DOKUZ EYLÜL UNIVERSITY
ENGINEERING FACULTY
DEPARTMENT OF COMPUTER ENGINEERING

CME 4416 INTRODUCTION TO DATA MINING

NAIVE BAYES, K-NEAREST NEIGHBOR, K-MEANS, LINEAR REGRESSION, MULTILAYER PERCEPTRON, NON LINEAR REGRESSION (M5P) ALGORITHM WITH REAL ESTATE

by
2017510067 - Şefika Özlem PUL
2017510049 - Ece KAZAN

Advisor
Assoc. Prof. Dr. Feriştah DALKILIÇ

December 2020
İZMİR

CONTENTS

1. Information About the Project

The project to be done is to develop a mobile application in the field of real estate using augmented reality. Augmented Reality also allows the physical environment of the real world to be combined with the virtual environment via the phone. Therefore, the project will be developed by combining augmented reality and mobile application.

By using the mobile application, location information and phone camera to be developed, the information of the places for sale or rental will be displayed to the users. At the same time, advertisements similar to the advertisement watched by the user and advertisements viewed by other users will be shown to the user as suggestions. In addition, users will be able to make their ads visible to other users with the ad sharing feature. When users want to add ads, the existing ads are analyzed according to the selected features and a price suggestion is made to the user.

2. Explanation About Our Algorithm

2.1 Naive Bayes

It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. The Naive Bayes model is a conditional probability classification with Bayes Theorem applied.

Conditional probability defines the probability of an event occurring given the occurrence of another event. Another interesting thing we could do with the conditional probability is use it to calculate the joint probability – which is the probability of two or more simultaneous events. On the flip side, the joint probability could also be used to calculate the conditional probability.

Naive Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions. It is a probabilistic classifier, which means it predicts on the basis of the probability of an object.

Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

The purpose of using this algorithm in the project is to provide a price suggestion to the user when the user wants to add a new ad, when the features of this ad and the features of the ads in the dataset are highly matched.

2.2 K-Nearest Neighbor

K-Nearest Neighbor is one of the simplest Machine Learning algorithms based on the Supervised Learning technique.

This algorithm assumes the similarity between the new state/data and the existing states and places the new state in the category most similar to the existing categories and classifies a new data point according to the similarity, keeping all existing data. This means that when new data appears, it can easily be classified into a well pack category using the K-NN algorithm.

It is also called a lazy learner algorithm because it does not learn from the training set immediately, but instead stores the dataset and performs an operation on the dataset at the time of classification. The K-NN working can be explained on the basis of the below algorithm:

Step-1: Select the number K of the neighbors

Step-2: Calculate the Euclidean distance of K number of neighbors

Step-3: Take the K nearest neighbors as per the calculated Euclidean distance.

Step-4: Among these k neighbors, count the number of the data points in each category.

Step-5: Assign the new data points to that category for which the number of the neighbor is maximum.

Step-6: Our model is ready.

The purpose of using this algorithm in the project is to list the k advertisement in the dataset as closely as possible to the features the user wants. In this way, the user can easily access the ads with the desired features instead of examining all the advertisement.

2.3 K-Means

K-Means Clustering is an unsupervised learning algorithm that is used to solve the clustering problems in machine learning or data science. K-Means Clustering is an Unsupervised Learning algorithm, which groups the unlabeled dataset into different clusters. It is an iterative algorithm that divides the unlabeled dataset into k different clusters in such a way that each dataset belongs only one group that has similar properties. It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabeled dataset on its own without the need for any training.

It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters.

The algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm.

The k-means clustering algorithm mainly performs two tasks: Determines the best value for K center points or centroids by an iterative process. Assigns each data point to its closest k-center. Those data points which are near to the particular k-center, create a cluster. Hence each cluster has datapoints with some commonalities, and it is away from other clusters. The working of the K-Means algorithm is explained in the below steps:

Step-1: Select the number K to decide the number of clusters.

Step-2: Select random K points or centroids. (It can be other from the input dataset).

Step-3: Assign each data point to their closest centroid, which will form the predefined K clusters.

Step-4: Calculate the variance and place a new centroid of each cluster.

Step-5: Repeat the third steps, which means reassign each datapoint to the new closest centroid of each cluster.

Step-6: If any reassignment occurs, then go to step-4 else go to FINISH.

Step-7: The model is ready.

The purpose of using this algorithm is to divide the data in the dataset into k clusters. In this way, the advertisements in the data set are clustered and the advertisements with similar characteristics are placed in the same cluster.

2.4 Linear Regression

Linear regression may be defined as the statistical model that analyzes the linear relationship between a dependent variable with given set of independent variables. Linear relationship between variables means that when the value of one or more independent variables will change (increase or decrease), the value of dependent variable will also change accordingly (increase or decrease).

Linear regression attempts to model the relationship between two variables by fitting a linear equation to observed data. One variable is considered to be an explanatory variable, and the other is considered to be a dependent variable.

The purpose of using this algorithm in the project is to offer a price suggestion to the user by analyzing the features of this ad and the features of the ads in the dataset when the user wants to add a new ad.

2.5 Multilayer Perceptron

A multilayer perceptron (MLP) is a feedforward artificial neural network that generates a set of outputs from a set of inputs. An MLP is characterized by several layers of input nodes connected as a directed graph between the input and output layers. MLP uses backpropagation for training the network. MLP is a deep learning method.

MLPs form the basis for all neural networks and have greatly improved the power of computers when applied to classification and regression problems. Computers are no longer limited by XOR cases and can learn rich and complex models thanks to the multilayer perceptron.

The purpose of using this algorithm in the project is to offer a price suggestion to the user by analyzing the features of this ad and the features of the ads in the dataset when the user wants to add a new ad.

2.6 Non-Linear Regression (M5P)

M5 model tree is a decision tree learner for regression task that is used to predict values of numerical response variable Y, which is a binary decision tree having linear regression functions at the terminal (leaf) nodes.

The splitting criterion of the M5 model tree algorithm is based on treating the standard deviation of the class values that reach a node as an error measure at that node and calculating the expected reduction in this error as a result of testing each attribute at that node.

3. Tools Used in the Project

3.1 Weka

WEKA - an open source software provides tools for data preprocessing, implementation of several Machine Learning algorithms, and visualization tools so that you can develop machine learning techniques and apply them to real-world data mining problems.

You would save the preprocessed data in your local storage for applying ML algorithms. Depending on the kind of ML model that you are trying to develop you would select one of the options such as Classify, Cluster, or Associate. The Attributes Selection allows the automatic selection of features to create a reduced dataset.

WEKA would give you the statistical output of the model processing. It provides you a visualization tool to inspect the data. The various models can be applied on the same dataset. You can then compare the outputs of different models and select the best that meets your purpose.

Analyzes were made by choosing Naive Bayes, K-NN Algorithm and K-Means algorithms in the data set Weka. Naive Bayes and K-NN classification algorithms were preferred for price suggestion. K-Means clustering algorithm was preferred to classify similar advertisements.

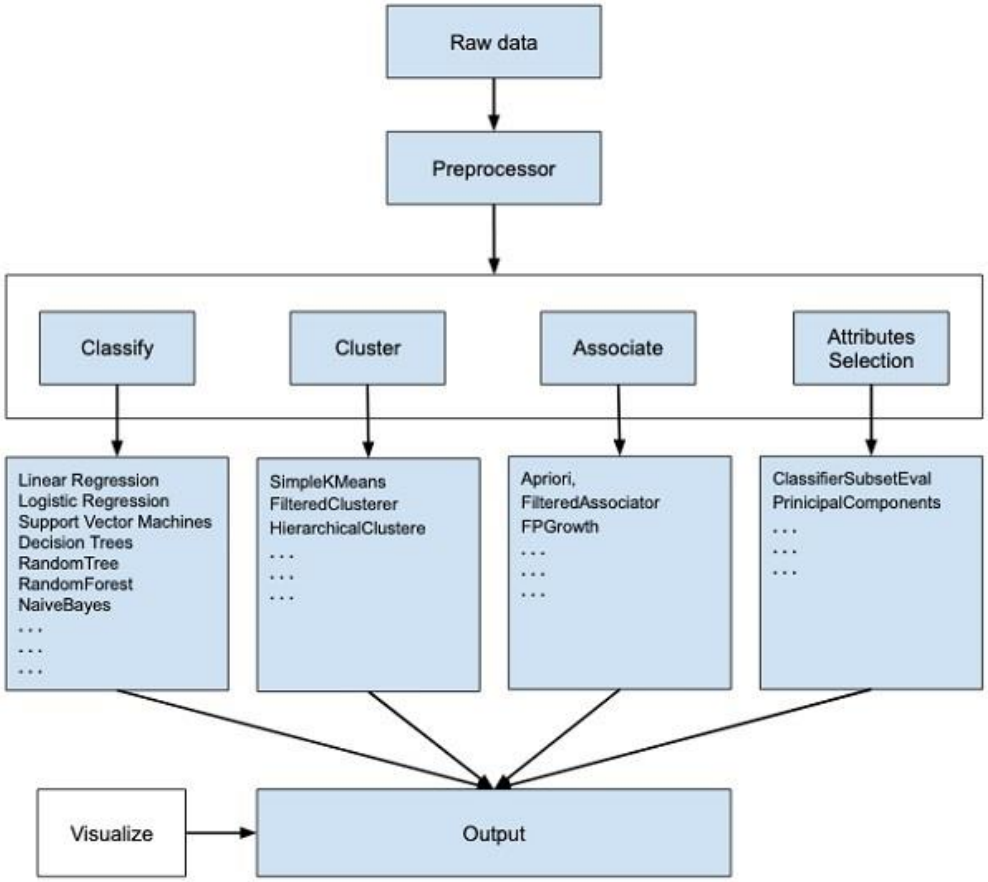

Figure 3.1 Working Diagram of Weka

3.2 Using Weka in C#

Classification and clustering algorithms made in Weka were imported into Weka libraries and Windows Form application was made in C# programming language. The user enters the arff file name of the data set as input. One of the Naive Bayes, K-NN and K-Means algorithms to be used is selected. If the K-NN or K-Means algorithm is selected, the k value is also requested from the user. As a result of the selections made, the model of the file is created. If the user chooses the Naive Bayes or K-NN algorithm, the user is asked to make input values and choices according to the features in the data set. As a result, if the selected algorithm is the classification algorithm, the class of the feature is determined, and if it is the clustering algorithm, it is determined which cluster the data in the data set belongs to.

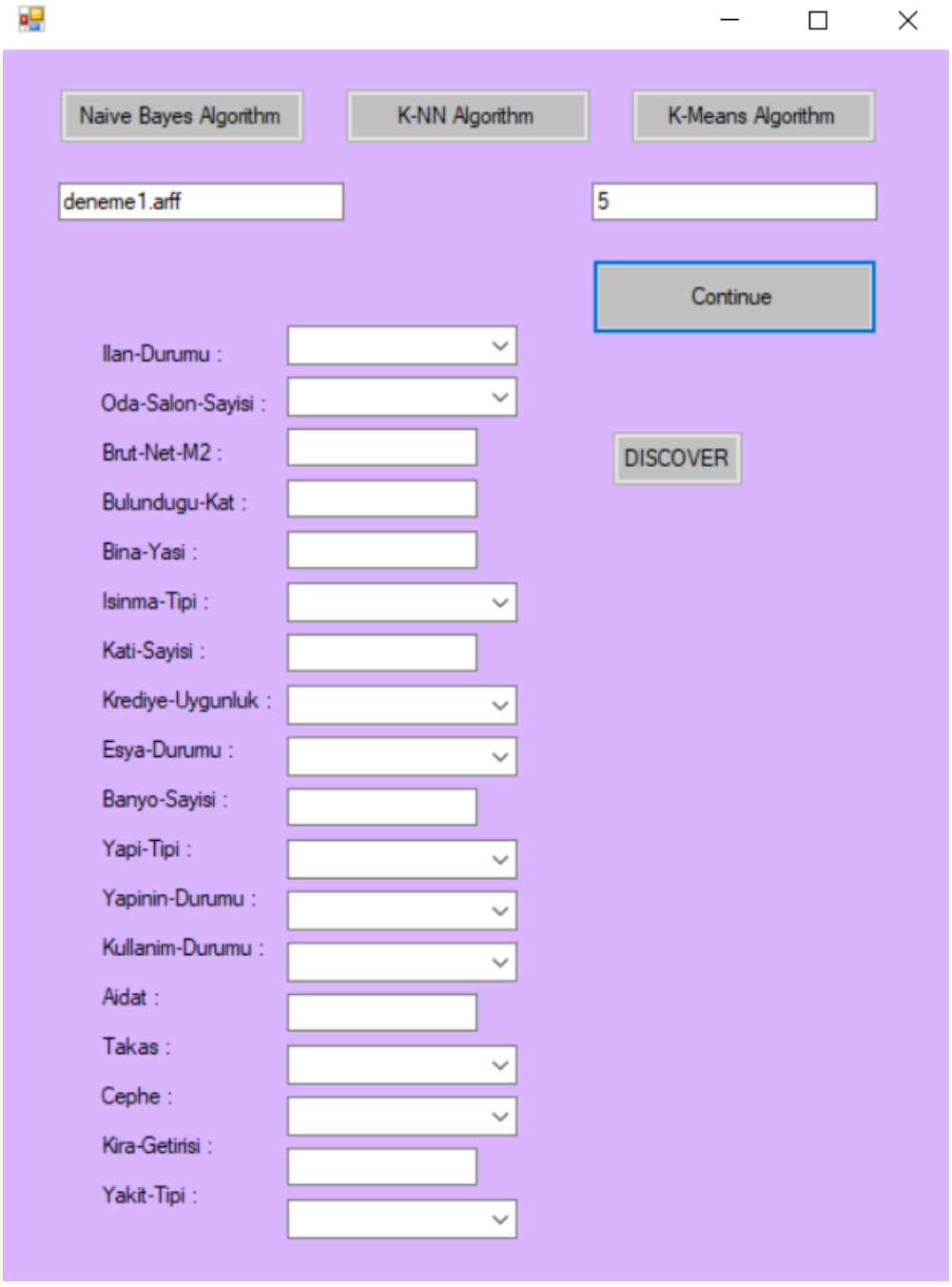

Figure 3.2 KNN Algorithm Selected

4. Project Results Screenshots

4.1 Definition of Some Terms

The TPR, or “Sensitivity”, is a measure of the proportion of positive cases in the data that are correctly identified as such. It is defined in eq. 1 as the total number of correctly identified positive cases divided by the total number of positive cases.

False positive rate (FPR) is a measure of accuracy for a test: be it a medical diagnostic test, a machine learning model, or something else. In technical terms, the false positive rate is defined as the probability of falsely rejecting the null hypothesis.

Precision (also called positive predictive value) is the fraction of relevant instances among the retrieved instances, while recall (also known as sensitivity) is the fraction of relevant instances that were retrieved.

The ability of a model to find all the relevant cases within a data set. Mathematically, we define recall as the number of true positives divided by the number of true positives plus the number of false negatives.

The F-score or F-measure is a measure of a test's accuracy. It is calculated from the precision and recall of the test, where the precision is the number of true positive results divided by the number of all positive results, including those not identified correctly, and the recall is the number of true positive results divided by the number of all samples that should have been identified as positive.

The Matthews correlation coefficient (MCC), instead, is a more reliable statistical rate which produces a high score only if the prediction obtained good results in all of the four confusion matrix categories (true positives, false negatives, true negatives, and false positives), proportionally both to the size of positive elements and the size of negative elements in the dataset.

An ROC curve (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters: True Positive Rate, False Positive Rate

"Precision-Recall Curve classification tree" or simply the "PRC classification tree" modifies two crucial stages in tree building. The first stage is to maximize the area under the precision-recall curve in node variable selection.

4.2 Naïve Bayes

4.2.1 Use Training Set

By choosing the Naive Bayes algorithm, 59.0227% of the 2067 data, 1220 data, were classified correctly.

```
=== Evaluation on training set ===

Time taken to test model on training data: 0.17 seconds

=== Summary ===

Correctly Classified Instances        1220               59.0227 %
Kappa statistic                          0.5683
Mean absolute error                      0.0177
Root mean squared error                  0.0955
Relative absolute error                 52.894  %
Root relative squared error             73.8009 %
Total Number of Instances             2067

=== Detailed Accuracy By Class ===
```

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0,767	0,066	0,345	0,767	0,476	0,483	0,947	0,425	250000-300000
0,179	0,014	0,194	0,179	0,187	0,172	0,934	0,206	300000-350000
0,220	0,017	0,239	0,220	0,229	0,211	0,899	0,268	350000-400000
0,347	0,042	0,231	0,347	0,278	0,252	0,888	0,241	400000-450000
0,299	0,017	0,370	0,299	0,331	0,313	0,900	0,270	450000-500000
0,173	0,012	0,273	0,173	0,212	0,201	0,886	0,164	500000-550000
0,379	0,046	0,212	0,379	0,272	0,252	0,885	0,212	550000-600000
0,098	0,007	0,222	0,098	0,136	0,136	0,886	0,198	600000-650000
0,324	0,039	0,230	0,324	0,269	0,242	0,870	0,193	650000-700000
0,114	0,007	0,250	0,114	0,156	0,157	0,886	0,234	700000-750000
0,140	0,005	0,353	0,140	0,200	0,212	0,910	0,208	750000-800000
0,333	0,009	0,345	0,333	0,339	0,329	0,901	0,259	800000-850000
0,146	0,007	0,333	0,146	0,203	0,209	0,879	0,168	850000-900000
0,000	0,001	0,000	0,000	0,000	-0,003	0,879	0,098	900000-950000
0,125	0,005	0,231	0,125	0,162	0,163	0,945	0,244	950000-1000000
0,000	0,000	?	0,000	?	?	0,957	0,214	1000000-1050000

Figure 4.2.1 Result When “Use Training Set” Option is Selected in Weka (1)

