1016	Mean :66.57	Mean :13.52
3rd Qu.:185.0	3rd Qu.:1021	3rd Qu.:82.00	3rd Qu.:17.30
Max. :436.0	Max. :1034	Max. :96.00	Max. :33.60

soiltemp	w.degree	w.speed	w.direction
Min. : 8.10	Min. : 0.0	Min. :0.000	C: 68
1st Qu.:12.50	1st Qu.: 26.0	1st Qu.:0.900	E: 607
Median :14.20	Median :156.0	Median :1.700	N:2456
Mean :15.15	Mean :172.1	Mean :1.942	S: 732
3rd Qu.:17.90	3rd Qu.:332.0	3rd Qu.:2.700	W: 420
Max. :23.70	Max. :360.0	Max. :9.300	

The values of the radon variable contained in the data set vary between 0 and 436 Bq. The median value is smaller than the average value. It is understood that the radon variable has a right-skewed distribution. The median of the actual pressure variable in the value range of 997 hPa to 1034 hPa is 1015 hPa, the average is 1016 hPa, The first quarters is 1010 hPa and the 3rd quarter is 1021 hPa. The values suggest that the distribution of the actual pressure is normal. The same interpretations can be made for distributions of temperature and soil temperature variables. The

wind direction variable consists of five categories: center, east, west, north, and south.

Transformation is also one of the important data preprocessing step, so the histogram was examined to see the distribution of the dependent variable, radon, to investigate the normality assumption.

According to Figure 4.5, it is seen that the distribution of the radon variable is skewed to the right. In order to eliminate this skewness, the transformation must be applied on the dependent variable.

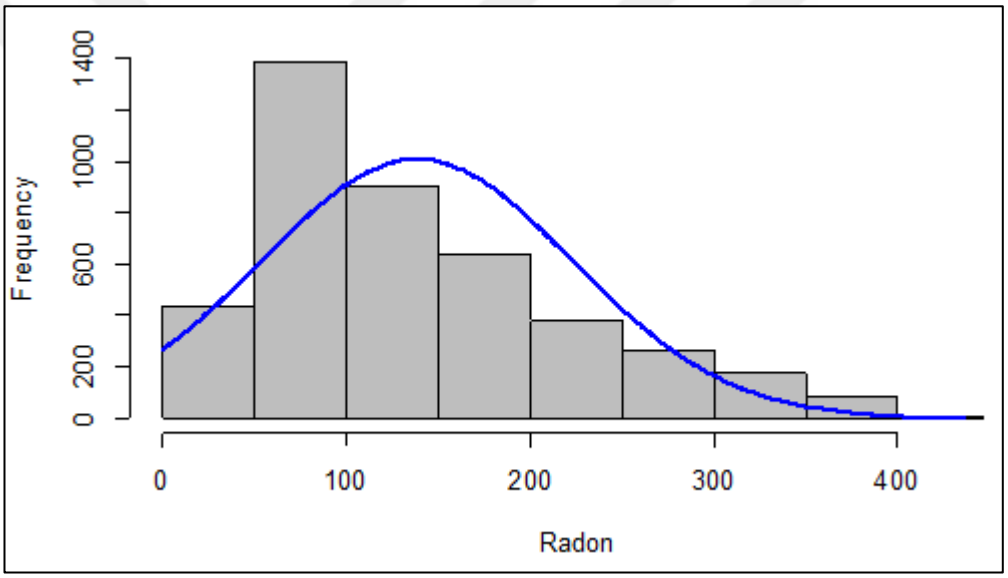

Figure 4.5 Histogram of radon

Since the Radon variable contains 0 values, `yeo.Johnson` function in the `VGAM` package in R is applied, Figure 4.6.

```
library(VGAM)
yeo_radon <- yeo.johnson(radon_data$radon, lambda = 0.5)
par(mfrow=c(1,1))
plotNormalHistogram(yeo_radon,main="Radon Histogram")
```

Figure 4.6 Transformation on radon variable and drawing histogram for radon

After transformation on the radon variable, the histogram of radon is shown in Figure 4.7.

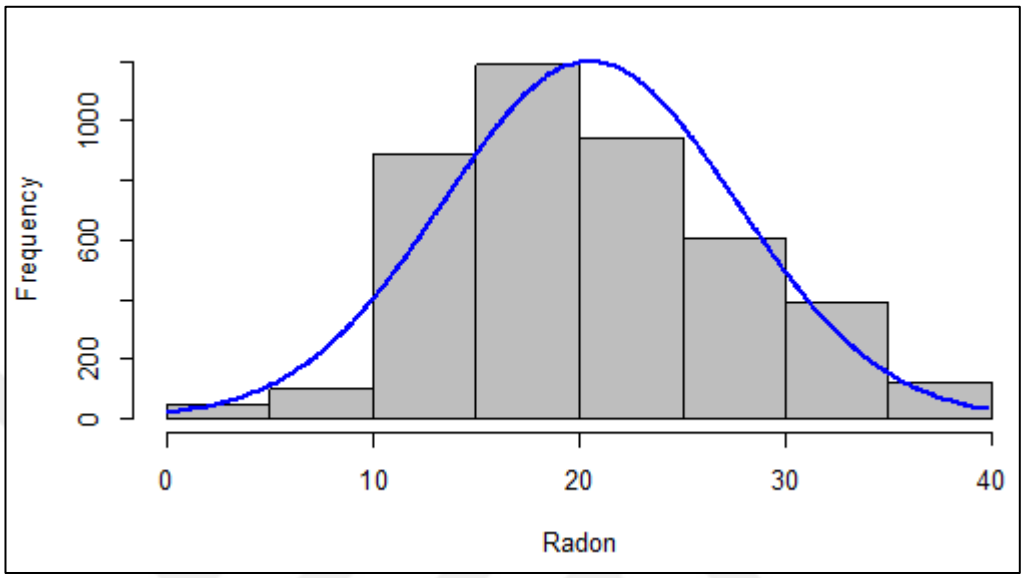

Figure 4.7 Histogram for radon after transformation

It seems that the distribution after the transformation is closer to normal. However, to achieve an objective result, the Anderson-Darling normality test in the `library(nortest)` is performed in Figure 4.8 and Figure 4.9.

H_0 : The radon variable is normally distributed.

H_1 : The radon variable is not distributed normally.

```
library(nortest)
ad.test(yeo_radon)
boxplot(yeo_radon)
```

Figure 4.8 Anderson-Darling normality test for radon

```
        Anderson-Darling normality test

data:  yeo_radon
A = 28.249, p-value < 2.2e-16
```

Figure 4.9 Anderson-Darling normality test result for radon

According to Figure 4.9, H_0 is rejected ($p < \alpha = 0.05$). The assumption of normality has not been achieved. Nevertheless, taking into account the effect on the distribution, the original Radon values in the analysis were transformed with `yeo.Johnson`, and the analysis continued with transformed Radon data.