0,059	0,001	0,250	0,059	0,095	0,118	0,945	0,166	1050000-1100000
0,333	0,000	1,000	0,333	0,500	0,577	0,962	0,638	1100000-1150000
0,000	0,000	0,000	0,000	0,000	-0,002	0,929	0,204	1150000-1200000
0,000	0,000	0,000	0,000	0,000	-0,002	0,938	0,156	1200000-1250000
0,280	0,012	0,219	0,280	0,246	0,237	0,935	0,364	1250000-1300000
0,143	0,000	1,000	0,143	0,250	0,377	0,955	0,332	1300000-1350000
0,000	0,000	?	0,000	?	?	0,903	0,104	1350000-1400000
0,167	0,000	1,000	0,167	0,286	0,408	0,985	0,438	1400000-1450000
0,500	0,012	0,242	0,500	0,327	0,341	0,968	0,409	1450000-1500000
0,083	0,000	1,000	0,083	0,154	0,288	0,939	0,272	1500000-1550000
0,176	0,003	0,300	0,176	0,222	0,225	0,962	0,337	1550000-1600000
0,083	0,001	0,333	0,083	0,133	0,164	0,934	0,124	1600000-1650000
0,125	0,003	0,222	0,125	0,160	0,162	0,951	0,242	1650000-1700000
0,000	0,000	?	0,000	?	?	0,974	0,110	1700000-1750000
0,143	0,001	0,333	0,143	0,200	0,217	0,976	0,260	1750000-1800000
0,000	0,000	?	0,000	?	?	0,998	0,546	1800000-1850000
0,500	0,000	1,000	0,500	0,667	0,707	1,000	1,000	1850000-1900000
0,000	0,000	?	0,000	?	?	0,967	0,282	1900000-1950000
0,000	0,000	?	0,000	?	?	0,997	0,564	1950000-2000000
0,653	0,052	0,436	0,653	0,523	0,499	0,939	0,584	2000000-2050000
0,926	0,001	0,893	0,926	0,909	0,908	0,997	0,945	500-1000
0,979	0,005	0,955	0,979	0,967	0,963	0,995	0,992	1000-2000
0,983	0,001	0,992	0,983	0,987	0,986	0,993	0,990	2000-3000
0,986	0,014	0,843	0,986	0,909	0,905	0,995	0,991	3000-4000
0,963	0,003	0,928	0,963	0,945	0,943	0,985	0,949	4000-5000
0,917	0,004	0,830	0,917	0,871	0,869	0,999	0,970	5000-6000
0,889	0,000	0,970	0,889	0,928	0,927	0,999	0,973	6000-7000
0,840	0,001	0,913	0,840	0,875	0,874	1,000	0,975	7000-8000
0,593	0,002	0,800	0,593	0,681	0,685	0,997	0,719	8000-9000
0,500	0,000	1,000	0,500	0,667	0,706	1,000	0,957	9000-10000
0,375	0,000	0,750	0,375	0,500	0,529	0,999	0,833	10000-11000
0,750	0,000	0,857	0,750	0,800	0,801	0,994	0,809	11000-12000
0,563	0,000	0,900	0,563	0,692	0,710	0,999	0,879	12000-13000

Figure 4.2.1 Result When “Use Training Set” Option is Selected in Weka (2)

	0,500	0,000	1,000	0,500	0,667	0,707	0,999	0,831	13000-14000
	0,333	0,000	0,667	0,333	0,444	0,470	0,997	0,659	14000-15000
	0,700	0,001	0,778	0,700	0,737	0,737	0,991	0,715	15000-16000
	0,000	0,000	?	0,000	?	?	0,999	0,478	16000-17000
	0,400	0,000	0,667	0,400	0,500	0,516	0,997	0,617	17000-18000
	0,250	0,000	1,000	0,250	0,400	0,500	0,999	0,635	18000-19000
	0,500	0,000	1,000	0,500	0,667	0,707	0,999	0,625	19000-20000
	0,954	0,005	0,849	0,954	0,899	0,897	0,999	0,974	20000-21000
Weighted Avg.	0,590	0,015	?	0,590	?	?	0,952	0,602	

Figure 4.2.1 Result When “Use Training Set” Option is Selected in Weka (3)

With the use training set, the actual values in the price class in the data set and the estimated price class values as a result of the Naive Bayes algorithm were compared.

```
=== Predictions on training set ===

    inst#     actual  predicted error prediction
        1 36:2000000-2050000 29:1650000-1700000   +   0.188
        2 19:1150000-1200000 36:2000000-2050000   +   0.243
        3 14:900000-950000 2:300000-350000   +   0.229
        4 14:900000-950000 6:500000-550000   +   0.133
        5 36:2000000-2050000 36:2000000-2050000       0.844
        6 4:400000-450000 1:250000-300000   +   0.315
        7 36:2000000-2050000 36:2000000-2050000       0.757
        8 36:2000000-2050000 36:2000000-2050000       0.598
        9 36:2000000-2050000 36:2000000-2050000       0.823
       10 36:2000000-2050000 4:400000-450000   +   0.293
       11 25:1450000-1500000 25:1450000-1500000       0.762
       12 26:1500000-1550000 10:700000-750000   +   0.131
       13 36:2000000-2050000 36:2000000-2050000       0.359
       14 5:450000-500000 5:450000-500000       0.215
       15 27:1550000-1600000 27:1550000-1600000       0.531
       16 34:1900000-1950000 27:1550000-1600000   +   0.439
       17 4:400000-450000 4:400000-450000       0.358
       18 1:250000-300000 36:2000000-2050000   +   0.278
       19 5:450000-500000 1:250000-300000   +   0.44
       20 23:1350000-1400000 8:600000-650000   +   0.191
       21 21:1250000-1300000 36:2000000-2050000   +   0.377
       22 36:2000000-2050000 36:2000000-2050000       0.556
       23 4:400000-450000 6:500000-550000   +   0.329
       24 1:250000-300000 1:250000-300000       0.309
       25 1:250000-300000 9:650000-700000   +   0.245
       26 27:1550000-1600000 27:1550000-1600000       0.726
       27 1:250000-300000 7:550000-600000   +   0.355
       28 13:850000-900000 4:400000-450000   +   0.21
       29 15:950000-1000000 5:450000-500000   +   0.138
       30 36:2000000-2050000 36:2000000-2050000       0.908
```

Figure 4.2.1 Class Comparison in Dataset

4.2.2 Cross Validation

Cross-validation, a standard evaluation technique, is a systematic way of running repeated percentage splits. Divide a dataset into 10 pieces (“folds”), then hold out each piece in turn for testing and train on the remaining 9 together.

When the n value of the Cross Validation is 10, 47.7504 % of the 2067 data, 987 data were classified correctly.

```
Time taken to build model: 0.02 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         987               47.7504 %
Kappa statistic                          0.4491
Mean absolute error                      0.0203
Root mean squared error                  0.1058
Relative absolute error                 60.6983 %
Root relative squared error             81.7953 %
Total Number of Instances             2067

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0,544     0,079     0,239       0,544   0,332      0,318    0,912     0,269     250000-300000
               0,026     0,011     0,042       0,026   0,032      0,018    0,855     0,083     300000-350000
               0,100     0,020     0,111       0,100   0,105      0,084    0,821     0,110     350000-400000
               0,111     0,056     0,067       0,111   0,083      0,043    0,816     0,094     400000-450000
               0,015     0,026     0,019       0,015   0,017      -0,012   0,820     0,089     450000-500000
               0,038     0,016     0,057       0,038   0,046      0,027    0,777     0,059     500000-550000
               0,227     0,048     0,134       0,227   0,169      0,139    0,830     0,140     550000-600000
               0,000     0,010     0,000       0,000   0,000      -0,014   0,747     0,038     600000-650000
               0,183     0,055     0,107       0,183   0,135      0,099    0,790     0,092     650000-700000
               0,045     0,008     0,105       0,045   0,063      0,056    0,790     0,069     700000-750000
               0,000     0,008     0,000       0,000   0,000      -0,013   0,813     0,068     750000-800000
               0,033     0,012     0,040       0,033   0,036      0,024    0,797     0,071     800000-850000
               0,042     0,009     0,095       0,042   0,058      0,048    0,791     0,075     850000-900000
               0,000     0,001     0,000       0,000   0,000      -0,004   0,700     0,021     900000-950000
               0,083     0,004     0,182       0,083   0,114      0,116    0,830     0,060     950000-1000000
               0,000     0,000     0,000       0,000   0,000      -0,002   0,736     0,017     1000000-1050000
```

Figure 4.2.2 Result When "Cross Validation" is Selected and n=10 in Weka (1)

```
0,000   0,003   0,000   0,000   0,000   -0,005   0,828   0,026   1050000-1100000
0,000   0,000   ?       0,000   ?       ?        0,851   0,040   1100000-1150000
0,000   0,001   0,000   0,000   0,000   -0,003   0,743   0,020   1150000-1200000
0,000   0,000   ?       0,000   ?       ?        0,759   0,014   1200000-1250000
0,200   0,012   0,167   0,200   0,182   0,172    0,855   0,154   1250000-1300000
0,000   0,000   ?       0,000   ?       ?        0,631   0,054   1300000-1350000
0,000   0,000   ?       0,000   ?       ?        0,737   0,016   1350000-1400000
0,000   0,000   0,000   0,000   0,000   -0,001   0,731   0,020   1400000-1450000
0,313   0,012   0,167   0,313   0,217   0,220    0,857   0,241   1450000-1500000
0,000   0,000   ?       0,000   ?       ?        0,684   0,014   1500000-1550000
0,118   0,006   0,133   0,118   0,125   0,118    0,896   0,068   1550000-1600000
0,000   0,000   ?       0,000   ?       ?        0,775   0,019   1600000-1650000
0,000   0,006   0,000   0,000   0,000   -0,007   0,848   0,035   1650000-1700000
0,000   0,000   ?       0,000   ?       ?        0,334   0,002   1700000-1750000
0,000   0,000   ?       0,000   ?       ?        0,761   0,013   1750000-1800000
0,000   0,000   ?       0,000   ?       ?        0,688   0,010   1800000-1850000
0,000   0,000   ?       0,000   ?       ?        0,141   0,001   1850000-1900000
0,000   0,000   0,000   0,000   0,000   -0,001   0,497   0,003   1900000-1950000
0,000   0,000   ?       0,000   ?       ?        0,642   0,007   1950000-2000000
0,512   0,053   0,376   0,512   0,434   0,398    0,919   0,465   2000000-2050000
0,852   0,002   0,852   0,852   0,852   0,850    0,980   0,896   500-1000
0,979   0,009   0,917   0,979   0,947   0,942    0,993   0,988   1000-2000
0,975   0,004   0,971   0,975   0,973   0,970    0,990   0,986   2000-3000
0,980   0,019   0,796   0,980   0,878   0,873    0,990   0,985   3000-4000
0,913   0,005   0,880   0,913   0,896   0,892    0,980   0,925   4000-5000
0,813   0,010   0,650   0,813   0,722   0,720    0,988   0,819   5000-6000
0,667   0,003   0,774   0,667   0,716   0,714    0,976   0,816   6000-7000
0,480   0,002   0,750   0,480   0,585   0,596    0,916   0,735   7000-8000
0,333   0,002   0,643   0,333   0,439   0,458    0,986   0,491   8000-9000
0,000   0,000   0,000   0,000   0,000   -0,002   0,886   0,248   9000-10000
0,125   0,000   1,000   0,125   0,222   0,353    0,971   0,270   10000-11000
0,250   0,002   0,333   0,250   0,286   0,286    0,959   0,345   11000-12000
0,000   0,002   0,000   0,000   0,000   -0,004   0,941   0,175   12000-13000
```

Figure 4.2.2 Result When "Cross Validation" is Selected and n=10 in Weka (2)

	0,000	0,001	0,000	0,000	0,000	-0,002	0,609	0,041	13000-14000
	0,000	0,002	0,000	0,000	0,000	-0,002	0,943	0,038	14000-15000
	0,400	0,002	0,444	0,400	0,421	0,419	0,962	0,370	15000-16000
	0,000	0,000	?	0,000	?	?	0,117	0,001	16000-17000
	0,000	0,000	0,000	0,000	0,000	-0,001	0,796	0,063	17000-18000
	0,000	0,001	0,000	0,000	0,000	-0,001	0,413	0,003	18000-19000
	0,000	0,000	?	0,000	?	?	0,077	0,001	19000-20000
	0,877	0,012	0,704	0,877	0,781	0,778	0,994	0,883	20000-21000
Weighted Avg.	0,478	0,019	?	0,478	?	?	0,890	0,480	

Figure 4.2.2 Result When "Cross Validation" is Selected and n=10 in Weka (3)

4.2.3 %90 Train, %10 Test Set

Percentage Split (Fixed or Holdout) is a re-sampling method that leave out random N% of the original data.

When the data set was 90% train and 10% test set, 87 data, 42,029% of 207 data, were classified correctly.

```
=== Evaluation on test split ===

Time taken to test model on test split: 0.01 seconds

=== Summary ===

Correctly Classified Instances          87               42.029  %
Kappa statistic                          0.388
Mean absolute error                      0.0213
Root mean squared error                  0.1104
Relative absolute error                 63.4097 %
Root relative squared error             85.2929 %
Total Number of Instances              207

=== Detailed Accuracy By Class ===
```

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0,500	0,065	0,188	0,500	0,273	0,273	0,920	0,232	250000-300000
0,000	0,020	0,000	0,000	0,000	-0,017	0,908	0,125	300000-350000
0,000	0,020	0,000	0,000	0,000	-0,028	0,880	0,181	350000-400000
0,000	0,083	0,000	0,000	0,000	-0,030	0,478	0,014	400000-450000
0,000	0,035	0,000	0,000	0,000	-0,040	0,803	0,129	450000-500000
0,000	0,005	0,000	0,000	0,000	-0,014	0,844	0,134	500000-550000
0,250	0,044	0,100	0,250	0,143	0,132	0,868	0,214	550000-600000
0,000	0,005	0,000	0,000	0,000	-0,013	0,776	0,129	600000-650000
0,167	0,075	0,063	0,167	0,091	0,058	0,734	0,068	650000-700000
0,000	0,000	?	0,000	?	?	0,793	0,106	700000-750000
0,000	0,020	0,000	0,000	0,000	-0,022	0,566	0,048	750000-800000
0,000	0,010	0,000	0,000	0,000	-0,012	0,915	0,108	800000-850000
0,000	0,010	0,000	0,000	0,000	-0,020	0,653	0,091	850000-900000
0,000	0,000	?	0,000	?	?	0,552	0,071	900000-950000
0,000	0,000	?	0,000	?	?	0,913	0,152	950000-1000000
0,000	0,000	?	0,000	?	?	0,815	0,087	1000000-1050000

Figure 4.2.3 Results with 10% of the Dataset (1)

?	0,000	?	?	?	?	?	?	1150000-1200000
?	0,000	?	?	?	?	?	?	1200000-1250000
0,500	0,015	0,250	0,500	0,333	0,345	0,971	0,571	1250000-1300000
0,000	0,000	?	0,000	?	?	1,000	1,000	1300000-1350000
0,000	0,000	?	0,000	?	?	0,898	0,149	1350000-1400000
0,000	0,000	?	0,000	?	?	0,155	0,006	1400000-1450000
0,500	0,015	0,250	0,500	0,333	0,345	0,976	0,236	1450000-1500000
0,000	0,000	?	0,000	?	?	0,778	0,092	1500000-1550000
0,000	0,005	0,000	0,000	0,000	-0,007	0,902	0,080	1550000-1600000
0,000	0,000	?	0,000	?	?	0,756	0,034	1600000-1650000
0,000	0,010	0,000	0,000	0,000	-0,007	0,612	0,012	1650000-1700000
?	0,000	?	?	?	?	?	?	1700000-1750000
0,000	0,000	?	0,000	?	?	0,844	0,063	1750000-1800000
?	0,000	?	?	?	?	?	?	1800000-1850000
?	0,000	?	?	?	?	?	?	1850000-1900000
0,000	0,000	?	0,000	?	?	0,214	0,006	1900000-1950000
?	0,000	?	?	?	?	?	?	1950000-2000000
0,545	0,082	0,273	0,545	0,364	0,338	0,914	0,381	2000000-2050000
1,000	0,000	1,000	1,000	1,000	1,000	1,000	1,000	500-1000
1,000	0,000	1,000	1,000	1,000	1,000	1,000	1,000	1000-2000
0,935	0,023	0,879	0,935	0,906	0,890	0,968	0,973	2000-3000
0,917	0,005	0,917	0,917	0,917	0,912	1,000	1,000	3000-4000
0,875	0,000	1,000	0,875	0,933	0,933	0,996	0,946	4000-5000
1,000	0,005	0,667	1,000	0,800	0,815	1,000	1,000	5000-6000
0,833	0,000	1,000	0,833	0,909	0,911	1,000	1,000	6000-7000
0,000	0,005	0,000	0,000	0,000	-0,007	0,988	0,367	7000-8000
0,000	0,005	0,000	0,000	0,000	-0,007	0,990	0,450	8000-9000
?	0,000	?	?	?	?	?	?	9000-10000
?	0,000	?	?	?	?	?	?	10000-11000
0,000	0,010	0,000	0,000	0,000	-0,007	0,981	0,200	11000-12000
0,000	0,000	?	0,000	?	?	0,998	0,833	12000-13000
?	0,000	?	?	?	?	?	?	13000-14000
?	0,000	?	?	?	?	?	?	14000-15000
?	0,000	?	?	?	?	?	?	15000-16000

Figure 4.2.3 Results with 10% of the Dataset (2)

	?	0,000	?	?	?	?	?	?	16000-17000
	?	0,000	?	?	?	?	?	?	17000-18000
	0,000	0,015	0,000	0,000	0,000	-0,008	0,908	0,050	18000-19000
	?	0,000	?	?	?	?	?	?	19000-20000
	0,500	0,010	0,333	0,500	0,400	0,401	0,983	0,611	20000-21000
Weighted Avg.	0,420	0,018	?	0,420	?	?	0,878	0,484	

Figure 4.2.3 Results with 10% of the Dataset (3)

By splitting 90% of the data set, the actual values in the price class in the data set and the estimated price class values as a result of the Naive Bayes algorithm were compared.

```
Test mode:    split 90.0% train, remainder test

=== Predictions on test split ===

    inst#     actual  predicted error prediction
        1 7:550000-600000 1:250000-300000   +   0.466
        2 39:2000-3000 39:2000-3000       0.989
        3 21:1250000-1300000 21:1250000-1300000       0.425
        4 10:700000-750000 7:550000-600000   +   0.717
        5 3:350000-400000 1:250000-300000   +   0.788
        6 38:1000-2000 38:1000-2000       0.994
        7 39:2000-3000 39:2000-3000       0.808
        8 13:850000-900000 5:450000-500000   +   0.279
        9 10:700000-750000 7:550000-600000   +   0.303
       10 13:850000-900000 1:250000-300000   +   0.408
       11 24:1400000-1450000 36:2000000-2050000   +   0.623
       12 39:2000-3000 39:2000-3000       0.994
       13 5:450000-500000 4:400000-450000   +   0.176
       14 16:1000000-1050000 36:2000000-2050000   +   0.552
       15 38:1000-2000 38:1000-2000       0.995
       16 36:2000000-2050000 36:2000000-2050000       0.769
       17 10:700000-750000 9:650000-700000   +   0.473
       18 41:4000-5000 41:4000-5000       0.883
       19 36:2000000-2050000 36:2000000-2050000       0.903
       20 29:1650000-1700000 4:400000-450000   +   0.255
       21 2:300000-350000 1:250000-300000   +   0.547
       22 41:4000-5000 41:4000-5000       0.835
       23 14:900000-950000 11:750000-800000   +   0.234
       24 38:1000-2000 38:1000-2000       0.982
       25 8:600000-650000 1:250000-300000   +   0.469
       26 39:2000-3000 39:2000-3000       0.993
       27 3:350000-400000 5:450000-500000   +   0.533
       28 28:1600000-1650000 36:2000000-2050000   +   0.424
       29 14:900000-950000 48:11000-12000   +   0.536
```

Figure 4.2.3 Classes in the Data Set of The Price Class of the Data and in the Algorithm Result (1)

```
      207 40:3000-4000 57:20000-21000   +   0.378

=== Evaluation on test split ===

Time taken to test model on test split: 0.05 seconds

=== Summary ===

Correctly Classified Instances          87               42.029  %
Kappa statistic                          0.388
Mean absolute error                      0.0213
Root mean squared error                  0.1104
Relative absolute error                 63.4097 %
Root relative squared error             85.2929 %
Total Number of Instances              207
```

Figure 4.2.3 How Many Data are Correctly Classified (2)

4.2.4 Supplied Test Set

4.2.4.1 Test 1

Test data: Satilik,4+2,300,2,5,Merkezi,5,Bilinmiyor,Esyali,4,Betonarme,Ikinci-El, Bos, 1500, Hayir,Kuzey-Dogu,20000-21000,Dogalgaz, ?