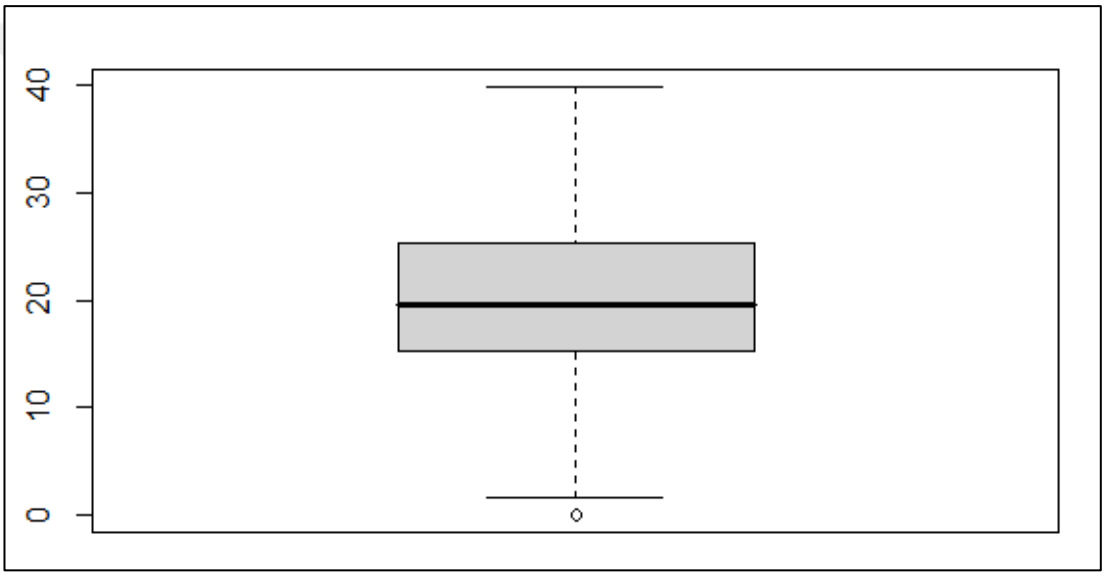

Figure 4.10 Box plot of transformed radon

The box plot of the transformed radon variable in Figure 4.10 shows that there are some outlier observations in Radon. Data preprocessing steps continue with the detection and removal stage of outlier values. Outlier values are removed from the data set 2.5% from the top and 2.5% from the bottom using the library(chemometrics) as shown in Figure 4.11.

```
library(chemometrics)
md <- Moutlier(radon_data_new[, 1:7])
ad <- which(md$rd > md$cutoff)
radon_data_new <- radon_data_new[-ad, ]
dim(radon_data_new)
```

Figure 4.11 Removing outliers from the data set

After removing outliers, the number of observations was 3944. In the final form of the data set, the Anderson-Darling normality test was performed, but it was seen that the distribution was still not normal. However, it was observed that the distribution is close to normal when looking at the histogram and box plot.

4.2 Multiple Linear Regression Analysis

The assumptions which are given in Section 2.1 have to be checked. Relationships between dependent and independent variables are examined. According to Figure 4.12, there is a strong inverse linear relationship between radon and actual pressure, a strong positive relationship between radon and temperature, and between radon and soil temperature. In addition, there is no significant relationship between radon and wind speed, and radon and wind degree. The `library(olsrr)` was used for operations used in multiple linear regression analysis.

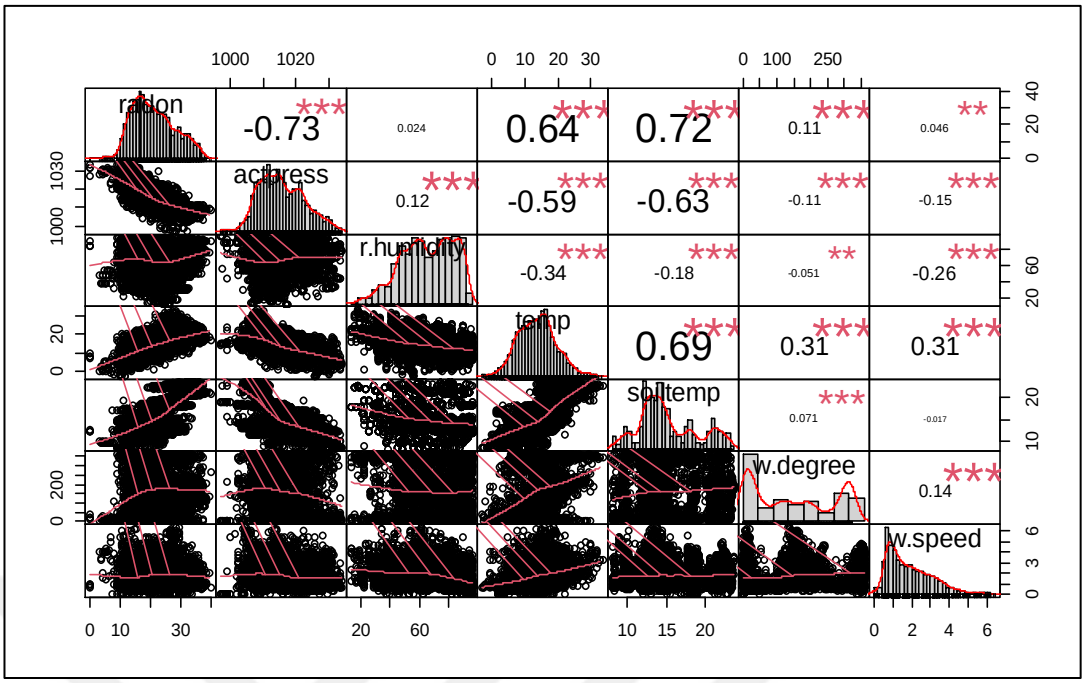

Figure 4.12 Correlation chart

Figure 4.13 is also another visulation of relationship between variables. When the relationships between independent variables are examined over this charts, it is seen that the relationships between actual pressure and temperature, and soil temperature, and temperature and soil temperature, are strong. VIF values were examined to determine whether there is multicollinearity problems.

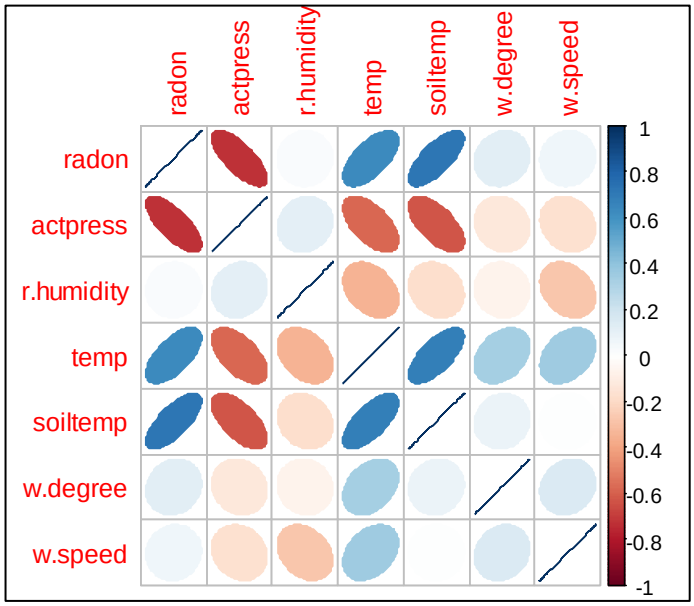

Figure 4.13 A chart of the degree and direction of the relationship between variables

A full model containing all variables has been established to examine VIF values. The codes and results are given in Figure 4.14 and Table 4.7, respectively.

```
model=lm(radon_data_new$radon~radon_data_new$actpress
+radon_data_new$r.humidity+radon_data_new$temp+
radon_data_new$soiltemp+radon_data_new$w.speed+
radon_data_new$w.degree+radon_data_new$w.direction)


ols_coll_diag(model)
```

Figure 4.14 VIF values for full model

Table 4.7 VIF values for full model

```
Tolerance and Variance Inflation Factor
---------------------------------------
                     Variables  Tolerance       VIF
1      radon_data_new$actpress 0.49604077  2.015963
2    radon_data_new$r.humidity 0.79385485  1.259676
3          radon_data_new$temp 0.29022658  3.445584
4      radon_data_new$soiltemp 0.35593605  2.809493
5       radon_data_new$w.speed 0.69394265  1.441041
6      radon_data_new$w.degree 0.75830019  1.318739
7  radon_data_new$w.directionE 0.07683384 13.015098
8  radon_data_new$w.directionN 0.04043797 24.729234
9  radon_data_new$w.directionS 0.06513491 15.352751
10 radon_data_new$w.directionW 0.10334814  9.676033
```

VIF values between numerical predictors are smaller than 5, so there is no multicollinearity problem (VIF cannot be used with categorical variable).

After all data preprocessing operations, the residual graphs of the model were examined in Figure 4.15 to investigate the validation of the assumptions.

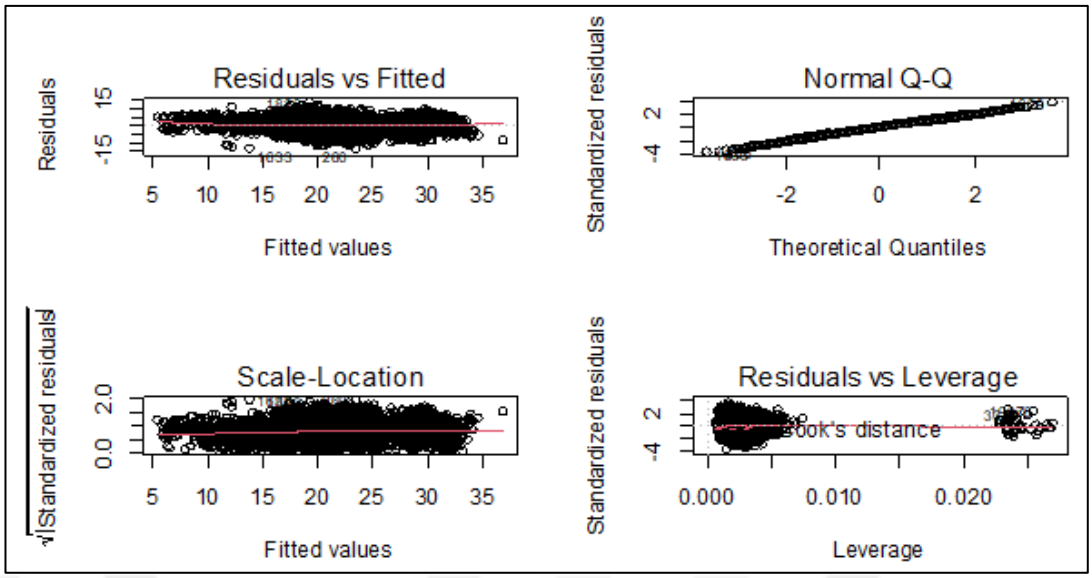

Figure 4.15 Residual charts for the full model obtained after data preprocessing operations

In the first chart, it is examined whether there is variance homogeneity. What is expected here is a random distribution of points around 0. In the chart, it has been suspected that constant variance cannot be achieved, and there may be a problem of increasing variance. In order to get more accurate results, the Breush-Pagan test in Table 4.8 was applied to control variance homogeneity.

Table 4.8 Breush-Pagan test for variance homogeneity

```
Breusch Pagan Test for Heteroskedasticity
-----------------------------------------
Ho: the variance is constant
Ha: the variance is not constant

                     Data
-------------------------------------------------
Response : radon_data_new$radon
Variables: fitted values of radon_data_new$radon

        Test Summary
-------------------------------
DF            =    1
Chi2          =    37.3540
Prob > Chi2   =    9.851875e-10
```

H_0 : Errors have fixed variance.

H_1 : Errors do not have fixed variance.

According Table 4.8, H_0 is rejected ($p < \alpha = 0.05$). Variance homogeneity has not been achieved.

The second chart specifies whether the residuals are normally distributed. It is expected that points are on a line. Here it can be interpreted that it is visually distributed close to normal. There is not much opening in the tails. A more accurate result can be obtained by performing a normality test. The test for this is shown in Figure 4.16.

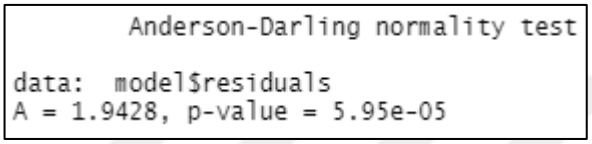

Figure 4.16 Anderson-Darling normality test result

H_0 : Error are normally distributed.

H_1 : Errors are not distributed normally.

According Figure 4.16, H_0 is rejected ($p < \alpha = 0.05$). Normality was still not achieved when the Anderson-Darling test was performed.

The third and the fourth charts in Figure 4.15 are using for detecting the outliers and influentials observations. It is clearly seen that there were some outliers and influentials observations in the model.

Table 4.9 were examined for the complete model created on the data set obtained after performing missing observation completion, transformation, outlier value cleanup operations, the coefficients of all other variables except the wind degree were significant (H_0 is rejected, $p < \alpha = 0.05$). The model is useful for predicting response variable (F=995, $p < \alpha = 0.05$). R^2 was found 71.67%. It means that 71.67% of the sample variation in Radon can be explained by the model.