In the test file, the 20000-21000 class was suggested at a rate of 0.987 for this data set.

```
=== Re-evaluation on test set ===

User supplied test set
Relation:     temp
Instances:     unknown (yet). Reading incrementally
Attributes:   19

=== Predictions on user test set ===

    inst#     actual  predicted error prediction
        1        1:? 57:20000-21000       0.987

=== Summary ===

Total Number of Instances                0
Ignored Class Unknown Instances                  1
```

Figure 4.2.4.1 Class Prediction in the Algorithm's Test-1 Data

4.2.4.2 Test 2

Test data: Kiralik,2+1,300,2, 5, Kombi, 5, Bilinmiyor, Esyali-Degil, 4, Betonarme, Ikinci-El, Bos, 30, Hayir, Kuzey-Dogu,20000-21000,Dogalgaz,?

In the test file, the 20000-21000 class was suggested at a rate of 0.992 for this data set.

```
=== Re-evaluation on test set ===

User supplied test set
Relation:     temp
Instances:     unknown (yet). Reading incrementally
Attributes:   19

=== Predictions on user test set ===

    inst#     actual  predicted error prediction
        1        1:? 57:20000-21000       0.992

=== Summary ===

Total Number of Instances                0
Ignored Class Unknown Instances                  1
```

Figure 4.2.4.2 Class Prediction in the Algorithm's Test-2 Data

4.2.5 Post-Classification Estimation Plot of the Data Set

In the price classification of the data in the data set, the data on the diagonal of the graph are classified correctly, as in the chart below, the ones other than the diagonal are classified incorrectly in the data set.

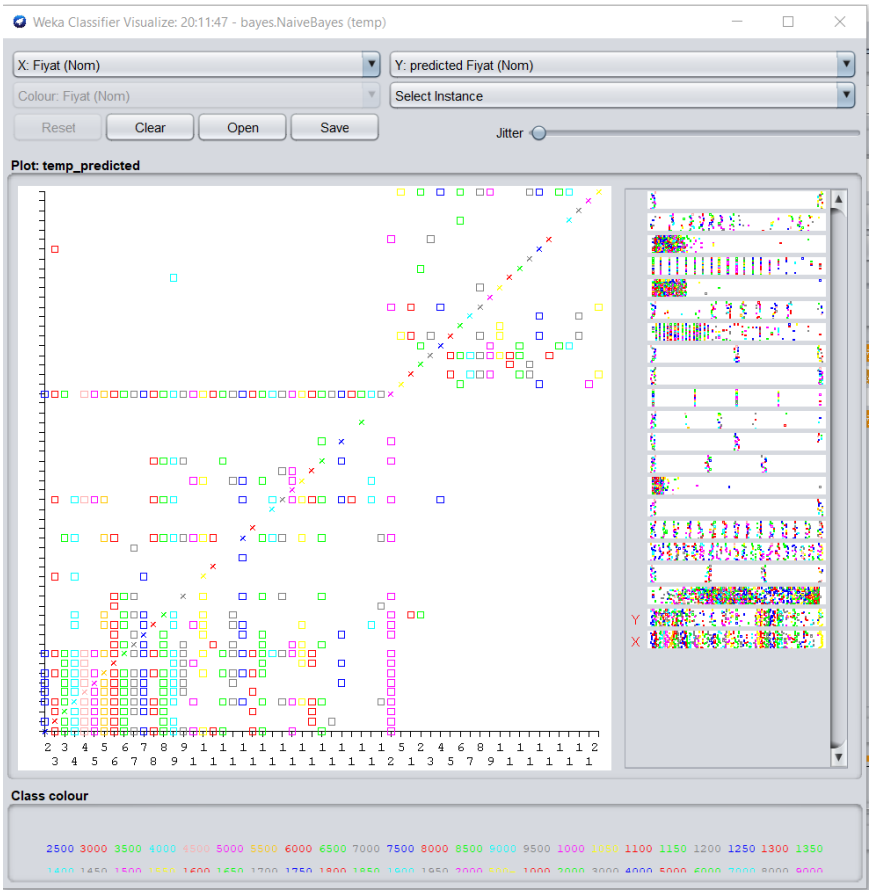

Figure 4.2.5 Post-Classification Estimation Plot of the Data Set

For example, when a point on the diagonal is selected, it is seen that the price is correctly classified.

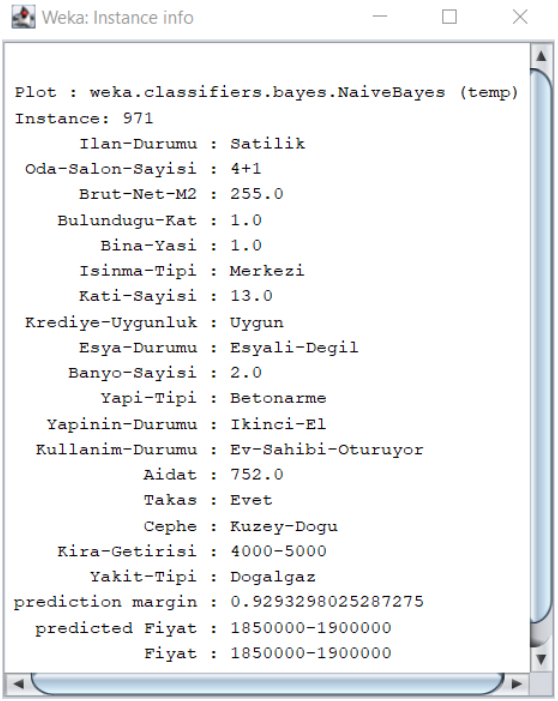

Figure 4.2.5 Correctly Classified Data

For example, when a point other than the diagonal is selected, it has been observed that the price is incorrectly classified.

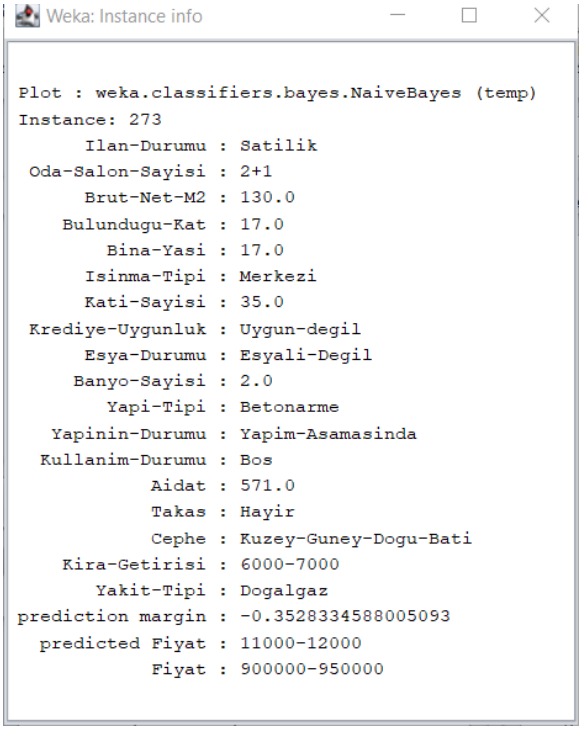

Figure 4.2.5 Misclassified Data

4.2.6 Weka in C#

When the Naive Bayes algorithm is run in the form application made in the C# programming language, a screen like the one in the figure is shown. Input values are requested from the user and the price class is suggested with the Naive Bayes algorithm.

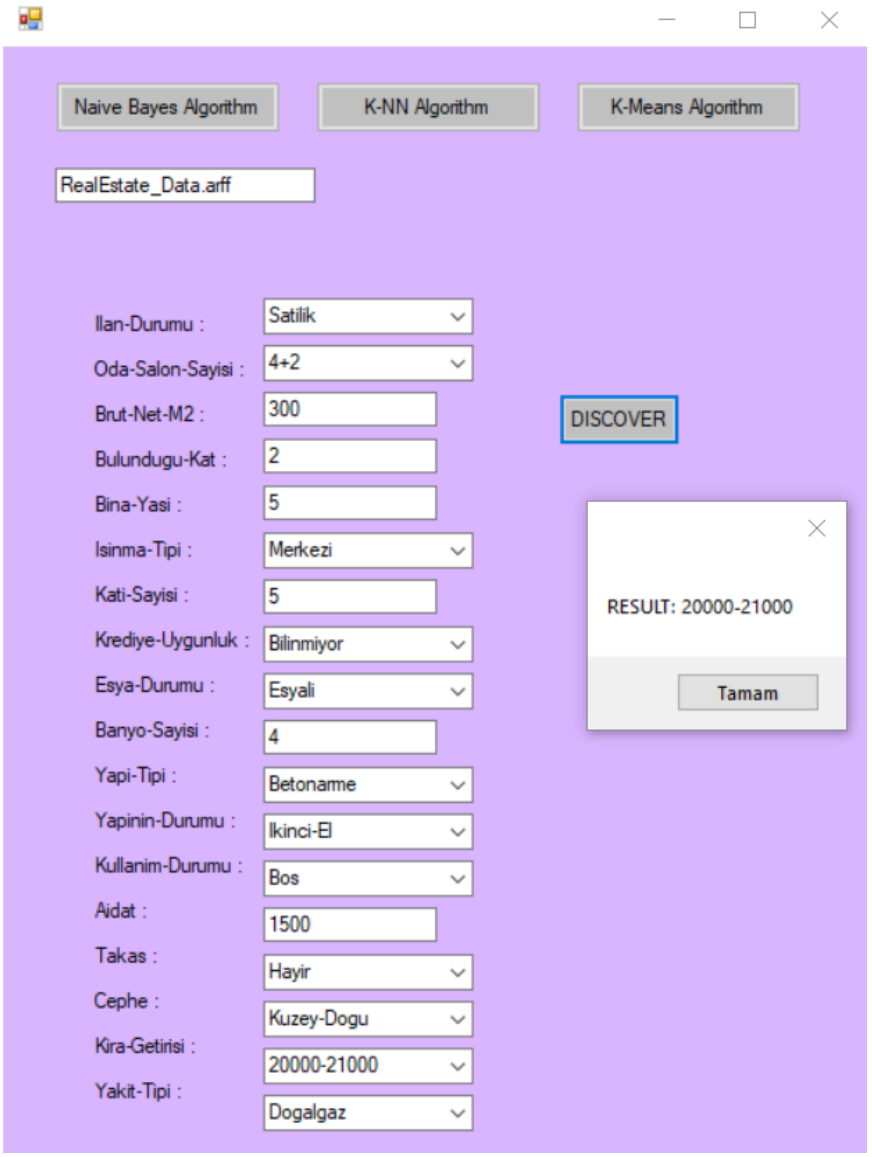

Figure 4.2.6 Naive Bayes Classification in C#

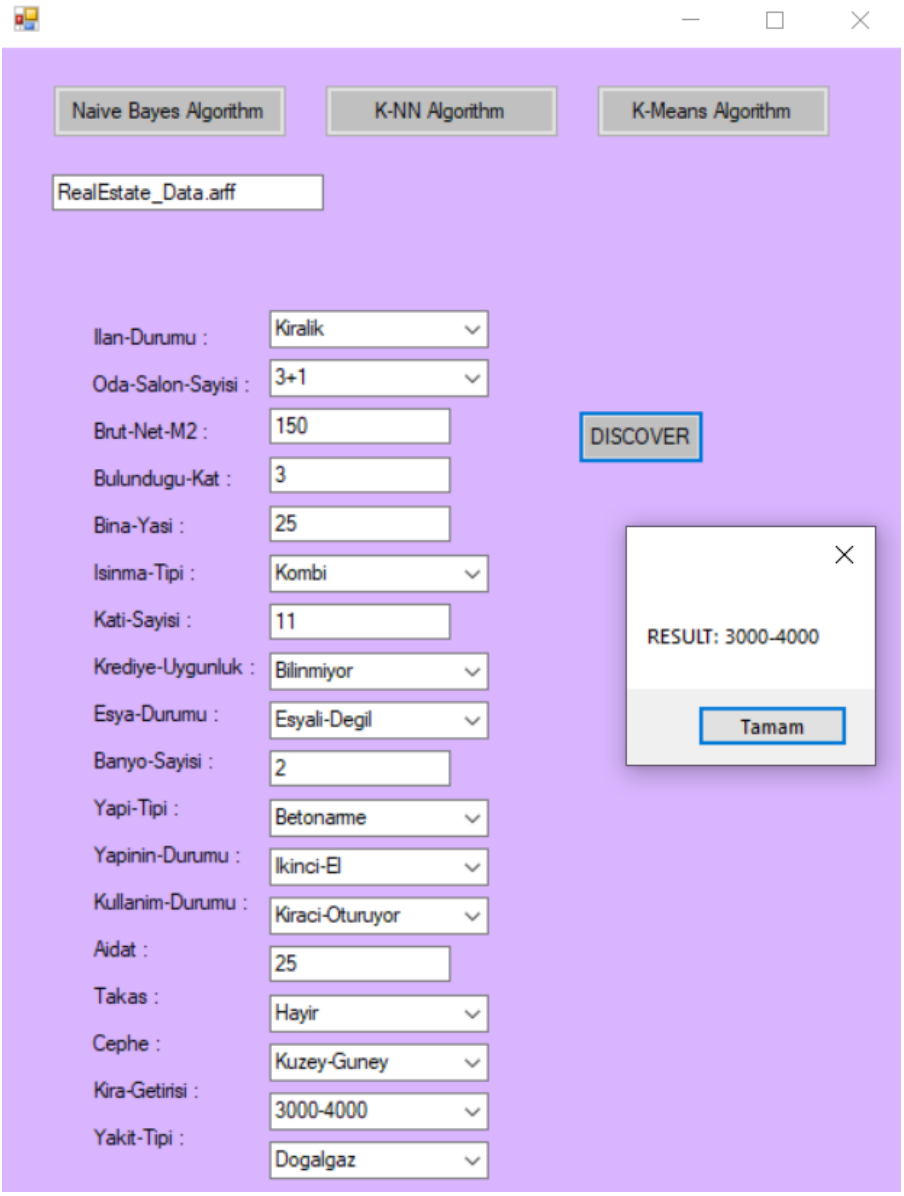

Figure 4.2.6 Naive Bayes Classification in C#

4.3 K-Nearest Neighbor

4.3.1 When k is 5 and Euclidean Distance is Selected

4.3.1.1 Use Training Set

By choosing k = 5 and Euclidean Distance in the KNN algorithm, 60.4257% of the 2067 data in the data set, 1249 data, were classified correctly.

```
Test mode:    evaluate on training data

=== Classifier model (full training set) ===

IB1 instance-based classifier
using 5 nearest neighbour(s) for classification


Time taken to build model: 0 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0.38 seconds

=== Summary ===

Correctly Classified Instances        1249               60.4257 %
Kappa statistic                          0.5822
Mean absolute error                      0.0173
Root mean squared error                  0.0902
Relative absolute error                 51.754  %
Root relative squared error             69.7411 %
Total Number of Instances             2067

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0,822    0,076    0,330      0,822    0,471      0,490    0,965     0,529     250000-300000
               0,487    0,027    0,260      0,487    0,339      0,339    0,976     0,301     300000-350000
               0,400    0,024    0,290      0,400    0,336      0,321    0,971     0,346     350000-400000
               0,528    0,035    0,352      0,528    0,422      0,406    0,956     0,358     400000-450000
               0,418    0,019    0,431      0,418    0,424      0,405    0,967     0,382     450000-500000
               0,192    0,015    0,244      0,192    0,215      0,199    0,953     0,224     500000-550000
```

Figure 4.3.1.1 Result when “Use Training Set” Option is Selected in Weka (1)

0,379	0,025	0,329	0,379	0,352	0,330	0,955	0,325	550000-600000
0,293	0,011	0,353	0,293	0,320	0,309	0,974	0,341	600000-650000
0,423	0,022	0,411	0,423	0,417	0,396	0,960	0,401	650000-700000
0,250	0,012	0,306	0,250	0,275	0,262	0,967	0,259	700000-750000
0,233	0,009	0,345	0,233	0,278	0,271	0,964	0,253	750000-800000
0,300	0,005	0,474	0,300	0,367	0,370	0,981	0,303	800000-850000
0,167	0,010	0,276	0,167	0,208	0,200	0,965	0,274	850000-900000
0,182	0,004	0,333	0,182	0,235	0,240	0,981	0,266	900000-950000
0,083	0,002	0,286	0,083	0,129	0,149	0,985	0,292	950000-1000000
0,300	0,001	0,600	0,300	0,400	0,422	0,995	0,365	1000000-1050000
0,412	0,005	0,412	0,412	0,412	0,407	0,991	0,347	1050000-1100000
0,111	0,000	1,000	0,111	0,200	0,333	0,995	0,354	1100000-1150000
0,200	0,002	0,429	0,200	0,273	0,289	0,989	0,282	1150000-1200000
0,143	0,001	0,500	0,143	0,222	0,265	0,990	0,343	1200000-1250000
0,400	0,005	0,500	0,400	0,444	0,441	0,985	0,388	1250000-1300000
0,000	0,000	?	0,000	?	?	0,993	0,194	1300000-1350000
0,063	0,001	0,250	0,063	0,100	0,122	0,987	0,243	1350000-1400000
0,000	0,000	?	0,000	?	?	0,995	0,222	1400000-1450000
0,188	0,007	0,176	0,188	0,182	0,175	0,987	0,275	1450000-1500000
0,250	0,000	0,750	0,250	0,375	0,431	0,994	0,388	1500000-1550000
0,294	0,001	0,714	0,294	0,417	0,456	0,991	0,423	1550000-1600000
0,000	0,000	?	0,000	?	?	0,989	0,214	1600000-1650000
0,000	0,001	0,000	0,000	0,000	-0,003	0,983	0,195	1650000-1700000
0,000	0,000	?	0,000	?	?	0,997	0,235	1700000-1750000
0,000	0,000	0,000	0,000	0,000	-0,001	0,992	0,171	1750000-1800000
0,000	0,000	?	0,000	?	?	0,998	0,286	1800000-1850000
0,000	0,000	?	0,000	?	?	0,999	0,250	1850000-1900000
0,000	0,000	?	0,000	?	?	0,995	0,214	1900000-1950000
0,000	0,000	?	0,000	?	?	0,998	0,333	1950000-2000000
0,612	0,022	0,632	0,612	0,622	0,599	0,975	0,644	2000000-2050000
0,852	0,002	0,821	0,852	0,836	0,834	0,998	0,834	500-1000
1,000	0,017	0,858	1,000	0,923	0,918	1,000	0,994	1000-2000
0,992	0,025	0,842	0,992	0,910	0,902	1,000	0,996	2000-3000