Table 4.9 Summary results for the full model

```
Residuals:
     Min       1Q   Median       3Q      Max
-13.8553  -2.3270   0.0173   2.4679  13.1092

Coefficients:
                            Estimate Std. Error t value Pr(>|t|)
(Intercept)                3.943e+02  1.277e+01  30.876  < 2e-16 ***
radon_data_new$actpress   -3.890e-01  1.243e-02 -31.305  < 2e-16 ***
radon_data_new$r.humidity  6.374e-02  3.451e-03  18.467  < 2e-16 ***
radon_data_new$temp        2.374e-01  1.895e-02  12.530  < 2e-16 ***
radon_data_new$soiltemp    6.496e-01  2.563e-02  25.343  < 2e-16 ***
radon_data_new$w.speed    -1.871e-01  5.983e-02  -3.128  0.00177 **
radon_data_new$w.degree   -1.247e-04  5.058e-04  -0.246  0.80533
radon_data_new$w.directionE  5.890e+00  5.871e-01  10.031  < 2e-16 ***
radon_data_new$w.directionN  3.990e+00  5.861e-01   6.808 1.14e-11 ***
radon_data_new$w.directionS  5.959e+00  5.976e-01   9.972  < 2e-16 ***
radon_data_new$w.directionW  4.376e+00  6.113e-01   7.158 9.70e-13 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.676 on 3933 degrees of freedom
Multiple R-squared:  0.7167,    Adjusted R-squared:  0.716
F-statistic:   995 on 10 and 3933 DF,  p-value: < 2.2e-16
```

The best possible subset selection, forward selection, and backward elimination methods are used when deciding the variables that will be included in the model. From these methods, the variable selection process was started with the best possible subset selection method.

```
a <- ols_step_all_possible(model)
a


largest <- max(a$adjr)
largest
```

Figure 4.17 The best possible subset selection

As a result of this function in Figure 4.17, 127 sub-models possible were derived from the current model. Some of these sub-models are shown in Table 4.10.

Table 4.10 Examples of possible submodels obtained by the best possible subset selection method

	R-Square	Adj. R-Square	Mallow's Cp		R-Square	Adj. R-Square	Mallow's Cp
1	0.5358267285	0.535708978	2503.938755	71	0.5508494600	0.550393354	2301.383958
4	0.5234516446	0.523330755	2675.737278	88	0.5198264400	0.518972473	2732.064597
3	0.4094594951	0.409309688	4258.246385	89	0.5128358557	0.511969456	2829.112186
7	0.1735907171	0.172751510	7532.721788	87	0.4992221222	0.498713589	3018.106406
6	0.0126804336	0.012429972	9766.577280	97	0.4867301894	0.485817362	3191.527097
5	0.0021522417	0.001899109	9912.736112	93	0.1898624381	0.188421645	7312.827752
2	0.0005558614	0.000302324	9934.898046	101	0.7159888484	0.715411443	10.817432
10	0.6482789574	0.648100464	944.807770	107	0.7052881945	0.704689035	159.370488
25	0.6023534092	0.601848525	1582.374469	106	0.7041743044	0.703572880	174.834196
9	0.6005853440	0.600382647	1606.919837	99	0.6977356770	0.697351898	264.219180
13	0.5694424631	0.568895793	2039.264458	100	0.6973523735	0.696968108	269.540436
19	0.5603904988	0.560167403	2164.929392	110	0.6918829430	0.691256530	345.470432
8	0.5479033269	0.547673894	2338.283990	111	0.6900286591	0.689398476	371.212742
15	0.5477257234	0.547496201	2340.749590	113	0.6881059963	0.687471904	397.904328
11	0.5399918592	0.539758412	2448.115849	105	0.6749494192	0.674536709	580.552022
12	0.5366925553	0.536457434	2493.918814	103	0.6686259240	0.667952228	668.338670
24	0.5272240676	0.526984141	2625.366182	109	0.6617170741	0.661287563	764.251572
23	0.5269137394	0.526673655	2629.674349	102	0.6591000839	0.658667250	800.582239
14	0.4742497897	0.473982979	3360.787680	104	0.6588833430	0.658189840	803.591170
22	0.4679160752	0.467240499	3448.716198	115	0.6460410513	0.645321440	981.875759
20	0.4360003713	0.435714150	3891.789624	116	0.6459147055	0.645194837	983.629769
21	0.4181425766	0.417847292	4139.702510	112	0.6428462940	0.642120187	1026.227345
28	0.1891618877	0.188132383	7318.553223	118	0.6259161091	0.625155583	1261.262581
27	0.1750226514	0.173975194	7514.842794	119	0.6179523486	0.617175632	1371.820401
18	0.1740806991	0.173032046	7527.919556	114	0.6113754104	0.610881982	1463.125501
26	0.0136498726	0.013149314	9755.118932	108	0.5811955178	0.580344073	1882.101326
17	0.0135425036	0.013041891	9756.609493	117	0.5231610585	0.522191627	2687.771373
16	0.0035250866	0.003019390	9895.677435	121	0.7166916471	0.716043509	3.060748
40	0.6863446824	0.685866671	418.355972	122	0.7159913448	0.715341605	12.782776
30	0.6730190392	0.672770069	603.350744	124	0.7053864342	0.704712433	160.006666
34	0.6578186873	0.657558143	814.371375	120	0.6982375615	0.697777675	259.251711
39	0.6491776783	0.648910555	934.331175	125	0.6921315380	0.691427213	344.019284
38	0.6485124151	0.648244785	943.566767	123	0.6704315909	0.669677621	645.271293
29	0.6435875679	0.643316188	1011.936524	126	0.6461055464	0.645295925	982.980399
37	0.6243214807	0.623748945	1279.400194	127	0.7166960230	0.715975698	5.000000

Examining Table 4.10, the 127. sub-model is preferred, because R^2_{adj} is closed to full model R^2 and C_p value is less than the number of predictors in the model.

Model127→radon_data_new$actpress+radon_data_new$r.humidity+radon_data_new$temp+radon_data_new$soiltemp+radon_data_new$w.speed+radon_data_new$w.degree+ radon_data_new$w.direction

Analysis was continued with the `ols_step_forward_aic(model)` function for variable selection with forward selection.

Table 4.11 Forward selection method summary table

Selection Summary

Variable	AIC	Sum Sq	RSS	R-Sq	Adj. R-Sq
radon_data_new$actpress	23403.025	100502.226	87062.560	0.53583	0.53571
radon_data_new$soiltemp	22310.882	121594.304	65970.482	0.64828	0.64810
radon_data_new$w.direction	21867.122	128734.094	58830.693	0.68634	0.68587
radon_data_new$r.humidity	21640.758	132043.750	55521.037	0.70399	0.70346
radon_data_new$temp	21479.557	134294.295	53270.491	0.71599	0.71541
radon_data_new$w.speed	21471.785	134426.116	53138.671	0.71669	0.71604

Table 4.11 shows the significant variables that should be included in the model. These variables are actural pressure, soil temperature, wind direction, relative humidity, temperature, and wind speed. In the reverse elimination method, the appropriate model was examined using the `ols_step_backward_aic(model)` function and the model was re-established with the specified variables.