Figure 4.3.1.1 Result When “Use Training Set” Option is Selected in Weka (2)

	0,980	0,015	0,837	0,980	0,903	0,898	0,999	0,988	3000-4000
	0,825	0,005	0,868	0,825	0,846	0,840	0,998	0,957	4000-5000
	0,729	0,001	0,946	0,729	0,824	0,827	0,997	0,879	5000-6000
	0,667	0,000	0,960	0,667	0,787	0,797	0,998	0,887	6000-7000
	0,640	0,000	0,941	0,640	0,762	0,774	0,998	0,832	7000-8000
	0,593	0,000	1,000	0,593	0,744	0,768	0,998	0,847	8000-9000
	0,300	0,000	1,000	0,300	0,462	0,547	0,996	0,520	9000-10000
	0,375	0,000	1,000	0,375	0,545	0,612	0,999	0,768	10000-11000
	0,500	0,000	1,000	0,500	0,667	0,706	0,996	0,558	11000-12000
	0,438	0,000	1,000	0,438	0,609	0,660	0,997	0,738	12000-13000
	0,000	0,000	?	0,000	?	?	0,999	0,545	13000-14000
	0,333	0,000	1,000	0,333	0,500	0,577	0,999	0,714	14000-15000
	0,000	0,000	?	0,000	?	?	0,999	0,738	15000-16000
	0,000	0,000	?	0,000	?	?	0,998	0,231	16000-17000
	0,200	0,000	1,000	0,200	0,333	0,447	0,998	0,467	17000-18000
	0,250	0,000	1,000	0,250	0,400	0,500	1,000	0,750	18000-19000
	0,000	0,000	?	0,000	?	?	0,999	0,400	19000-20000
	0,923	0,002	0,923	0,923	0,923	0,921	0,999	0,971	20000-21000
Weighted Avg.	0,604	0,017	?	0,604	?	?	0,984	0,635	

Figure 4.3.1.1 Result When “Use Training Set” Option is Selected in Weka (3)

With the use training set, the actual values in the price class in the data set and the estimated price class values as a result of the K-NN Algorithm k=5 were compared.

```
=== Predictions on training set ===

    inst#     actual  predicted error prediction
        1 36:2000000-2050000 14:900000-950000   +   0.222
        2 19:1150000-1200000 5:450000-500000   +   0.222
        3 14:900000-950000 36:2000000-2050000   +   0.222
        4 14:900000-950000 4:400000-450000   +   0.222
        5 36:2000000-2050000 36:2000000-2050000       0.332
        6 4:400000-450000 1:250000-300000   +   0.222
        7 36:2000000-2050000 36:2000000-2050000       0.443
        8 36:2000000-2050000 36:2000000-2050000       0.332
        9 36:2000000-2050000 36:2000000-2050000       0.554
       10 36:2000000-2050000 36:2000000-2050000       0.332
       11 25:1450000-1500000 25:1450000-1500000       0.332
       12 26:1500000-1550000 26:1500000-1550000       0.332
       13 36:2000000-2050000 36:2000000-2050000       0.443
       14 5:450000-500000 5:450000-500000       0.222
       15 27:1550000-1600000 27:1550000-1600000       0.332
       16 34:1900000-1950000 27:1550000-1600000   +   0.443
       17 4:400000-450000 1:250000-300000   +   0.222
       18 1:250000-300000 11:750000-800000   +   0.222
       19 5:450000-500000 1:250000-300000   +   0.332
       20 23:1350000-1400000 5:450000-500000   +   0.332
       21 21:1250000-1300000 25:1450000-1500000   +   0.443
       22 36:2000000-2050000 13:850000-900000   +   0.222
       23 4:400000-450000 9:650000-700000   +   0.222
       24 1:250000-300000 1:250000-300000       0.332
       25 1:250000-300000 12:800000-850000   +   0.222
       26 27:1550000-1600000 27:1550000-1600000       0.443
       27 1:250000-300000 7:550000-600000   +   0.222
       28 13:850000-900000 5:450000-500000   +   0.111
       29 15:950000-1000000 6:500000-550000   +   0.222
       30 36:2000000-2050000 27:1550000-1600000   +   0.222
       31 8:600000-650000 1:250000-300000   +   0.222
```

Figure 4.3.1.1 Class Comparison in Dataset (1)

```
     2067 37:500-1000 40:3000-4000   +   0.443

=== Evaluation on training set ===

Time taken to test model on training data: 1.02 seconds

=== Summary ===

Correctly Classified Instances        1155               55.8781 %
Kappa statistic                          0.5332
Mean absolute error                      0.0203
Root mean squared error                  0.0969
Relative absolute error                 60.5272 %
Root relative squared error             74.9166 %
Total Number of Instances             2067
```

Figure 4.3.1.1 Class Comparison in Dataset (2)

4.3.1.2 Cross Validation

When the n value of the Cross Validation is 10, 44.1219 % of the 2067 data, 912 data were classified correctly.

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         912               44.1219 %
Kappa statistic                          0.4084
Mean absolute error                      0.0221
Root mean squared error                  0.111
Relative absolute error                 65.9406 %
Root relative squared error             85.8216 %
Total Number of Instances             2067
```

Figure 4.3.1.2 Result When "Cross Validation" is Selected and n=10 in Weka (1)

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0,456	0,098	0,174	0,456	0,252	0,230	0,755	0,193	250000-300000
0,051	0,032	0,030	0,051	0,038	0,015	0,650	0,044	300000-350000
0,120	0,047	0,059	0,120	0,079	0,052	0,676	0,086	350000-400000
0,222	0,047	0,147	0,222	0,177	0,144	0,698	0,103	400000-450000
0,104	0,032	0,099	0,104	0,101	0,070	0,719	0,099	450000-500000
0,000	0,026	0,000	0,000	0,000	-0,026	0,547	0,030	500000-550000
0,212	0,032	0,179	0,212	0,194	0,166	0,645	0,112	550000-600000
0,024	0,011	0,043	0,024	0,031	0,018	0,585	0,032	600000-650000
0,099	0,029	0,109	0,099	0,104	0,074	0,702	0,094	650000-700000
0,068	0,016	0,086	0,068	0,076	0,059	0,582	0,038	700000-750000
0,023	0,011	0,042	0,023	0,030	0,016	0,561	0,028	750000-800000
0,100	0,010	0,130	0,100	0,113	0,103	0,639	0,050	800000-850000
0,021	0,014	0,034	0,021	0,026	0,009	0,562	0,033	850000-900000
0,000	0,004	0,000	0,000	0,000	-0,006	0,589	0,018	900000-950000
0,000	0,001	0,000	0,000	0,000	-0,003	0,512	0,015	950000-1000000
0,000	0,000	?	0,000	?	?	0,579	0,014	1000000-1050000
0,118	0,002	0,286	0,118	0,167	0,179	0,712	0,076	1050000-1100000
0,000	0,001	0,000	0,000	0,000	-0,002	0,517	0,007	1100000-1150000
0,000	0,004	0,000	0,000	0,000	-0,005	0,485	0,010	1150000-1200000
0,000	0,001	0,000	0,000	0,000	-0,003	0,577	0,010	1200000-1250000
0,120	0,004	0,250	0,120	0,162	0,166	0,652	0,106	1250000-1300000
0,000	0,000	?	0,000	?	?	0,430	0,003	1300000-1350000
0,000	0,001	0,000	0,000	0,000	-0,003	0,501	0,008	1350000-1400000
0,000	0,000	?	0,000	?	?	0,478	0,003	1400000-1450000
0,125	0,007	0,118	0,125	0,121	0,114	0,621	0,048	1450000-1500000
0,250	0,001	0,600	0,250	0,353	0,385	0,525	0,154	1500000-1550000
0,059	0,002	0,167	0,059	0,087	0,095	0,649	0,033	1550000-1600000
0,000	0,000	0,000	0,000	0,000	-0,002	0,512	0,006	1600000-1650000
0,000	0,002	0,000	0,000	0,000	-0,004	0,499	0,008	1650000-1700000
0,000	0,000	?	0,000	?	?	0,645	0,003	1700000-1750000
0,000	0,001	0,000	0,000	0,000	-0,002	0,619	0,007	1750000-1800000

Figure 4.3.1.2 Result When "Cross Validation" is Selected and n=10 in Weka (2)

	0,000	0,000	?	0,000	?	?	0,524	0,002	1800000-1850000
	0,000	0,000	?	0,000	?	?	0,648	0,001	1850000-1900000
	0,000	0,000	?	0,000	?	?	0,397	0,002	1900000-1950000
	0,000	0,000	?	0,000	?	?	0,548	0,003	1950000-2000000
	0,405	0,032	0,441	0,405	0,422	0,389	0,814	0,363	2000000-2050000
	0,593	0,002	0,762	0,593	0,667	0,668	0,895	0,683	500-1000
	1,000	0,028	0,788	1,000	0,881	0,875	0,998	0,978	1000-2000
	0,983	0,042	0,757	0,983	0,856	0,843	0,998	0,984	2000-3000
	0,959	0,026	0,738	0,959	0,834	0,828	0,997	0,962	3000-4000
	0,650	0,007	0,788	0,650	0,712	0,705	0,960	0,833	4000-5000
	0,521	0,002	0,833	0,521	0,641	0,653	0,918	0,674	5000-6000
	0,444	0,001	0,842	0,444	0,582	0,607	0,920	0,641	6000-7000
	0,240	0,000	0,857	0,240	0,375	0,451	0,937	0,499	7000-8000
	0,296	0,000	1,000	0,296	0,457	0,542	0,841	0,490	8000-9000
	0,000	0,000	?	0,000	?	?	0,757	0,163	9000-10000
	0,125	0,000	1,000	0,125	0,222	0,353	0,802	0,245	10000-11000
	0,250	0,001	0,400	0,250	0,308	0,314	0,862	0,233	11000-12000
	0,063	0,000	1,000	0,063	0,118	0,249	0,736	0,198	12000-13000
	0,000	0,000	?	0,000	?	?	0,649	0,004	13000-14000
	0,167	0,000	1,000	0,167	0,286	0,408	0,823	0,240	14000-15000
	0,000	0,000	?	0,000	?	?	0,688	0,179	15000-16000
	0,000	0,000	?	0,000	?	?	0,646	0,002	16000-17000
	0,000	0,000	?	0,000	?	?	0,589	0,095	17000-18000
	0,000	0,000	?	0,000	?	?	0,737	0,065	18000-19000
	0,000	0,000	?	0,000	?	?	0,152	0,001	19000-20000
	0,785	0,002	0,911	0,785	0,843	0,841	0,991	0,905	20000-21000
Weighted Avg.	0,441	0,025	?	0,441	?	?	0,793	0,444	

Figure 4.3.1.2 Result When "Cross Validation" is Selected and n=10 in Weka (3)

4.3.1.3 %90 Train, %10 Test Set

When the data set was 90% train and 10% test set, 85 data, 41,0628% of 207 data, were classified correctly.

```
Time taken to test model on test split: 0.04 seconds

=== Summary ===

Correctly Classified Instances          85               41.0628 %
Kappa statistic                          0.3763
Mean absolute error                      0.0232
Root mean squared error                  0.1148
Relative absolute error                 69.2954 %
Root relative squared error             88.733  %
Total Number of Instances              207
```

Figure 4.3.1.3 Results with 10% of the Dataset (1)

```
=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0,667    0,095    0,174      0,667    0,276      0,305    0,771     0,144     250000-300000
               0,333    0,029    0,143      0,333    0,200      0,201    0,788     0,079     300000-350000
               0,000    0,030    0,000      0,000    0,000      -0,035   0,702     0,110     350000-400000
               0,000    0,073    0,000      0,000    0,000      -0,028   0,398     0,010     400000-450000
               0,111    0,030    0,143      0,111    0,125      0,091    0,618     0,080     450000-500000
               0,000    0,030    0,000      0,000    0,000      -0,035   0,493     0,038     500000-550000
               0,250    0,059    0,077      0,250    0,118      0,108    0,680     0,273     550000-600000
               0,143    0,010    0,333      0,143    0,200      0,201    0,689     0,219     600000-650000
               0,167    0,035    0,125      0,167    0,143      0,115    0,584     0,049     650000-700000
               0,000    0,000    ?          0,000    ?          ?        0,452     0,039     700000-750000
               0,000    0,030    0,000      0,000    0,000      -0,027   0,455     0,024     750000-800000
               0,000    0,010    0,000      0,000    0,000      -0,012   0,463     0,014     800000-850000
               0,000    0,015    0,000      0,000    0,000      -0,024   0,504     0,039     850000-900000
               0,000    0,000    ?          0,000    ?          ?        0,600     0,037     900000-950000
               0,000    0,005    0,000      0,000    0,000      -0,008   0,639     0,037     950000-1000000
               0,000    0,000    ?          0,000    ?          ?        0,498     0,019     1000000-1050000
               ?        0,005    0,000      ?        ?          ?        ?         ?         1050000-1100000
               0,000    0,000    ?          0,000    ?          ?        0,995     0,333     1100000-1150000
               ?        0,010    0,000      ?        ?          ?        ?         ?         1150000-1200000
               ?        0,005    0,000      ?        ?          ?        ?         ?         1200000-1250000
               0,000    0,010    0,000      0,000    0,000      -0,010   0,717     0,041     1250000-1300000
               0,000    0,000    ?          0,000    ?          ?        0,985     0,143     1300000-1350000
               0,000    0,000    ?          0,000    ?          ?        0,493     0,010     1350000-1400000
               0,000    0,000    ?          0,000    ?          ?        0,490     0,005     1400000-1450000
               0,000    0,005    0,000      0,000    0,000      -0,007   0,732     0,067     1450000-1500000
               0,000    0,000    ?          0,000    ?          ?        0,493     0,014     1500000-1550000
               0,500    0,000    1,000      0,500    0,667      0,705    0,740     0,505     1550000-1600000
               0,000    0,000    ?          0,000    ?          ?        0,490     0,010     1600000-1650000
               0,000    0,005    0,000      0,000    0,000      -0,005   0,478     0,005     1650000-1700000
               ?        0,000    ?          ?        ?          ?        ?         ?         1700000-1750000
```

Figure 4.3.1.3 Results with 10% of the Dataset (2)

	0,000	0,000	?	0,000	?	?	0,493	0,010	1750000-1800000
	?	0,000	?	?	?	?	?	?	1800000-1850000
	?	0,000	?	?	?	?	?	?	1850000-1900000
	0,000	0,000	?	0,000	?	?	0,490	0,005	1900000-1950000
	?	0,000	?	?	?	?	?	?	1950000-2000000
	0,364	0,031	0,400	0,364	0,381	0,348	0,772	0,249	2000000-2050000
	0,500	0,000	1,000	0,500	0,667	0,705	0,746	0,505	500-1000
	1,000	0,021	0,810	1,000	0,895	0,890	0,998	0,973	1000-2000
	0,968	0,028	0,857	0,968	0,909	0,894	0,997	0,980	2000-3000
	1,000	0,021	0,750	1,000	0,857	0,857	0,996	0,901	3000-4000
	0,625	0,005	0,833	0,625	0,714	0,712	0,996	0,884	4000-5000
	1,000	0,000	1,000	1,000	1,000	1,000	1,000	1,000	5000-6000
	0,333	0,000	1,000	0,333	0,500	0,572	0,910	0,616	6000-7000
	0,000	0,000	?	0,000	?	?	0,994	0,643	7000-8000
	0,500	0,000	1,000	0,500	0,667	0,705	0,995	0,667	8000-9000
	?	0,000	?	?	?	?	?	?	9000-10000
	?	0,005	0,000	?	?	?	?	?	10000-11000
	0,000	0,000	?	0,000	?	?	0,998	0,500	11000-12000
	0,000	0,000	?	0,000	?	?	0,745	0,171	12000-13000
	?	0,000	?	?	?	?	?	?	13000-14000
	?	0,000	?	?	?	?	?	?	14000-15000
	?	0,000	?	?	?	?	?	?	15000-16000
	?	0,000	?	?	?	?	?	?	16000-17000
	?	0,000	?	?	?	?	?	?	17000-18000
	0,000	0,000	?	0,000	?	?	0,500	0,005	18000-19000
	?	0,000	?	?	?	?	?	?	19000-20000
	0,500	0,010	0,333	0,500	0,400	0,401	0,998	0,750	20000-21000
Weighted Avg.	0,411	0,021	?	0,411	?	?	0,763	0,426	

Figure 4.3.1.3 Results with 10% of the Dataset (3)

4.3.1.4 Supplied Test Set

Test data: Satilik,5+1,300,9,11,Merkezi,30,Uygun,Esyali-Degil,2,Betonarme,Ikinci-El, Kiraci-Oturuyor,10,Hayir,Guney-Dogu-Bati,20000-21000,Dogalgaz, ?

In the test file, the 2000000-2050000 class was suggested at a rate of 0.995 for this data set.

```
=== Predictions on user test set ===

    inst#     actual  predicted error prediction
        1        1:? 36:2000000-2050000       0.995

=== Summary ===

Total Number of Instances                0
Ignored Class Unknown Instances                  1
```

Figure 4.3.1.4 Class Prediction in the Algorithm's Test Data

4.3.2 When k is 9 and Manhattan Distance is Selected

4.3.2.1 Use Training Set

By choosing k = 9 and Manhattan Distance in the KNN algorithm, 55.8781% of the 2067 data in the data set, 1155 data, were classified correctly.

```
=== Evaluation on training set ===

Time taken to test model on training data: 1.18 seconds

=== Summary ===

Correctly Classified Instances        1155               55.8781 %
Kappa statistic                          0.5332
Mean absolute error                      0.0203
Root mean squared error                  0.0969
Relative absolute error                 60.5272 %
Root relative squared error             74.9166 %
Total Number of Instances             2067
```

Figure 4.3.2.1 Result When "Use Training Set" Option is Selected in Weka (1)

```
=== Detailed Accuracy By Class ===
```

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0,733	0,073	0,313	0,733	0,439	0,445	0,953	0,440	250000-300000
0,282	0,019	0,220	0,282	0,247	0,233	0,961	0,215	300000-350000
0,300	0,021	0,263	0,300	0,280	0,262	0,952	0,275	350000-400000
0,333	0,037	0,245	0,333	0,282	0,256	0,921	0,231	400000-450000
0,418	0,021	0,406	0,418	0,412	0,392	0,949	0,329	450000-500000
0,135	0,018	0,159	0,135	0,146	0,126	0,919	0,188	500000-550000
0,439	0,030	0,322	0,439	0,372	0,352	0,928	0,247	550000-600000
0,195	0,011	0,267	0,195	0,225	0,215	0,956	0,253	600000-650000
0,394	0,029	0,326	0,394	0,357	0,333	0,937	0,272	650000-700000
0,136	0,010	0,222	0,136	0,169	0,160	0,940	0,178	700000-750000
0,233	0,010	0,323	0,233	0,270	0,261	0,941	0,209	750000-800000
0,200	0,007	0,300	0,200	0,240	0,236	0,967	0,208	800000-850000
0,229	0,010	0,355	0,229	0,278	0,272	0,944	0,224	850000-900000
0,182	0,001	0,667	0,182	0,286	0,345	0,966	0,207	900000-950000
0,000	0,002	0,000	0,000	0,000	-0,005	0,974	0,189	950000-1000000
0,100	0,001	0,333	0,100	0,154	0,180	0,989	0,180	1000000-1050000
0,176	0,005	0,214	0,176	0,194	0,188	0,982	0,220	1050000-1100000
0,000	0,000	?	0,000	?	?	0,990	0,201	1100000-1150000
0,067	0,002	0,200	0,067	0,100	0,112	0,979	0,153	1150000-1200000
0,071	0,001	0,250	0,071	0,111	0,131	0,980	0,188	1200000-1250000
0,280	0,006	0,350	0,280	0,311	0,306	0,974	0,361	1250000-1300000
0,000	0,000	?	0,000	?	?	0,985	0,106	1300000-1350000
0,063	0,002	0,167	0,063	0,091	0,098	0,974	0,129	1350000-1400000
0,000	0,000	?	0,000	?	?	0,988	0,107	1400000-1450000
0,375	0,008	0,261	0,375	0,308	0,306	0,982	0,233	1450000-1500000
0,250	0,002	0,375	0,250	0,300	0,303	0,987	0,243	1500000-1550000
0,294	0,003	0,417	0,294	0,345	0,346	0,981	0,240	1550000-1600000
0,000	0,000	?	0,000	?	?	0,978	0,125	1600000-1650000
0,000	0,003	0,000	0,000	0,000	-0,005	0,970	0,120	1650000-1700000
0,000	0,000	?	0,000	?	?	0,992	0,114	1700000-1750000

Figure 4.3.2.1 Result When "Use Training Set" Option is Selected in Weka (2)

	0,143	0,002	0,200	0,143	0,167	0,167	0,988	0,157	1750000-1800000
	0,000	0,000	?	0,000	?	?	0,996	0,182	1800000-1850000
	0,000	0,000	?	0,000	?	?	0,998	0,222	1850000-1900000
	0,000	0,000	?	0,000	?	?	0,991	0,133	1900000-1950000
	0,000	0,000	?	0,000	?	?	0,996	0,250	1950000-2000000
	0,562	0,035	0,496	0,562	0,527	0,497	0,966	0,591	2000000-2050000
	0,630	0,001	0,895	0,630	0,739	0,748	0,997	0,858	500-1000
	1,000	0,021	0,832	1,000	0,908	0,903	1,000	0,999	1000-2000
	1,000	0,037	0,780	1,000	0,876	0,867	1,000	0,997	2000-3000
	0,986	0,020	0,788	0,986	0,876	0,872	1,000	0,993	3000-4000
	0,813	0,004	0,890	0,813	0,850	0,845	0,999	0,968	4000-5000
	0,646	0,001	0,912	0,646	0,756	0,763	0,996	0,881	5000-6000
	0,556	0,000	1,000	0,556	0,714	0,742	0,999	0,944	6000-7000
	0,480	0,000	0,923	0,480	0,632	0,663	0,998	0,882	7000-8000
	0,519	0,000	0,933	0,519	0,667	0,693	0,996	0,828	8000-9000
	0,000	0,000	?	0,000	?	?	0,995	0,453	9000-10000
	0,250	0,000	1,000	0,250	0,400	0,499	0,997	0,631	10000-11000
	0,375	0,000	0,750	0,375	0,500	0,529	0,994	0,448	11000-12000
	0,188	0,000	1,000	0,188	0,316	0,432	0,995	0,665	12000-13000
	0,000	0,000	?	0,000	?	?	0,997	0,300	13000-14000
	0,167	0,000	1,000	0,167	0,286	0,408	0,998	0,625	14000-15000
	0,100	0,000	1,000	0,100	0,182	0,316	0,997	0,554	15000-16000
	0,000	0,000	?	0,000	?	?	0,995	0,136	16000-17000
	0,000	0,000	?	0,000	?	?	0,998	0,521	17000-18000
	0,000	0,000	?	0,000	?	?	0,998	0,450	18000-19000
	0,000	0,000	?	0,000	?	?	0,999	0,250	19000-20000
	0,877	0,001	0,950	0,877	0,912	0,910	0,999	0,973	20000-21000
Weighted Avg.	0,559	0,020	?	0,559	?	?	0,975	0,589	

Figure 4.3.2.1 Result When “Use Training Set” Option is Selected in Weka (3)

```
=== Predictions on training set ===

    inst#     actual  predicted error prediction
        1 36:2000000-2050000 14:900000-950000   +   0.222
        2 19:1150000-1200000 5:450000-500000   +   0.222
        3 14:900000-950000 36:2000000-2050000   +   0.222
        4 14:900000-950000 4:400000-450000   +   0.222
        5 36:2000000-2050000 36:2000000-2050000       0.332
        6 4:400000-450000 1:250000-300000   +   0.222
        7 36:2000000-2050000 36:2000000-2050000       0.443
        8 36:2000000-2050000 36:2000000-2050000       0.332
        9 36:2000000-2050000 36:2000000-2050000       0.554
       10 36:2000000-2050000 36:2000000-2050000       0.332
       11 25:1450000-1500000 25:1450000-1500000       0.332
       12 26:1500000-1550000 26:1500000-1550000       0.332
       13 36:2000000-2050000 36:2000000-2050000       0.443
       14 5:450000-500000 5:450000-500000       0.222
       15 27:1550000-1600000 27:1550000-1600000       0.332
       16 34:1900000-1950000 27:1550000-1600000   +   0.443
       17 4:400000-450000 1:250000-300000   +   0.222
       18 1:250000-300000 11:750000-800000   +   0.222
       19 5:450000-500000 1:250000-300000   +   0.332
       20 23:1350000-1400000 5:450000-500000   +   0.332
       21 21:1250000-1300000 25:1450000-1500000   +   0.443
       22 36:2000000-2050000 13:850000-900000   +   0.222
       23 4:400000-450000 9:650000-700000   +   0.222
       24 1:250000-300000 1:250000-300000       0.332
       25 1:250000-300000 12:800000-850000   +   0.222
       26 27:1550000-1600000 27:1550000-1600000       0.443
       27 1:250000-300000 7:550000-600000   +   0.222
       28 13:850000-900000 5:450000-500000   +   0.111
```

Figure 4.3.2.1 Result When “Use Training Set” Option is Selected in Weka (4)

```
     2067 37:500-1000 40:3000-4000   +   0.443

=== Evaluation on training set ===

Time taken to test model on training data: 3.22 seconds

=== Summary ===

Correctly Classified Instances        1155               55.8781 %
Kappa statistic                          0.5332
Mean absolute error                      0.0203
Root mean squared error                  0.0969
Relative absolute error                 60.5272 %
Root relative squared error             74.9166 %
Total Number of Instances             2067
```

Figure 4.3.2.1 Result When “Use Training Set” Option is Selected in Weka (5)

4.3.2.2 Cross Validation

When the n value of the Cross Validation is 10, 44.9927 % of the 2067 data, 930 data were classified correctly.

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         930               44.9927 %
Kappa statistic                          0.4166
Mean absolute error                      0.0232
Root mean squared error                  0.109
Relative absolute error                 69.1273 %
Root relative squared error             84.2572 %
Total Number of Instances             2067

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0,578     0,091     0,224       0,578   0,323      0,315    0,817     0,225     250000-300000
               0,103     0,026     0,071       0,103   0,084      0,064    0,697     0,058     300000-350000
               0,120     0,031     0,088       0,120   0,102      0,077    0,721     0,090     350000-400000
               0,181     0,049     0,118       0,181   0,143      0,108    0,715     0,095     400000-450000
               0,134     0,032     0,123       0,134   0,129      0,098    0,737     0,106     450000-500000
               0,000     0,015     0,000       0,000   0,000      -0,020   0,574     0,036     500000-550000
               0,212     0,042     0,141       0,212   0,170      0,140    0,708     0,124     550000-600000
               0,073     0,011     0,120       0,073   0,091      0,079    0,607     0,035     600000-650000
               0,127     0,036     0,113       0,127   0,119      0,086    0,728     0,087     650000-700000
               0,068     0,012     0,107       0,068   0,083      0,070    0,598     0,033     700000-750000
               0,070     0,009     0,143       0,070   0,094      0,087    0,589     0,042     750000-800000
               0,067     0,009     0,095       0,067   0,078      0,068    0,645     0,052     800000-850000
               0,000     0,011     0,000       0,000   0,000      -0,016   0,613     0,040     850000-900000
               0,045     0,001     0,333       0,045   0,080      0,120    0,586     0,019     900000-950000
               0,000     0,003     0,000       0,000   0,000      -0,006   0,478     0,012     950000-1000000
               0,000     0,000     ?           0,000   ?          ?        0,612     0,015     1000000-1050000
               0,000     0,003     0,000       0,000   0,000      -0,005   0,770     0,055     1050000-1100000
```

Figure 4.3.2.2 Result When "Cross Validation" is Selected and n=10 in Weka

4.3.2.3 %90 Training, %10 Test

When the data set was 90% train and 10% test set, 85 data, 41,0628% of 207 data, were classified correctly.

```
=== Evaluation on test split ===

Time taken to test model on test split: 0.05 seconds

=== Summary ===

Correctly Classified Instances          85               41.0628 %
Kappa statistic                          0.3751
Mean absolute error                      0.0236
Root mean squared error                  0.1114
Relative absolute error                 70.4583 %
Root relative squared error             86.0966 %
Total Number of Instances              207
```

Figure 4.3.2.3 Results with 10% of the Dataset (1)

```
=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0,667    0,070    0,222      0,667    0,333      0,355    0,845     0,174     250000-300000
               0,000    0,029    0,000      0,000    0,000      -0,021   0,924     0,120     300000-350000
               0,250    0,045    0,182      0,250    0,211      0,176    0,755     0,165     350000-400000
               0,000    0,049    0,000      0,000    0,000      -0,022   0,373     0,010     400000-450000
               0,111    0,025    0,167      0,111    0,133      0,104    0,704     0,110     450000-500000
               0,000    0,015    0,000      0,000    0,000      -0,024   0,659     0,072     500000-550000
               0,000    0,059    0,000      0,000    0,000      -0,035   0,744     0,066     550000-600000
               0,286    0,010    0,500      0,286    0,364      0,362    0,808     0,193     600000-650000
               0,167    0,045    0,100      0,167    0,125      0,095    0,637     0,061     650000-700000
               0,000    0,005    0,000      0,000    0,000      -0,014   0,447     0,039     700000-750000
               0,000    0,025    0,000      0,000    0,000      -0,025   0,438     0,025     750000-800000
               0,000    0,020    0,000      0,000    0,000      -0,017   0,600     0,021     800000-850000
               0,000    0,010    0,000      0,000    0,000      -0,020   0,479     0,038     850000-900000
               0,000    0,010    0,000      0,000    0,000      -0,014   0,582     0,027     900000-950000
               0,000    0,010    0,000      0,000    0,000      -0,012   0,631     0,029     950000-1000000
               0,000    0,000    ?          0,000    ?          ?        0,500     0,020     1000000-1050000
               ?        0,010    0,000      ?        ?          ?        ?         ?         1050000-1100000
               0,000    0,000    ?          0,000    ?          ?        0,985     0,143     1100000-1150000
               ?        0,010    0,000      ?        ?          ?        ?         ?         1150000-1200000
               ?        0,000    ?          ?        ?          ?        ?         ?         1200000-1250000
               0,000    0,010    0,000      0,000    0,000      -0,010   0,706     0,031     1250000-1300000
               0,000    0,005    0,000      0,000    0,000      -0,005   0,973     0,100     1300000-1350000
               0,000    0,000    ?          0,000    ?          ?        0,485     0,010     1350000-1400000
               0,000    0,000    ?          0,000    ?          ?        0,488     0,005     1400000-1450000
               0,000    0,015    0,000      0,000    0,000      -0,012   0,720     0,043     1450000-1500000
               0,000    0,000    ?          0,000    ?          ?        0,495     0,015     1500000-1550000
               0,000    0,000    ?          0,000    ?          ?        0,739     0,255     1550000-1600000
               0,000    0,000    ?          0,000    ?          ?        0,488     0,010     1600000-1650000
               0,000    0,005    0,000      0,000    0,000      -0,005   0,468     0,005     1650000-1700000
               ?        0,000    ?          ?        ?          ?        ?         ?         1700000-1750000
               0,000    0,000    ?          0,000    ?          ?        0,735     0,060     1750000-1800000
```

Figure 4.3.2.3 Results with 10% of the Dataset (2)

	?	0,000	?	?	?	?	?	?	1800000-1850000
	?	0,000	?	?	?	?	?	?	1850000-1900000
	0,000	0,000	?	0,000	?	?	0,495	0,005	1900000-1950000
	?	0,000	?	?	?	?	?	?	1950000-2000000
	0,364	0,036	0,364	0,364	0,364	0,328	0,802	0,428	2000000-2050000
	0,500	0,000	1,000	0,500	0,667	0,705	0,995	0,667	500-1000
	1,000	0,021	0,810	1,000	0,895	0,890	1,000	1,000	1000-2000
	1,000	0,040	0,816	1,000	0,899	0,885	0,999	0,990	2000-3000
	1,000	0,015	0,800	1,000	0,889	0,888	1,000	0,988	3000-4000
	0,625	0,000	1,000	0,625	0,769	0,785	0,991	0,869	4000-5000
	1,000	0,005	0,667	1,000	0,800	0,815	1,000	1,000	5000-6000
	0,167	0,000	1,000	0,167	0,286	0,403	0,995	0,833	6000-7000
	0,000	0,000	?	0,000	?	?	0,991	0,611	7000-8000
	0,500	0,005	0,500	0,500	0,500	0,495	0,990	0,611	8000-9000
	?	0,000	?	?	?	?	?	?	9000-10000
	?	0,000	?	?	?	?	?	?	10000-11000
	0,000	0,005	0,000	0,000	0,000	-0,005	0,995	0,500	11000-12000
	0,000	0,000	?	0,000	?	?	0,745	0,105	12000-13000
	?	0,000	?	?	?	?	?	?	13000-14000
	?	0,000	?	?	?	?	?	?	14000-15000
	?	0,000	?	?	?	?	?	?	15000-16000
	?	0,000	?	?	?	?	?	?	16000-17000
	?	0,000	?	?	?	?	?	?	17000-18000
	0,000	0,000	?	0,000	?	?	0,510	0,005	18000-19000
	?	0,000	?	?	?	?	?	?	19000-20000
	0,500	0,005	0,500	0,500	0,500	0,495	0,993	0,667	20000-21000
Weighted Avg.	0,411	0,022	?	0,411	?	?	0,795	0,447	

Figure 4.3.2.3 Results with 10% of the Dataset (3)

4.3.2.4 Supplied Test Set

Test data: Kiralik,3+1,155,6,5,Kombi,6,Bilinmiyor,Esyali-Degil,2,Betonarme,Ikinci-El,Bos, 53, Hayir,Kuzey-Dogu-Bati,1000-2000,Dogalgaz,?

In the test file, the 20000-21000 class was suggested at a rate of 0.997 for this data set.

```
=== Predictions on user test set ===

    inst#     actual  predicted error prediction
        1        1:? 57:20000-21000       0.997

=== Summary ===

Total Number of Instances                0
Ignored Class Unknown Instances                  1
```

Figure 4.3.2.4 Class Prediction in the Algorithm's Test Data

4.3.2.5 Post-Classification Estimation Plot of the Data Set

In the price classification of the data in the data set, the data on the diagonal of the graph are classified correctly, as in the chart below, the ones other than the diagonal are classified incorrectly in the data set.