Table 4.12 Backward elimination method summary table

Backward Elimination Summary

Variable	AIC	RSS	Sum Sq	R-Sq	Adj. R-Sq
Full Model	21473.724	53137.850	134426.936	0.71670	0.71598
radon_data_new$w.degree	21471.785	53138.671	134426.116	0.71669	0.71604

In the Table 4.12, it is seen that wind degree to be removed from the model. After variable selection methods, it was observed that the variable that should be removed from the model is the wind degree. As in Figure 4.18, the new model was established with actual pressure, soil temperature, wind direction, relative humidity, temperature, wind speed variables.

```
model_new=lm(radon_data_new$radon~radon_data_new$actpress+
radon_data_new$r.humidity+radon_data_new$temp+radon_data_new$soiltemp50
+radon_data_new$w.speed+radon_data_new$w.direction)
```

Figure 4.18 Creating the final model

When the summary results of the new model established without wind degree were examined in Table 4.13, all the coefficients are significant and the model was valid.

Table 4.13 Summary results for the final model

```
Call:
lm(formula = radon_data_new$radon ~ radon_data_new$actpress +
    radon_data_new$r.humidity + radon_data_new$temp + radon_data_new$soiltemp +
    radon_data_new$w.speed + radon_data_new$w.direction)

Residuals:
     Min       1Q   Median       3Q      Max
-13.8432  -2.3309   0.0144   2.4815  13.0865

Coefficients:
                            Estimate Std. Error t value Pr(>|t|)
(Intercept)                394.202204  12.760826  30.892  < 2e-16 ***
radon_data_new$actpress     -0.388854   0.012415 -31.322  < 2e-16 ***
radon_data_new$r.humidity    0.063640   0.003429  18.560  < 2e-16 ***
radon_data_new$temp          0.235857   0.017890  13.184  < 2e-16 ***
radon_data_new$soiltemp      0.650799   0.025191  25.835  < 2e-16 ***
radon_data_new$w.speed      -0.186860   0.059816  -3.124   0.0018 **
radon_data_new$w.directionE  5.882754   0.586381  10.032  < 2e-16 ***
radon_data_new$w.directionN  3.973415   0.582145   6.825 1.01e-11 ***
radon_data_new$w.directionS  5.946879   0.595529   9.986  < 2e-16 ***
radon_data_new$w.directionW  4.349869   0.602009   7.226 5.96e-13 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.675 on 3934 degrees of freedom
Multiple R-squared:  0.7167,    Adjusted R-squared:  0.716
F-statistic:  1106 on 9 and 3934 DF,  p-value: < 2.2e-16
```

In order to further improve the model, model alternatives have been tried by adding squared expressions of temperature and soil temperature based on the scatter charts between dependent and independent variables, but no significant contribution has been obtained. The best model obtained as a result is defined below.

$\widehat{\text{Radon}} = \widehat{\beta_0} + \widehat{\beta_1}\text{actpressure} + \widehat{\beta_2}\text{relativehumidity} + \widehat{\beta_3}\text{temperature} + \widehat{\beta_4}\text{soiltemperature} + \widehat{\beta_5}\text{windspeed} + \widehat{\beta_6}\text{winddirection}$

One of the goals of this study is to compare the model performance of machine learning algorithms used in practice. For this purpose, the model performance criteria for the model established using multiple linear regression were obtained. While obtaining performance criteria *K*=5-fold cross-validation tests were applied to the algorithms. `Library(MLmetrics)` package in R was used for Model Performance Criteria. The process steps are shown in Figure 4.19 below.

```
library(MLmetrics)
model_multiple <- train(radon~actpress+r.humidity+temp+soiltemp+w.speed+w.direction,
                        data = radon_data_new,
                        method = "lm",
                        trControl = trainControl("cv", number = 5))
predicted. multiple <- predict(model_ multiple, radon_data_new)
R2(predicted. multiple, radon_data_new$radon)
MSE(predicted. multiple, radon_data_new$radon)
RMSE(predicted. multiple, radon_data_new$radon)
MAE(predicted. multiple, radon_data_new$radon)
```

Figure 4.19 Analysis of model performance criteria with multiple linear regression algorithm

Table 4.14 Model performance criteria for multiple linear regression algorithm

R^2	MSE	RMSE	MAE
0.72	13.47	3.67	2.90

It is seen in Table 4.14 that Mean Absolute Error and Root Mean Squared Error values are close to zero in the results obtained. This means that the model error is low. The reason for the average value Mean Squared Error slightly higher is the square expression. For the coefficient of determination, it can be argued that it is not a failure for the model in terms of explainability. Although these values alone do not make sense, meaningful results will be achieved when that are compared to criteria obtained from other algorithms.

4.3 Support Vector Regression Analysis

In this section, Support Vector Regression was applied with kernel functions. The most commonly used kernel functions are linear, polynomial and radial basis functions. Model performance criteria were obtained by applying a *K*=5-fold cross-validation test to these functions.

First, analysis was performed using the Linear Kernel Function in Figure 4.20. The `e1071` package was used for Support Vector Regression in R. With this function, parameters related to the model are also automatically selected to determine the best model.

```
library(e1071)
svm_model <- train( radon~actpress+r.humidity+temp+soiltemp+w.speed+w.direction,
    data = radon_data_new,
    method = "svmLinear",
    trControl = trainControl("cv", number = 5))
svm_model$bestTune
predicted.svm <- predict(svm_model, radon_data_new)
R2(predicted.svm, radon_data_new$radon)
MSE(predicted.svm, radon_data_new$radon)
RMSE(predicted.svm, radon_data_new$radon)
MAE(predicted.svm, radon_data_new$radon)
```

Figure 4.20 Analysis of model performance criteria of support vector regression algorithm for linear kernel function

The regulating parameter for the Support Vector Regression parameter is determined $C = 1$. Performance criteria are found in Table 4.15.

Table 4.15 Support vector regression algorithm model performance criteria for linear kernel function

R^2	MSE	RMSE	MAE
0.72	13.52	3.68	2.9

When the results are evaluated, it seems that they are almost close to the results of Multiple Linear Regression Analysis. Here again it means that the model error is low because the Mean Absolute Error and Root Mean Squared Error values are close to zero. It can be argued that the model is explainable at a rate of 0.72 with the coefficient of determination. For more meaningful expression of values, the results from other algorithms should be examined.