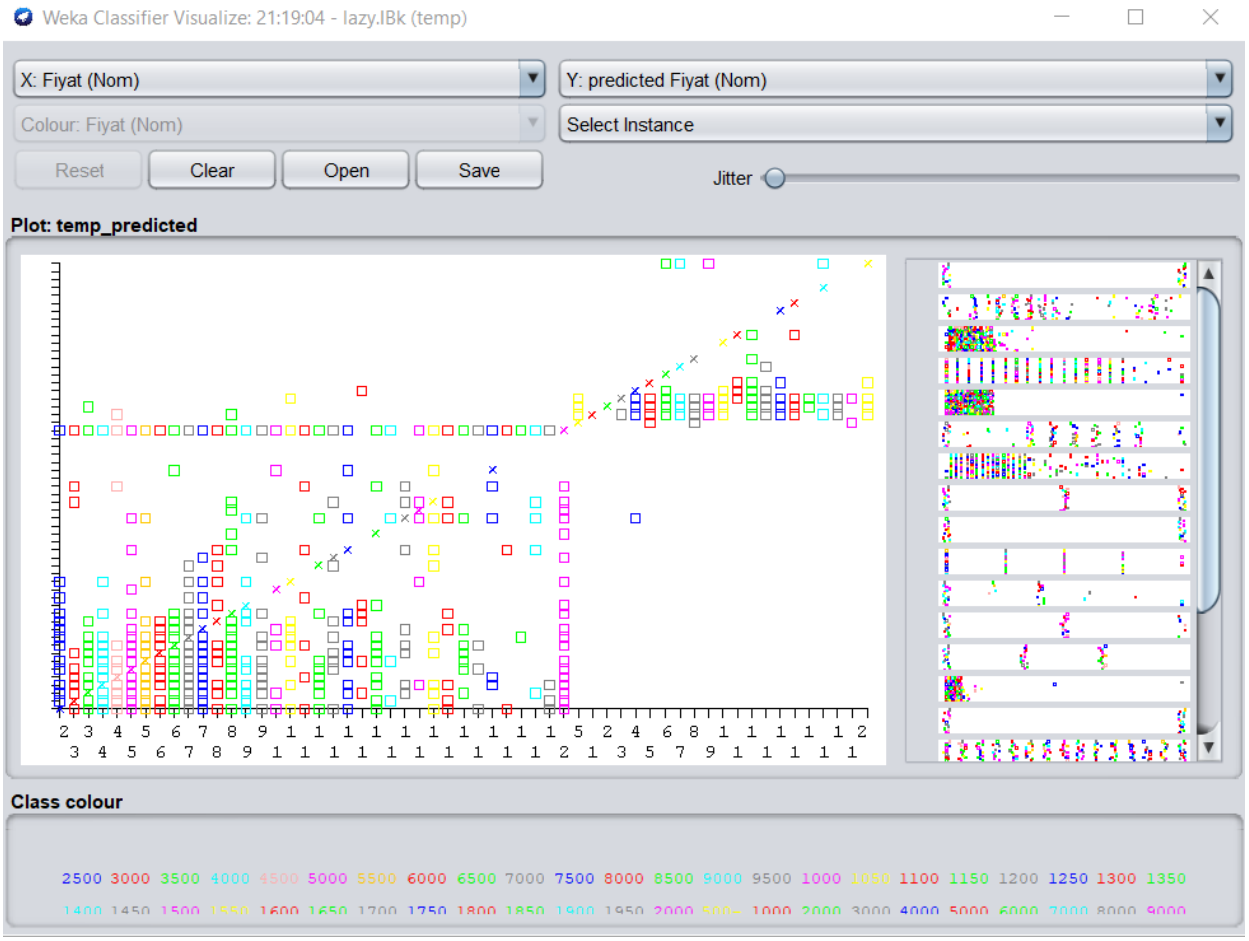

Figure 4.3.2.5 Post-Classification Estimation Plot of the Data Set

For example, when a point on the diagonal is selected, it is seen that the price is correctly classified.

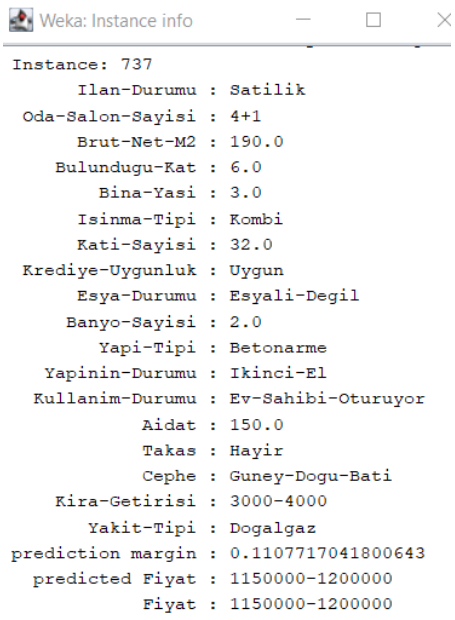

Figure 4.3.2.5 Correctly Classified Data

For example, when a point other than the diagonal is selected, it has been observed that the price is incorrectly classified.

Figure 4.3.2.5 Misclassified Data

4.3.2.6 Weka in C#

When the K-NN k=5 algorithm is run in the form application made in the C# programming language, a screen like the one in the figure is shown. Input values are requested from the user and the price class is suggested with the K-NN algorithm.

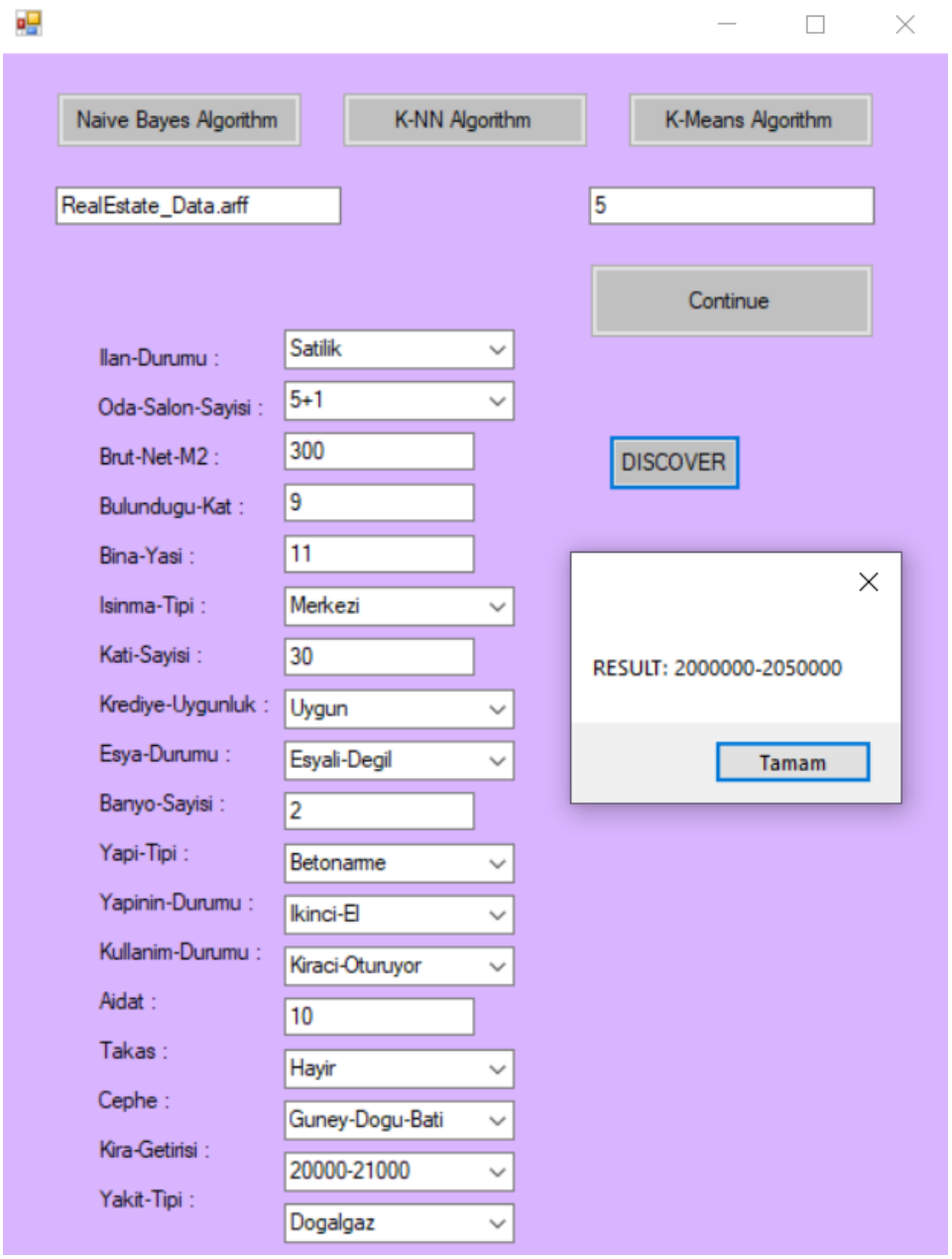

Figure 4.3.2.6 K-NN Classification and K=5 in C#

When the K-NN k=9 algorithm is run in the form application made in the C# programming language, a screen like the one in the figure is shown. Input values are requested from the user and the price class is suggested with the K-NN algorithm.

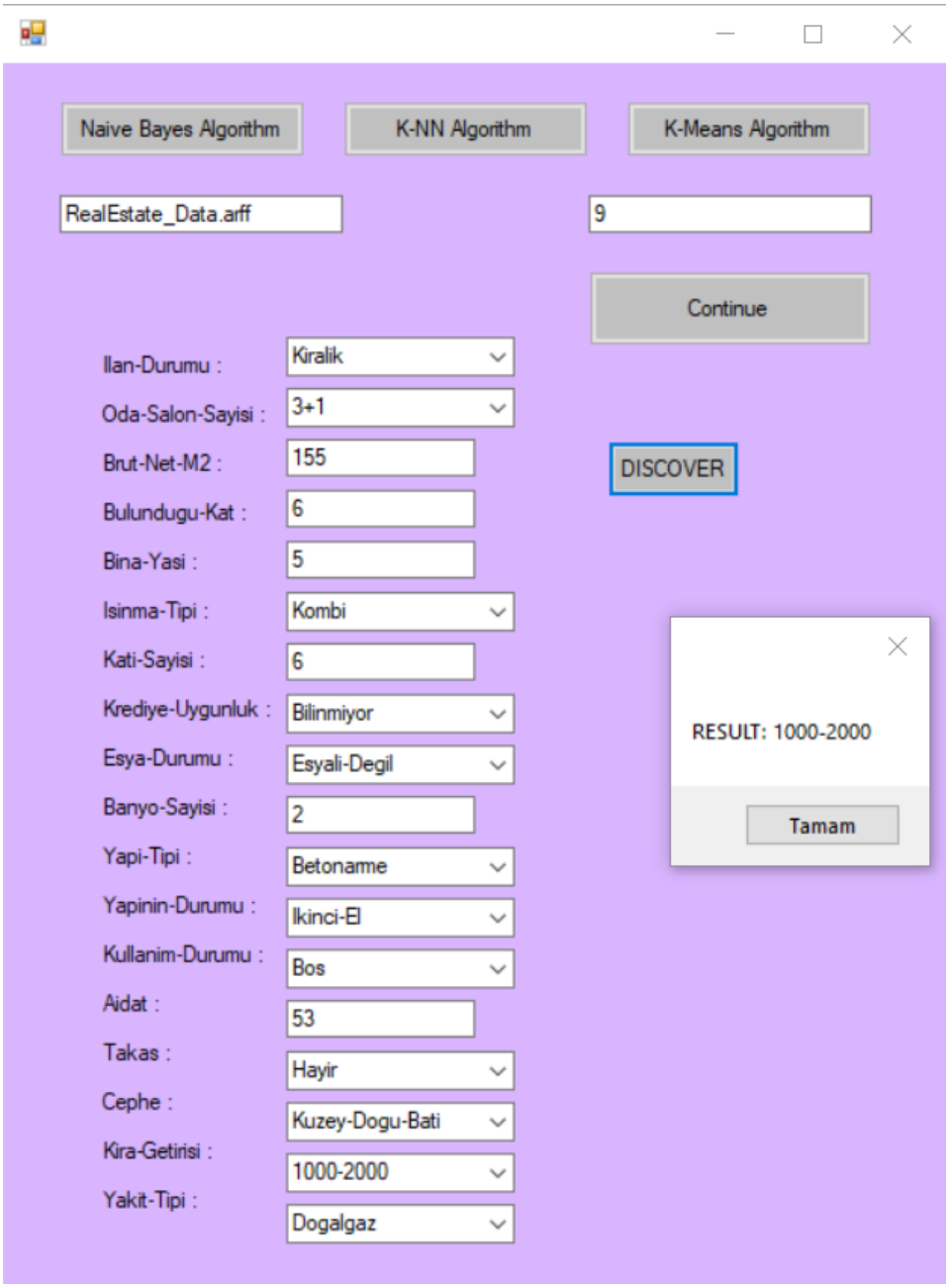

Figure 4.3.2.6 K-NN Classification and K=9 in C#

4.4 K-Means

4.4.1 When k is 5 and Euclidean Distance in Weka

Euclidean Distance was used when measuring distance. As in Figure 4.4.1, 5 data were randomly selected from the data set and these were determined as the centers of the clusters.

```
=== Clustering model (full training set) ===


kMeans
======

Number of iterations: 11
Within cluster sum of squared errors: 8490.158536860112

Initial starting points (random):

Cluster 0: Satilik,5+1,300,3,13,Kombi,3,Uygun,Esyali-Degil,2,Betonarme,Ikinci-El,Kiraci-Oturuyor,255,Evet,Guney,1000-2000,Dogalgaz,650000-700000
Cluster 1: Satilik,3+1,125,2,8,Kombi,3,Uygun,Esyali-Degil,1,Betonarme,Sifir,Bos,340,Evet,Dogu-Bati,1000-2000,Dogalgaz,550000-600000
Cluster 2: Satilik,3+1,135,12,15,Kombi,13,Uygun,Esyali-Degil,1,Betonarme,Sifir,Bos,939,Evet,Guney-Dogu-Bati,8000-9000,Dogalgaz,1050000-1100000
Cluster 3: Kiralik,2+1,95,2,3,Kombi,3,Bilinmiyor,Esyali-Degil,1,Betonarme,Ikinci-El,Bos,257,Hayir,Kuzey-Guney-Dogu-Bati,3000-4000,Dogalgaz,3000-4000
Cluster 4: Satilik,3+1,195,4,2,Kombi,4,Uygun,Esyali-Degil,1,Betonarme,Sifir,Bos,364,Evet,Dogu-Bati,6000-7000,Dogalgaz,1250000-1300000

Missing values globally replaced with mean/mode
```

Figure 4.4.1 Randomly Selected Centers

The centers of the clusters formed as a result of the K-Means clustering algorithm.

```
Final cluster centroids:
                                          Cluster#
Attribute                     Full Data           0           1           2           3           4
                              (2067.0)    (306.0)     (434.0)     (166.0)     (902.0)     (259.0)
=============================================================================================================================
Ilan-Durumu                    Satilik    Satilik     Satilik     Satilik     Kiralik     Satilik
Oda-Salon-Sayisi                   3+1        3+1         2+1         3+1         3+1         3+1
Brut-Net-M2                    140.3624   153.4542    116.5829    171.1325    135.5776    161.6834
Bulundugu-Kat                   4.6933     4.8758      3.6636      6.6687      5.0443      3.7143
Bina-Yasi                      18.3851    14.9837     16.7166     18.9819     19.3936     21.305
Isinma-Tipi                     Kombi      Kombi       Kombi       Kombi       Kombi       Kombi
Kati-Sayisi                     7.4194     7.1438      5.5184      9.7892      8.1918      6.722
Krediye-Uygunluk                Uygun      Uygun       Uygun       Uygun     Bilinmiyor    Uygun
Esya-Durumu                Esyali-Degil Esyali-Degil Esyali-Degil Esyali-Degil Esyali-Degil Esyali-Degil
Banyo-Sayisi                    1.4224     1.4804      1.1313      1.8193      1.3503      1.8378
Yapi-Tipi                     Betonarme  Betonarme   Betonarme   Betonarme   Betonarme   Betonarme
Yapinin-Durumu                Ikinci-El  Ikinci-El   Ikinci-El     Sifir     Ikinci-El     Sifir
Kullanim-Durumu                   Bos  Kiraci-Oturuyor    Bos        Bos         Bos         Bos
Aidat                          326.9119   299.7712    261.8871    409.7831    343.3404    357.61
Takas                           Hayir      Hayir       Hayir       Hayir       Hayir       Hayir
Cephe                  Kuzey-Guney-Dogu-Bati  Guney Kuzey-Guney-Dogu-Bati Guney-Dogu-Bati Guney-Dogu Kuzey-Guney-Dogu-Bati
Kira-Getirisi               2000-3000  2000-3000   1000-2000   2000-3000   2000-3000   6000-7000
Yakit-Tipi                    Dogalgaz   Dogalgaz    Dogalgaz    Dogalgaz    Dogalgaz    Dogalgaz
Fiyat                       2000-3000 2000000-2050000 250000-300000 2000000-2050000 2000-3000 2000000-2050000
```

Figure 4.4.1 Centers After Clustering Algorithm

The data numbers and rates in the clusters formed as a result of the algorithm are shown.

```
Time taken to build model (full training data) : 0.03 seconds

=== Model and evaluation on training set ===

Clustered Instances

0       306 ( 15%)
1       434 ( 21%)
2       166 (  8%)
3       902 ( 44%)
4       259 ( 13%)
```

Figure 4.4.1 Number of Elements in Sets

It is the graph showing which clusters the data in the data set belongs to.

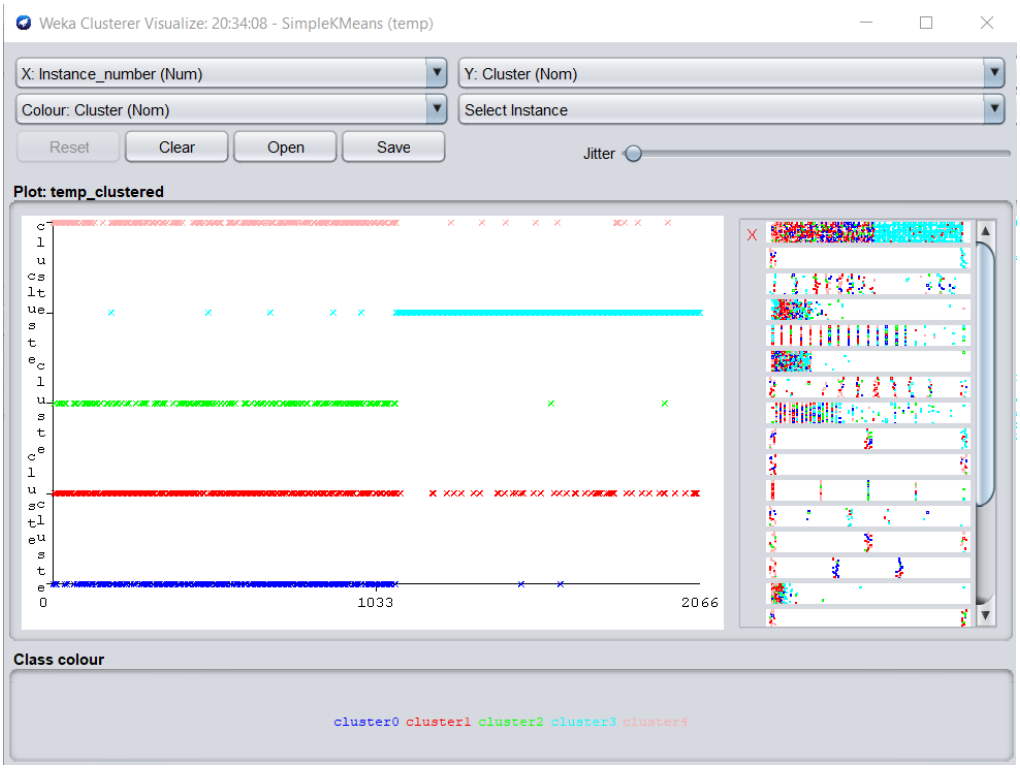

Figure 4.4.1 Post-Clustering Estimation Plot of the Dataset

It is the price distribution graph in the clusters.

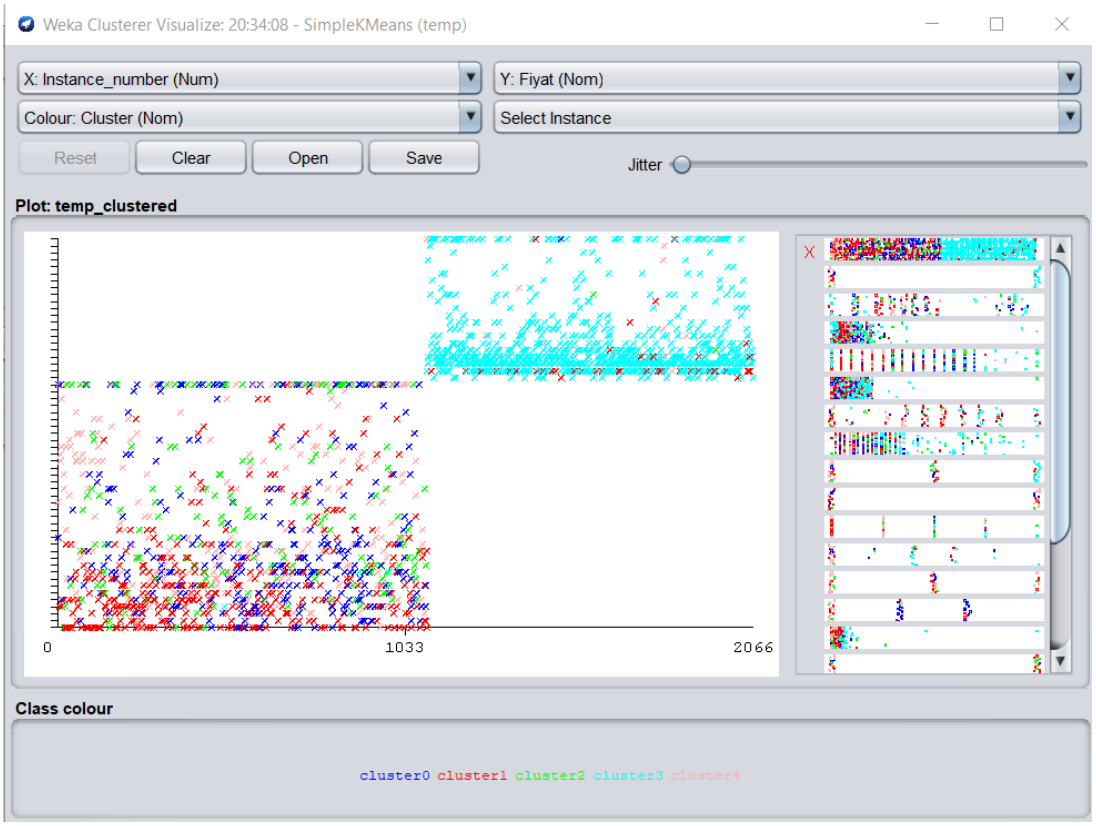

Figure 4.4.1 Price-Instance Graph

4.4.2 When k is 5 and Clustering in C#

When the K-Means k=5 clustering algorithm is run in the form application made in the C# programming language, a screen like the one in the figure is shown. The result of the clusters are printed to the file.

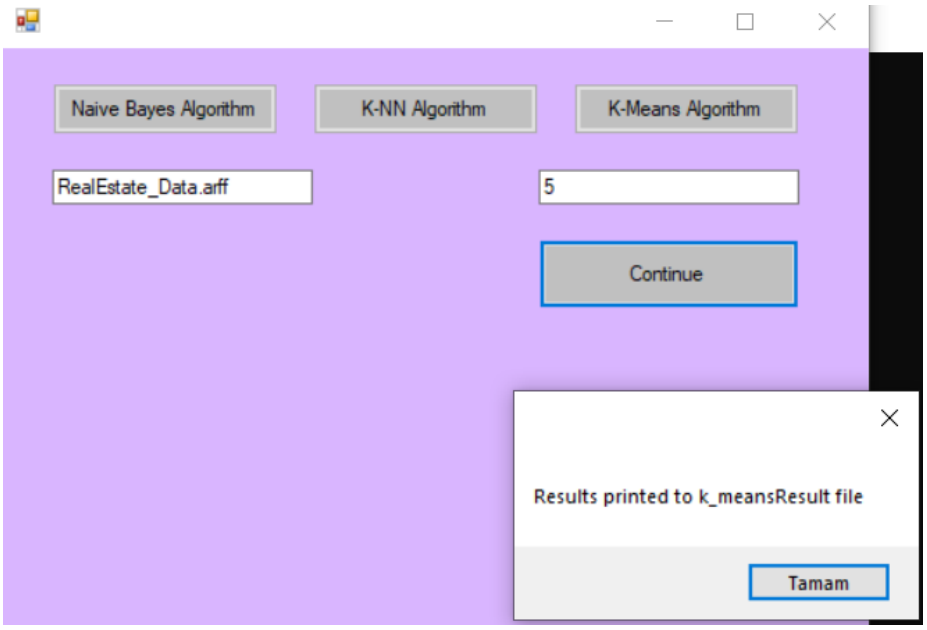

Figure 4.4.2 K-Means Clustering and K=5 in C#

The centers of the clusters formed and which data belong to which cluster are shown.

Figure 4.4.2 Cluster Centers and Cluster Elements

4.4.3 When k is 11 and Manhattan Distance in Weka

Manhattan Distance was used when measuring distance. As in Figure 4.4.3, 11 data were randomly selected from the data set and these were determined as the centers of the clusters.

```
kMeans
======

Number of iterations: 7
Sum of within cluster distances: 7873.858457688149

Initial starting points (random):

Cluster 0: Satilik,5+1,300,3,13,Kombi,3,Uygun,Esyali-Degil,2,Betonarme,Ikinci-El,Kiraci-Oturuyor,255,Evet,Guney,1000-2000,Dogalgaz,650000-700000
Cluster 1: Satilik,3+1,125,2,8,Kombi,3,Uygun,Esyali-Degil,1,Betonarme,Sifir,Bos,340,Evet,Dogu-Bati,1000-2000,Dogalgaz,550000-600000
Cluster 2: Satilik,3+1,135,12,15,Kombi,13,Uygun,Esyali-Degil,1,Betonarme,Sifir,Bos,939,Evet,Guney-Dogu-Bati,8000-9000,Dogalgaz,1050000-1100000
Cluster 3: Kiralik,2+1,95,2,3,Kombi,3,Bilinmiyor,Esyali-Degil,1,Betonarme,Ikinci-El,Bos,257,Hayir,Kuzey-Guney-Dogu-Bati,3000-4000,Dogalgaz,3000-4000
Cluster 4: Satilik,3+1,195,4,2,Kombi,4,Uygun,Esyali-Degil,1,Betonarme,Sifir,Bos,364,Evet,Dogu-Bati,6000-7000,Dogalgaz,1250000-1300000
Cluster 5: Kiralik,3+1,144,3,2,Merkezi,30,Bilinmiyor,Esyali-Degil,2,Betonarme,Ikinci-El,Bos,1000,Hayir,Kuzey-Bati,13000-14000,Dogalgaz,13000-14000
Cluster 6: Kiralik,2+1,130,2,11,Klima,3,Bilinmiyor,Esyali,1,Betonarme,Ikinci-El,Bos,875,Hayir,Guney-Bati,3000-4000,Elektrik,3000-4000
Cluster 7: Kiralik,3+1,250,1,18,Kombi,13,Bilinmiyor,Esyali-Degil,2,Betonarme,Ikinci-El,Bos,280,Hayir,Kuzey-Dogu-Bati,20000-21000,Dogalgaz,20000-21000
Cluster 8: Kiralik,2+1,90,5,8,Kombi,5,Bilinmiyor,Esyali,1,Betonarme,Ikinci-El,Bos,358,Hayir,Bati,3000-4000,Dogalgaz,3000-4000
Cluster 9: Satilik,1+1,75,2,5,Klima,4,Uygun,Esyali-Degil,1,Betonarme,Ikinci-El,Bos,50,Evet,Kuzey-Guney-Dogu-Bati,1000-2000,Komur-Odun,350000-400000
Cluster 10: Kiralik,3+1,130,3,13,Kombi,6,Bilinmiyor,Esyali-Degil,1,Betonarme,Ikinci-El,Bos,75,Hayir,Kuzey-Guney-Bati,2000-3000,Dogalgaz,2000-3000

Missing values globally replaced with mean/mode
```

Figure 4.4.3 Randomly Selected Centers

The centers of the clusters formed as a result of the K-Means clustering algorithm.

```
Final cluster centroids:
                                        Cluster#
Attribute                 Full Data           0           1           2           3           4
                            (2067.0)    (333.0)     (215.0)     (174.0)     (178.0)     (202.0)
=============================================================================================================================
Ilan-Durumu                 Satilik     Satilik     Satilik     Satilik     Kiralik     Satilik
Oda-Salon-Sayisi               3+1         3+1         3+1         3+1         2+1         3+1
Brut-Net-M2                    130         135         120         165         110         160
Bulundugu-Kat                    3           3           3           5           3           3
Bina-Yasi                       16          11          20          14          20          20
Isinma-Tipi                 Kombi       Kombi       Kombi       Kombi       Kombi       Kombi
Kati-Sayisi                      5           5           4           7           5           5
Krediye-Uygunluk            Uygun       Uygun       Uygun       Uygun       Bilinmiyor  Uygun
Esya-Durumu           Esyali-Degil  Esyali-Degil  Esyali-Degil  Esyali-Degil  Esyali-Degil  Esyali-Degil
Banyo-Sayisi                     1           1           1           2           1           2
Yapi-Tipi                Betonarme   Betonarme   Betonarme   Betonarme   Betonarme   Betonarme
Yapinin-Durumu              Ikinci-El   Ikinci-El   Sifir       Sifir       Ikinci-El   Sifir
Kullanim-Durumu                Bos  Kiraci-Oturuyor  Bos        Bos         Bos         Bos
Aidat                          190         155         150         230         148         267
Takas                        Hayir       Hayir       Hayir       Hayir       Hayir       Hayir
Cephe         Kuzey-Guney-Dogu-Bati       Guney  Guney-Dogu  Guney-Dogu-Bati  Kuzey-Guney-Dogu-Bati  Kuzey-Guney-Dogu-Bati
Kira-Getirisi             2000-3000   1000-2000   1000-2000   2000-3000   3000-4000   6000-7000
Yakit-Tipi                Dogalgaz    Dogalgaz    Dogalgaz    Dogalgaz    Dogalgaz    Dogalgaz
Fiyat                     2000-3000   2000000-2050000  250000-300000  2000000-2050000  3000-4000  2000000-2050000
```

Figure 4.4.3 Centers After Clustering Algorithm (1)

5	6	7	8	9	10
(149.0)	(108.0)	(116.0)	(126.0)	(149.0)	(317.0)
Kiralik	Kiralik	Kiralik	Kiralik	Satilik	Kiralik
3+1	2+1	3+1	2+1	2+1	3+1
150	92	170	97.5	110	130
8	3	4	4	2	3
18	10	20	15	15	20
Merkezi	Klima	Kombi	Kombi	Klima	Kombi
13	5	5.5	6	4	5
Bilinmiyor	Bilinmiyor	Bilinmiyor	Bilinmiyor	Uygun	Bilinmiyor
Esyali-Degil	Esyali	Esyali-Degil	Esyali	Esyali-Degil	Esyali-Degil
2	1	2	1	1	1
Betonarme	Betonarme	Betonarme	Betonarme	Betonarme	Betonarme
Ikinci-El	Ikinci-El	Ikinci-El	Ikinci-El	Ikinci-El	Ikinci-El
Bos	Bos	Bos	Bos	Bos	Bos
440	239.5	280	120.5	50	100
Hayir	Hayir	Hayir	Hayir	Evet	Hayir
Guney-Dogu	Kuzey-Guney-Dogu-Bati	Guney-Dogu-Bati	Guney-Dogu	Kuzey-Guney-Dogu-Bati	Guney-Bati
5000-6000	1000-2000	20000-21000	3000-4000	2000-3000	2000-3000
Dogalgaz	Elektrik	Dogalgaz	Dogalgaz	Komur-Odun	Dogalgaz
5000-6000	1000-2000	20000-21000	3000-4000	400000-450000	2000-3000

Figure 4.4.3 Centers After Clustering Algorithm (2)

The data numbers and rates in the clusters formed as a result of the algorithm are shown.

```
Time taken to build model (full training data) : 0.04 seconds

=== Model and evaluation on training set ===

Clustered Instances

 0       333 ( 16%)
 1       215 ( 10%)
 2       174 (  8%)
 3       178 (  9%)
 4       202 ( 10%)
 5       149 (  7%)
 6       108 (  5%)
 7       116 (  6%)
 8       126 (  6%)
 9       149 (  7%)
10       317 ( 15%)
```

Figure 4.4.3 Number of Elements in Sets

It is the graph showing which clusters the data in the data set belongs to.

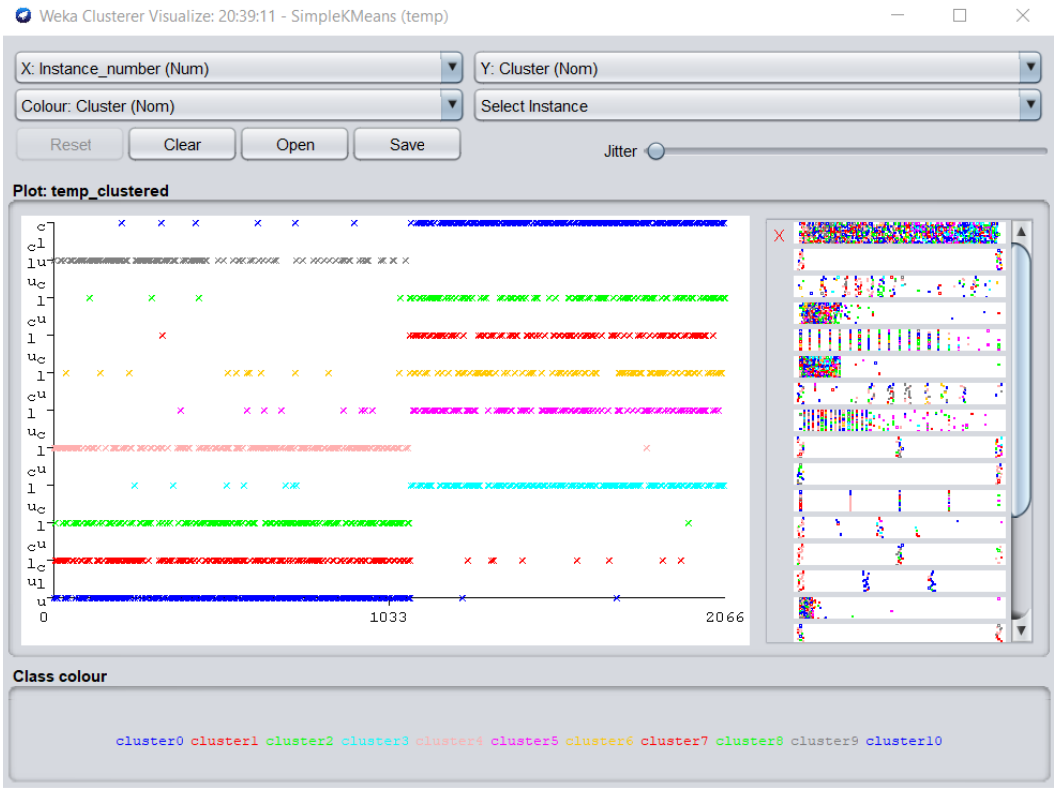

Figure 4.4.3 Post-Clustering Estimation Plot of the Dataset

It is the warming type distribution graph in the clusters.

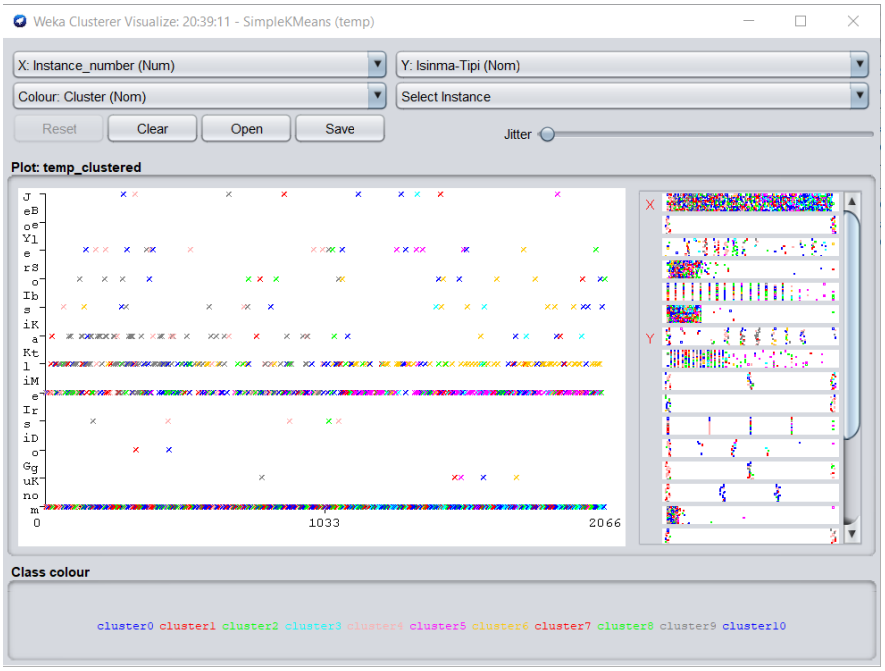

Figure 4.4.3 Warning Type-Instance Graph

4.4.4 When k is 11 and Clustering in C#

When the K-Means k=11 clustering algorithm is run in the form application made in the C# programming language, a screen like the one in the figure is shown. The result of the clusters are printed to the file.