Analysis was continued with the Polynomial Kernel Function as in Figure 4.21.

```
svm_model <- train( radon~actpress+r.humidity+temp+soiltemp+w.speed+w.direction,
    data = radon_data_new,
    method = "svmPoly",
    trControl = trainControl("cv", number = 5))
svm_model$bestTune
predicted.svm <- predict(svm_model, radon_data_new)
R2(predicted.svm, radon_data_new$radon)
MSE(predicted.svm, radon_data_new$radon)
RMSE(predicted.svm, radon_data_new$radon)
MAE(predicted.svm, radon_data_new$radon)
```

Figure 4.21 Analysis of model performance criteria of support vector regression algorithm for polynomial kernel function

In addition to the regulating parameter, degree and scale parameters are used in the Polynomial Kernel Function. C = 0.25, *degree* = 3 and *scale* = 0.1 are determined automatically with the `train()` function.

Table 4.16 Support vector regression algorithm model performance criteria for polynomial kernel function

R^2	MSE	RMSE	MAE
0.77	10.79	3.28	2.49

When the results in Table 4.16 are examined Mean Absolute Error and Root Mean Squared Error are closer to zero than Lineer Kernel Function. In addition, it can be asserted that the explainability of the radon dependent variable increases by the increase of the coefficient of determination to 0.77.

The another kernel function analysis was performed using the Radial Basis Kernel Function as shown in Figure 4.22. The algorithm parameters were determined as C = 1 and *sigma* = 0.1240049, the model performance criteria were obtained as in Table 4.17.

```
svm_model <- train(  radon~actpress+r.humidity+temp+soiltemp+w.speed+w.direction,
    data = radon_data_new,
    method = "svmRadial",
    trControl = trainControl("cv", number = 5))
svm_model$bestTune
predicted.svm <- predict(svm_model, radon_data_new)
R2(predicted.svm, radon_data_new$radon)
MSE(predicted.svm, radon_data_new$radon)
RMSE(predicted.svm, radon_data_new$radon)
MAE(predicted.svm, radon_data_new$radon)
```

Figure 4.22 Analysis of model performance criteria of support vector regression algorithm for radial basis kernel function

Table 4.17 Support vector regression algorithm model performance criteria for radial basis kernel function

R^2	MSE	RMSE	MAE
0.8	9.35	3.06	2.27

It has been observed that the values obtained here are better than the other kernel models. The explainability of the dependent variable radon by independent variables increased and the mean errors decreased.

4.4 Regression Trees Analysis

For regression trees, the `rpart` package in R was used. Model performance criteria were obtained by applying a *K*=5-fold cross-validation test to these functions. These steps are given in Figure 4.23.

```
library(rpart)
rtree_model <- train(radon~actpress+r.humidity+temp+soiltemp+w.speed+w.direction,
    data = radon_data_new,
    method = "rpart",
    trControl = trainControl("cv", number = 5))
rtree_model$bestTune
predicted.rt <- predict(rtree_model, radon_data_new)
R2(predicted.rt, radon_data_new$radon)
MSE(predicted.rt, radon_data_new$radon)
RMSE(predicted.rt, radon_data_new$radon)
MAE(predicted.rt, radon_data_new$radon)
```

Figure 4.23 Analysis of model performance criteria for regression trees algorithm

The parameter used in the Regression Trees algorithm with the corresponding function had the value *cp* = 0.0343546. Table 4.18 is shown model performance criteria.

Table 4.17 Model performance criteria for the regression trees algorithm

R^2	MSE	RMSE	MAE
0.61	18.47	4.3	3.39

It was observed that the values of model performance criteria for Regression Trees were worse than other algorithms. The coefficient of determination had the smallest value compared to the values in other algorithms, and the explainability has decreased. Mean Errors have increased further.

4.5 Bagging Analysis

The analysis continued with one of the Ensemble Learning Algorithm Techniques. The `ipred` package is used Bagging method in Figure 4.24 and model performance criteria were obtained in Table 4.19.

```
library(ipred)
bag_model <- train(radon~actpress+r.humidity+temp+soiltemp+w.speed+w.direction,
    data = radon_data_new,
    method = "treebag",
    trControl = trainControl("cv", number = 5))
predicted.bag <- predict(bag_model, radon_data_new)
R2(predicted.bag, radon_data_new$radon)
MSE(predicted.bag, radon_data_new$radon)
RMSE(predicted.bag, radon_data_new$radon)
MAE(predicted.bag, radon_data_new$radon)
```

Figure 4.24 Analysis of model performance criteria for bagging algorithm

Table 4.18 Model performance criteria for the bagging algorithm

R^2	MSE	RMSE	MAE
0.75	12.02	3.47	2.7

According to the results obtained in Regression Tree Analysis, the results obtained by Bagging method are quite acceptable. This is possible because it is expected that the results obtained from Regression Trees will improve by using the Bagging method.

4.6 XGBoost (Extreme Gradıent Boostıng) Analysis

There are quite a lot of parameters related to this algorithm. XGBoost library used and again the `train()` function automatically determined the optimum parameters for the best model, respectively, are *nrounds* = 150, *max_depth* = 3, *eta* = 0.4, $\gamma = 0$, *colsample_bytree* = 0.8, *min_child_weight* = 1, *subsample* = 1. The process steps are shown in Figure 4.25.

```
library(xgboost)
xgb_model <- train(radon~actpress+r.humidity+temp+soiltemp+w.speed+w.direction,
    data = radon_data_new,
    method = "xgbTree",
    trControl = trainControl("cv", number = 5))
xgb_model$bestTune
predicted.xgb <- predict(xgb_model, radon_data_new)
R2(predicted.xgb, radon_data_new$radon)
MSE(predicted.xgb, radon_data_new$radon)
RMSE(predicted.xgb, radon_data_new$radon)
MAE(predicted.xgb, radon_data_new$radon)
```

Figure 4.25 Analysis of model performance criteria for the xgboost algorithm

Table 4.19 Model performance criteria for xgboost algorithm

R^2	MSE	RMSE	MAE
0.93	3.29	1.81	1.36

It was mentioned in Section 2.5 that the analysis results obtained by the XGBoost method, which is an upgrade algorithm, are better than other methods. As shown in the Table 4.20, the best results have been achieved with XGBoost. The coefficient of determination was found as 0.93, meaning that the explainability is quite high. The Mean Errors are quite close to zero.

4.7 Random Forests Analysis

Analysis was continued with the Random Forests Algorithm in Figure 4.26, and `train()` function automatically determined the optimum *mtry*=9 using the packages of `randomForest` and `caret`.

```
library(randomForest)
rf_model <- train(radon~actpress+r.humidity+temp+soiltemp+w.speed+w.direction,
    data = radon_data_new,
    method = "rf",
    trControl = trainControl("cv", number = 5))
rf_model$bestTune
predicted.rf <- predict(rf_model, radon_data_new)
R2(predicted.rf, radon_data_new$radon)
MSE(predicted.rf, radon_data_new$radon)
RMSE(predicted.rf, radon_data_new$radon)
MAE(predicted.rf, radon_data_new$radon)
```

Figure 4.26 Analysis of model performance criteria for random forests algorithm

Table 4.20 Model performance criteria for random forests algorithm

R^2	MSE	RMSE	MAE
0.99	0.71	0.84	0.61

The resulting performance criteria have the best performance criteria among algorithms in Table 4.21. With the coefficient of determination (0.99), almost the entire dependent variable is explained by independent variables. Mean Absolute Error and and Root Mean Square Error values are almost zero, and it can be argued that the model error is the smallest one.

4.8 k-Nearest Neighbor (kNN) Analysis

There are quite a lot of parameters related to this algorithm. XGBoost library used and again the `train()` function automatically determined the optimum parameters for the best model, respectively, are *nrounds* = 150, *max_depth* = 3, *eta* = 0.4, $\gamma = 0$, *colsample_bytree* = 0.8, *min_child_weight* = 1, *subsample* = 1. The process steps are shown in Figure 4.25.

The `train()` function was used k-Nearest Neighbors Algorithm in R. The package automatically determined number of neighbours as k = 5. The analysis steps are given in Figure 4.27.

```
library(caret)
knn_model <- train(radon~actpress+r.humidity+temp+soiltemp+w.speed+w.direction,
    data = radon_data_new,
    method = "knn",
    trControl = trainControl("cv", number = 5))
knn_model$bestTune
predicted.knn <- predict(knn_model, radon_data_new)
R2(predicted.knn, radon_data_new$radon)
MSE(predicted.knn, radon_data_new$radon)
RMSE(predicted.knn, radon_data_new$radon)
MAE(predicted.knn, radon_data_new$radon)
```

Figure 4.27 Analysis of model performance criteria of k-nearest neighbor algorithm

Table 4.21 Model performance criteria for the k-nearest neighbor algorithm

R^2	MSE	RMSE	MAE
0.92	3.57	1.89	1.32

It can be claimed that the results obtained from Table 4.22 and the results obtained from the XGBoost algorithm give almost close results. Model error is quite small and model explainability is high. It is one of the best when compared to other algorithms.

CHAPTER 5
CONCLUSION

The aim of this study is to evaluate the effects of Hourly Actual Pressure (hPa), Hourly 50 cm Soil Temperature (°C), Hourly Relative Humidity (%), Hourly Temperature (°C), Hourly Wind Degree (°), Hourly Wind Speed (m/sec) and Wind Direction factors on Radon gas using Supervised Learning Algorithms and to estimate the Radon gas values. Multiple Linear Regression, Support Vector Regression, Regression Trees, Bagging, Random Forests, XGBoost, and k-Nearest Neighbor were used from among Supervised Learning Algorithms. To test the success of algorithms Coefficient of Determination (R^2), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) values with performing *K*= 5 Fold Cross-Validation.

All analysis results obtained in the previous section are presented by combining them in Table 5.1. According to the results, the Random Forests Algorithm for Radon estimation decisively showed the highest performance among Supervised Statistical Learning Methods. This method was followed by the XGBoost and k-Nearest Neighbors Algorithms, which gave very close results. The worst performance was demonstrated by the Regression Trees Algorithm.

Table 5. 1 Model performance criteria obtained for all algorithms

MACHINE LEARNING ALGORITHM	**R^2**	**MSE**	**RMSE**	**MAE**
MULTIPLE LINEAR REGRESSION	0.72	13.47	3.67	2.90
SUPPORT VECTOR MACHINE **"SVMLINEAR"**	0.72	13.52	3.68	2.90
SUPPORT VECTOR MACHINE **"SVMPOLY "**	0.77	10.79	3.28	2.49
SUPPORT VECTOR MACHINE **"SVMRADIAL "**	0.80	9.35	3.06	2.27
REGRESSION TREES	0.61	18.47	4.30	3.39
BAGGING	0.75	12.02	3.47	2.70
XGBOOST	0.93	3.29	1.81	1.36
RANDOM FORESTS	0.99	0.71	0.84	0.61
k-NEAREST NEIGHBORS	0.92	3.57	1.89	1.32

In addition, Radon gas is a radioactive substance that we are exposed substantially in our daily lives, and it has a close relationship with earthquakes, as well as meteorological parameters. In many studies in the literature, the relationship between radon gas and earthquake has been investigated. As a further research, it can be suggested to investigate the effect of radon gas on earthquakes through meteorological parameters affecting Radon gas.

REFERENCES

Abar, H. (2020). XGBOOST ve MARS yöntemleriyle altın fiyatlarının kestirimi. *Ekev Akademi Dergisi, 83,* 427-446.

Açıkkar, M., & Sivrikaya, O. (2020). Yıkanmış Türk linyit kömürlerinin üst ısıl değerinin destek vektör regresyonu ile tahmini. *Avrupa Bilim ve Teknoloji Dergisi,* (18), 16-24.

Albalate, A., & Minker, W. (2013). *Semi-supervised and unsupervised machine learning: novel strategies.* New Jersey: John Wiley & Sons.

Alkış, B. N.(2017). *Çoklu lojistik regresyon ve k-en yakın komşu yöntemleri ile BİST 100 endeks getiri yönünün tahmini.* Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi, İstanbul.

Amit, Y., & Geman, D. (1997). Shape quantization and recognition with randomized trees. *Neural computation, 9*(7), 1545-1588.

Anava, O., & Levy, K. (2016). k*-nearest neighbors: From global to local. In *Advances in neural information processing systems*, 4916-4924.

Arıkpınar, A.(2010). *İzmir Seferihisar ve Balçova jeotermal bölgelerdeki binalarda radon düzeylerinin belirlenmesi.* Yüksek Lisans Tezi, Ege Üniversitesi, İzmir.

Aşık, E. (2019). *Trabzon ilindeki akciğer kanserleri ile radon gazı arasındaki ilişkinin değerlendirilmesi.* Uzmanlık Tezi, Karadeniz Teknik Üniversitesi, Trabzon.

Breiman, L. (1994). *Bootstrap aggregating* (No. 421). Technical Report.

Breiman, L. (1996a). Bagging predictors. *Machine learning*, *24*(2), 123-140.

Breiman, L. (1996b). Out-of-bag estimation.ftp.stat.berkeley.edu/pub/users/ breiman. *OOBestimation. ps*, *199*(6).

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

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (2017). *Classification and regression trees*. Florida: Routledge.

Burges, C. J. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, *2*(2), 121-167.

Cebeci, Z. (2020). *Veri biliminde R ile veri önişleme* (1st ed.). Ankara: Nobel Akademik Yayıncılık.

Chatterjee, S., & Price, B. (1997). *Regression Analysis by Examples* (2nd ed.). New York: John Wiley & Sons.

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 785-794.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273-297.

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, *13*(1), 21-27.

Değerli, F. C. & Umaroğulları, F. (2017). Binalarda Radon ve sağlık üzerine etkileri. *Yeşil Bina Sürdürülebilir Yapı Teknolojileri Dergisi,45,* 38-45.

Ekelik, H., & Altaş, D. (2019). Dijital reklam verilerinden yararlanarak potansiyel konut alıcılarının rastgele orman yöntemiyle sınıflandırılması. *Journal of Research in Economics*, *3*(1), 28-45.

Elasan, S. (2019). *Veri Madenciliğinde farklı Karar Ağaçları ve K-en Yakın Komşuluk yöntemlerinin incelenmesi: kadın hastalıkları ve doğum verisinde bir uygulama*. Doktora Tezi, Van Yüzüncü Yıl Üniversitesi, Van.

Ercire, M. (2019). *Kısa süreli güç kalitesi bozulmalarının dalgacık analizi ve rastgele orman yöntemi ile sınıflandırılması*. Yüksek Lisans Tezi, Dumlupınar Üniversitesi, Kütahya.

Fix, E., & Hodges, J. L. (1951). *Discriminatory Analysis, Nonparametric Discrimination: Consistency Properties USAF School of Aviation Medicine, Randolph Field* (pp. 1-21). Texas, Tech. Report 4.

Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The elements of statistical learning* (2nd ed.). New York: Springer series in statistics.

Goyal, R., Chandra, P., & Singh, Y. (2014). Suitability of KNN regression in the development of interaction based software fault prediction models. *Ieri Procedia*, *6*, 15-21.

Göker, D. (2010). *Sürekli Radon Gazı Ölçümlerinin Deprem Tahmin Parametresi Olarak Kullanılması, İzmir Seferihisar Doğanbey Fay Hattı Örneği*. Yüksek Lisans Tezi, Ege Üniversitesi, İzmir.

Günay, O. (2016). *Batı Marmara bölgesinde toprak gazı radon konsantrasyon değişimleri ile yer kabuğu hareketleri arasındaki ilişkilerin incelenmesi.* Doktora Tezi, Ege Üniversitesi, İzmir.

Gürleyen, S. (2017). *Farklı metamorfik birimlerde aktif karbonla toprak gazı radon ölçümü ve radon potansiyelini etkileyen parametrelerin incelenmesi.* Yüksek Lisans Tezi, Ege Üniversitesi,İzmir.

Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine İntelligence, 20*(8), 832-844.

İlhan, M. Z. (2015). *Afyonkarahisar şehir merkezinde toprakta radon aktivite konsantrasyonunun belirlenmesi.*Yüksek Lisans Tezi, Afyon Kocatepe Üniversitesi, Afyonkarahisar.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning* (8th ed.). New York: Springer.

Kamışlıoğlu, M. (2016). *Kaotik zaman serileri analizi ile radon gazı (222Rn), deprem ve meteorolojik parametreler arasındaki ilişkinin modellenmesi.* Doktora Tezi, Fırat Üniversitesi, Elazığ.

Keskin, M. V. (2019). *Ağaca dayalı yöntemlerde bagging ve boosting arasında ne fark var?*. Retrieved July 15, 2021, from https://www.veribilimiokulu.com/agaca-dayali-yontemlerde-bagging-ve-boosting-arasinda-ne-fark-var/.

Korkem, E. (2013). *Mikroarray Gen Ekspresyon Veri Setlerinde Random Forest ve Naıve Bayes Sınıflama Yöntemleri Yaklaşımı.* Yüksek Lisans Tezi, Hacettepe Üniversitesi, Ankara.

Montgomery, D. C., Peck, E. A., & Vining, G. G. (2013). *Introduction to linear regression analysis* (5th ed.). New Jersey: John Wiley & Sons.

Morgan, J. N., & Sonquist, J. A. (1963). Problems in the analysis of survey data, and a proposal. *Journal of the American Statistical Association, 58*(302), 415-434.

Muratlar, E. R. (2020). *Gradient Boosted Regresyon Ağaçları.* Retrieved July 15, 2021, from https://www.veribilimiokulu.com/gradient-boosted-regresyon-agaclari/

Muratlar, E. R. (2020). *XGBoost nasıl çalışır? Neden iyi performans gösterir?*. Retrieved July 15, 2021, from https://www.veribilimiokulu.com/XGBoost-nasil-calisir/.

Neter, J., Kutner, M. H., Nachtsheim, C. J., & Wasserman, W. (1996). *Applied linear statistical models* (4th ed.). Chicago: Irwin.

Noyan, M. (2020). *XGBoost dosyaları.* Retrieved July 15, 2021, from https://medium.com/deep-learning-turkiye/XGBoost-dosyalar%C4%B1-25d31fda6900.

Özkan, K. (2012). Sınıflandırma ve regresyon ağacı tekniği (SRAT) ile ekolojik verinin modellenmesi. *Süleyman Demirel Üniversitesi Orman Fakültesi Dergisi, 13*(1), 1-4.

Pişkin, A. (2017). *Aydın ili Koçarlı bölgesi ve çevresinde radon konsantrasyonlarının toprak yapısı ve çevresel parametreler ile olan ilişkilerinin belirlenmesi.* Doktora Tezi, Ege Üniversitesi,İzmir.

Smola, A. J., & Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing, 14*(3), 199-222.

Smola, A., & Vishwanathan, S. V. N. (2008). Introduction to machine learning. *Cambridge University, UK, 32*(34), 2008.

Sullivan, W. (2017). *Machine learning for beginners guide algorithms: supervised & unsupervsied learning, decision tree & random forest ıntroduction*. Coimbatore: Healthy Pragmatic Solutions.

Sun, S., & Huang, R. (2010). An adaptive k-nearest neighbor algorithm. In *2010 Seventh İnternational Conference on Fuzzy Systems and Knowledge Discovery*, IEEE, *1*, 91-94.

Tabar, E. (2010). *Dikili ve Bergama Bölgelerindeki Tektonik Aktivitenin Jeotermal Sularda Radon Ölçümleri ile Değerlendirilmesi.* Yüksek Lisans Tezi, Ege Üniversitesi, İzmir.

Wettschereck, D., & Dietterich, T. G. (1994). Locally adaptive nearest neighbor algorithms. *Advances in Neural Information Processing Systems*, 184-184.

Yücel, B., Özdemir, T., Gökhan, K., Ataksor, B., & Albayrak, N. (2012). *Kapalı ortamlarda radon gazı*. Türkiye Enerji, Nükleer ve Maden Araştırma Kurumu Kurumsal Araştırma Arşivi.