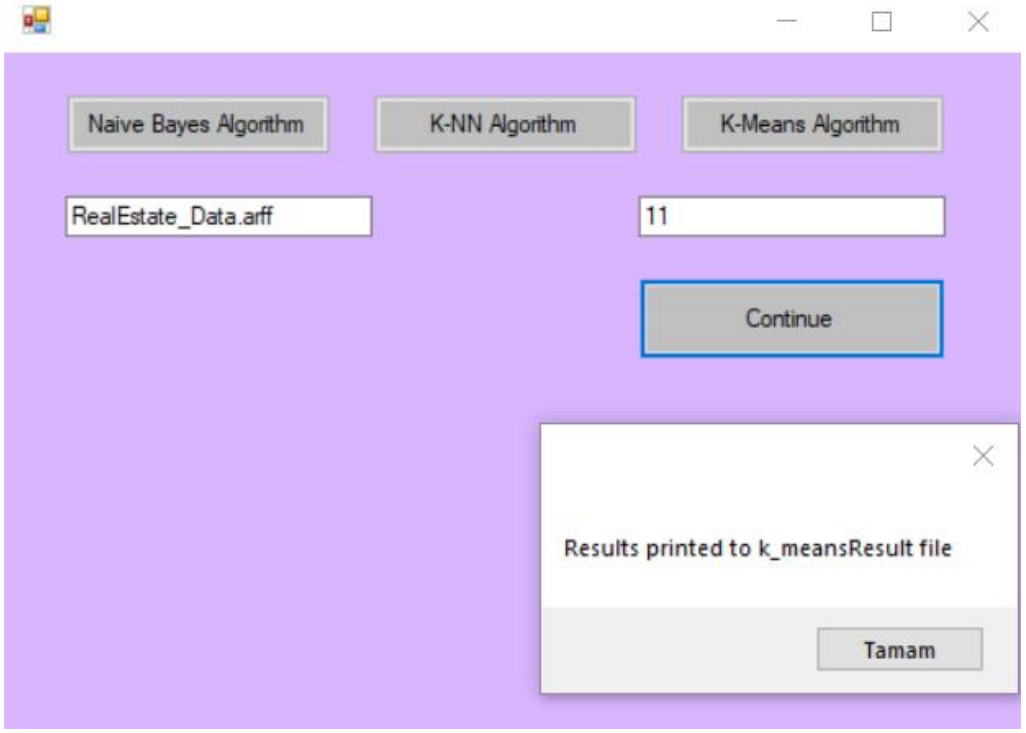

Figure 4.4.4 K-Means Clustering and k=11 in C#

The centers of the clusters formed and which data belong to which cluster are shown.

Figure 4.4.4 Cluster Centers and Cluster Elements

4.5 Linear Regression

Before performing linear regression, our data set was converted from nominal to binary.

4.5.1 Use Training Set

```
Linear Regression Model

Fiyat =

3405225.39   * Isinma-Tipi=Merkezi +
1899989.5536 * Krediye-Uygunluk=Uygun +
4411174.4238 * Kullanim-Durumu=Ev-Sahibi-Oturuyor +
9954081.4521 * Cephe=Dogu-Bati +
-1539129.7087

Time taken to build model: 0.63 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0.03 seconds

=== Summary ===

Correlation coefficient                  0.1263
Mean absolute error                2707887.1112
Root mean squared error           25186298.5016
Relative absolute error                163.4609 %
Root relative squared error             99.1988 %
Total Number of Instances             2063
```

Figure 4.5.1 Result When “Use Training Set” Option is Selected in Weka

4.5.2 Cross Validation

```
Linear Regression Model

Fiyat =

3405225.39   * Isinma-Tipi=Merkezi +
1899989.5536 * Krediye-Uygunluk=Uygun +
4411174.4238 * Kullanim-Durumu=Ev-Sahibi-Oturuyor +
9954081.4521 * Cephe=Dogu-Bati +
-1539129.7087

Time taken to build model: 0.4 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient                  0.0078
Mean absolute error                2818274.9277
Root mean squared error           25589724.8511
Relative absolute error                169.5992 %
Root relative squared error            100.7337 %
Total Number of Instances             2063
```

Figure 4.5.2 Result When "Cross Validation" is Selected and n=10 in Weka

4.5.3 %90 Training, %10 Test

```
Linear Regression Model

Fiyat =

3405225.39   * Isinma-Tipi=Merkezi +
1899989.5536 * Krediye-Uygunluk=Uygun +
4411174.4238 * Kullanim-Durumu=Ev-Sahibi-Oturuyor +
9954081.4521 * Cephe=Dogu-Bati +
-1539129.7087

Time taken to build model: 0.36 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0 seconds

=== Summary ===

Correlation coefficient                  0.2806
Mean absolute error                2148167.9896
Root mean squared error            3199254.6916
Relative absolute error               207.8031 %
Root relative squared error           267.9091 %
Total Number of Instances             206
```

Figure 4.5.3 Results With 10% of the Dataset

4.5.4 Supplied Test Set

4.5.4.1 Test 1

Test data: Satilik,3+1,125,2,10,Kombi,8,Uygun,Esyali-Degil,1,Betonarme,Ikinci El,Bos,135,Evet,Guney,5000,Dogalgaz,1270000

While our actual value was 1270000 when estimating the price in the linear regression process, the number 1239916.54 was found as an estimate, and the error rate was found to be -30083.46. This error rate is the difference between the estimated value and the true value.

```
=== Predictions on test set ===

    inst#     actual  predicted      error
        1 1270000 1239916.54 -30083.46

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.02 seconds

=== Summary ===

Correlation coefficient                  0
Mean absolute error                30083.4595
Root mean squared error            30083.4595
Relative absolute error              100      %
Root relative squared error          100      %
Total Number of Instances              1
```

Figure 4.5.4.1 Class Prediction in the Algorithm's Test Data

4.5.4.2 Test 2

Test data: Satilik, 3+1, 150, 10, 40,Kombi,11,Uygun,Esyali-Degil,2,Betonarme,Sifir,Bos, 274, Hayir, Dogu,6500,Dogalgaz,1650000

While our actual value was 1650000 when estimating the price in the linear regression process, the number 1239916.54 was found as an estimate, and the error rate was found to be -410083.46. This error rate is the difference between the estimated value and the true value.

```
=== Predictions on test set ===

    inst#     actual  predicted      error
        1 1650000 1239916.54 -410083.46

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.01 seconds

=== Summary ===

Correlation coefficient                  0
Mean absolute error                410083.4595
Root mean squared error            410083.4595
Relative absolute error               100      %
Root relative squared error           100      %
Total Number of Instances              1
```

Figure 4.5.4.2 Class Prediction in the Algorithm's Test Data

4.6 Multilayer Perceptron

4.6.1 Use Training Set

```
Time taken to build model: 98.07 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0.09 seconds

=== Summary ===

Correlation coefficient                  0.997
Mean absolute error                 743718.8898
Root mean squared error            2033473.9017
Relative absolute error                44.8944 %
Root relative squared error             8.009  %
Total Number of Instances            2063
```

Figure 4.6.1 Result When “Use Training Set” Option is Selected in Weka

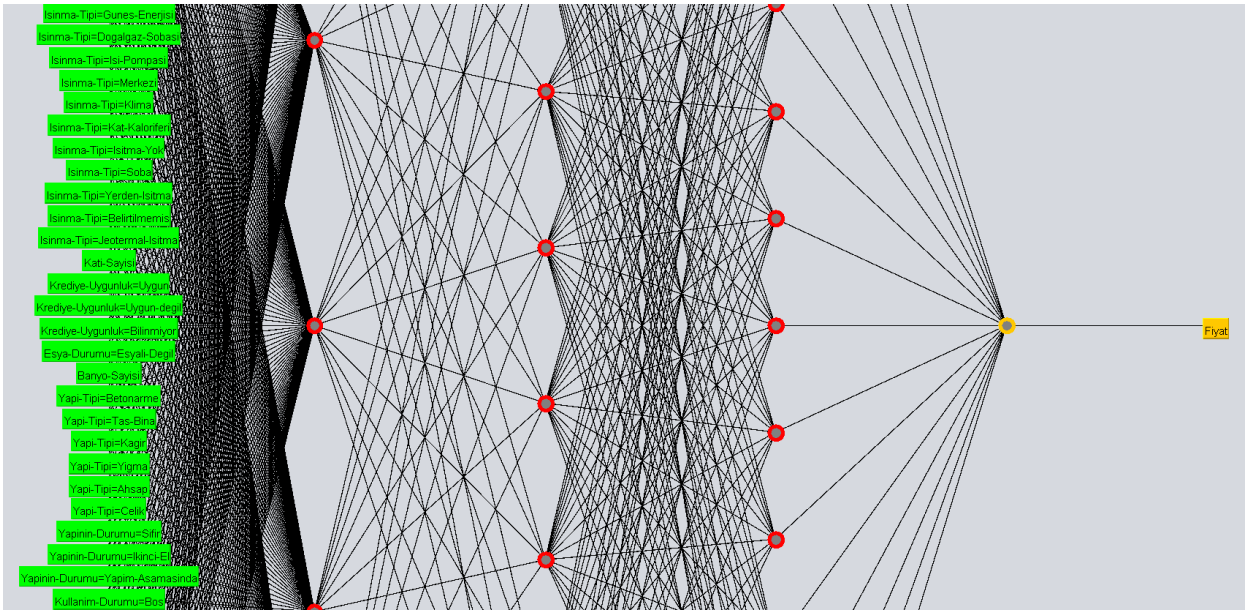

Figure 4.6.1 Neural Network in Weka

4.6.2 Cross Validation

```
=== Cross-validation ===
=== Summary ===

Correlation coefficient                  0.0008
Mean absolute error                1924017.6324
Root mean squared error           33101559.4033
Relative absolute error                115.7842 %
Root relative squared error            130.304  %
Total Number of Instances             2063
```

Figure 4.6.2 Result When "Cross Validation" is Selected and n=10 in Weka

4.6.3 %90 Train, %10 Test

```
=== Evaluation on test split ===

Time taken to test model on test split: 0 seconds

=== Summary ===

Correlation coefficient                  0.4613
Mean absolute error                 660524.9381
Root mean squared error            1091370.8956
Relative absolute error                 63.8959 %
Root relative squared error             91.3926 %
Total Number of Instances              206
```

Figure 4.6.3 Results With 10% of the Dataset

4.6.4 Supplied Test Set

4.6.4.1 Test1

Test Data: Satilik,4+1,250,4,15,Kombi,4,Uygun,Esyali-Degil,2,Betonarme,Ikinci-El,Bos,75,Hayir,Guney-Dogu,4500,Dogalgaz,969000

While our actual value was 969000 when estimating the price in the multilayer perceptron process, the number 1853268.342 was found as an estimate, and the error rate was found to be 884268.342. This error rate is the difference between the estimated value and the true value.

```
=== Predictions on test set ===

    inst#     actual  predicted      error
        1 969000     1853268.342 884268.342

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0 seconds

=== Summary ===

Correlation coefficient                  0
Mean absolute error                884268.3415
Root mean squared error            884268.3415
Relative absolute error               326.3988 %
Root relative squared error           326.3988 %
Total Number of Instances              1
```

Figure 4.6.4.1 Class Prediction in the Algorithm's Test Data

4.6.4.2 Test 2

Test Data: Satilik,4+1,250,2,20,Kombi,3,Bilinmiyor,Esyali,1,Betonarme,Ikinci-El,Bos,35,Hayir,Guney-Bati,15000,Dogalgaz,128000250

While our actual value was 128000250 when estimating the price in the multilayer perceptron process, the number 1292964579.111 was found as an estimate, and the error rate was found to be 1164964329.111. This error rate is the difference between the estimated value and the true value.

```
=== Predictions on test set ===

    inst#     actual  predicted      error
        1 128000250 1292964579.111 1164964329.111

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.01 seconds

=== Summary ===

Correlation coefficient                  0
Mean absolute error           1164964329.1108
Root mean squared error       1164964329.1108
Relative absolute error             919.0291 %
Root relative squared error         919.0291 %
Total Number of Instances              1
```

Figure 4.6.4.2 Class Prediction in the Algorithm's Test Data

4.7 Non Linear Regression (M5P)

4.7.1 Use Training Set

```
Bina-Yasi <= 29.5 :
|   Ilan-Durumu=Satilik <= 0.5 : LM1 (722/0.128%)
|   Ilan-Durumu=Satilik >  0.5 :
|   |   Banyo-Sayisi <= 1.5 : LM2 (496/2.343%)
|   |   Banyo-Sayisi >  1.5 :
|   |   |   Brut-Net-M2 <= 211.5 : LM3 (291/4.876%)
|   |   |   Brut-Net-M2 >  211.5 :
|   |   |   |   Isinma-Tipi=Dogalgaz-Sobasi,Isi-Pompasi,Belirtilmemis,Merkezi <= 0.5 : LM4 (67/8.945%)
|   |   |   |   Isinma-Tipi=Dogalgaz-Sobasi,Isi-Pompasi,Belirtilmemis,Merkezi >  0.5 : LM5 (24/25.147%)
Bina-Yasi >  29.5 :
|   Brut-Net-M2 <= 142.5 : LM6 (325/2.682%)
|   Brut-Net-M2 >  142.5 :
|   |   Cephe=Kuzey-Guney-Bati,Guney-Dogu-Bati,Dogu-Bati <= 0.5 :
|   |   |   Ilan-Durumu=Satilik <= 0.5 : LM7 (39/0.053%)
|   |   |   Ilan-Durumu=Satilik >  0.5 :
|   |   |   |   Oda-Salon-Sayisi=1+4,3+1,4+2,8+1,5+2,5+1,6+2 <= 0.5 : LM8 (24/2.399%)
|   |   |   |   Oda-Salon-Sayisi=1+4,3+1,4+2,8+1,5+2,5+1,6+2 >  0.5 :
|   |   |   |   |   Aidat <= 158.5 : LM9 (10/1.005%)
|   |   |   |   |   Aidat >  158.5 :
|   |   |   |   |   |   Aidat <= 471 : LM10 (15/6.059%)
|   |   |   |   |   |   Aidat >  471 :
|   |   |   |   |   |   |   Kira-Getirisi <= 13388.5 : LM11 (12/3.288%)
|   |   |   |   |   |   |   Kira-Getirisi >  13388.5 : LM12 (4/9.252%)
|   |   Cephe=Kuzey-Guney-Bati,Guney-Dogu-Bati,Dogu-Bati >  0.5 :
|   |   |   Kati-Sayisi <= 3.5 :
|   |   |   |   Ilan-Durumu=Satilik <= 0.5 : LM13 (5/0.031%)
|   |   |   |   Ilan-Durumu=Satilik >  0.5 : LM14 (4/984.472%)
|   |   |   Kati-Sayisi >  3.5 :
|   |   |   |   Bulundugu-Kat <= 2.5 : LM15 (7/24.487%)
|   |   |   |   Bulundugu-Kat >  2.5 : LM16 (18/2.695%)
```

Figure 4.7.1 Result Equations When “Use Training Set” Option is Selected in Weka

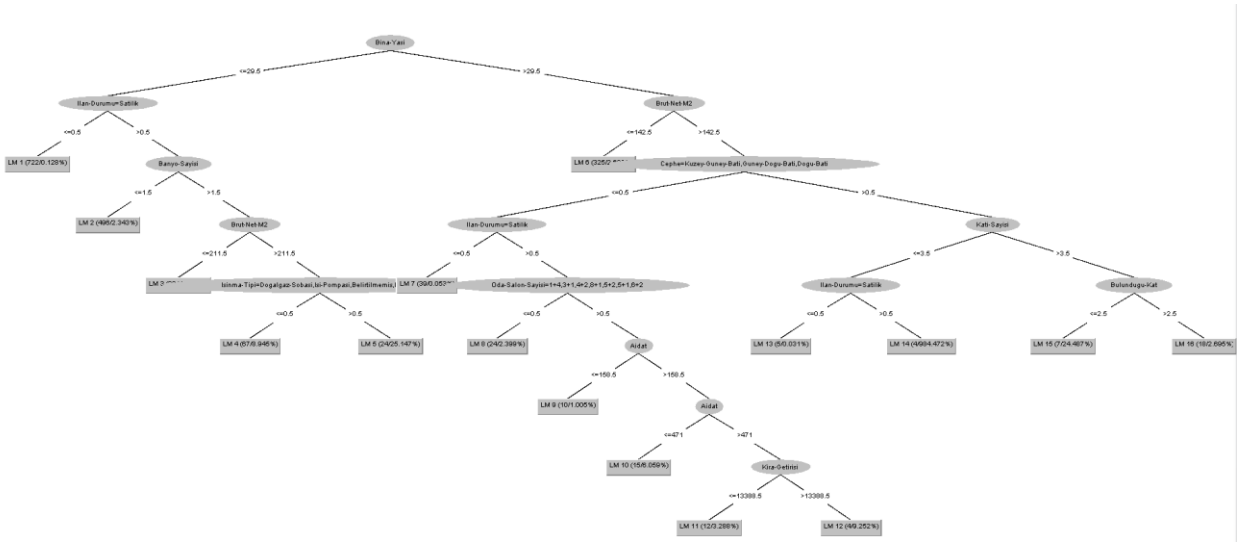

Figure 4.7.1 Result Tree in Weka

```
Number of Rules : 16

Time taken to build model: 0.17 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0 seconds

=== Summary ===

Correlation coefficient                  0.6252
Mean absolute error                1326882.9256
Root mean squared error           22446474.3381
Relative absolute error                80.0969 %
Root relative squared error            88.4078 %
Total Number of Instances           2063
```

Figure 4.7.1 Result When “Use Training Set” Option is Selected in Weka

4.7.2 Cross Validation

```
=== Cross-validation ===
=== Summary ===

Correlation coefficient                  0.0098
Mean absolute error                2313789.0547
Root mean squared error           25681350.0946
Relative absolute error               139.24   %
Root relative squared error           101.0944 %
Total Number of Instances           2063
```

Figure 4.7.2 Result When "Cross Validation" is Selected and n=10 in Weka

4.7.3 %90 Train, %10 Test

```
=== Evaluation on test split ===

Time taken to test model on test split: 0 seconds

=== Summary ===

Correlation coefficient                  0.289
Mean absolute error                1837374.1298
Root mean squared error           2456394.1918
Relative absolute error               177.7384 %
Root relative squared error           205.7012 %
Total Number of Instances           206
```

Figure 4.7.3 Results with 10% of the Dataset

4.7.4 Supplied Test Set

4.7.4.1 Test 1

Test Data:Satilik,3+1,130,1,22,Kombi,2,Uygun,Esyali Degil,2,Betonarme,Sifir,Bos,50, Evet,Guney-Dogu,2500,Dogalgaz,585000

While our actual value was 585000 when estimating the price in the multilayer perceptron process, the number 537152.981 was found as an estimate, and the error rate was found to be -47847.019. This error rate is the difference between the estimated value and the true value.

```
=== Predictions on test set ===

    inst#     actual  predicted      error
        1 585000     537152.981 -47847.019

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0 seconds

=== Summary ===

Correlation coefficient                  0
Mean absolute error                47847.0185
Root mean squared error            47847.0185
Relative absolute error                7.3058 %
Root relative squared error            7.3058 %
Total Number of Instances              1
```

Figure 4.7.4.1 Class Prediction in the Algorithm's Test Data

4.7.4.2 Test 2

Test Data: Satilik,3+1,150,8,5,Kombi,9,Uygun,Esyali-Degil,1,Betonarme,Ikinci-El,Bos,50,Evet,Kuzey-Guney-Bati,6932,Dogalgaz,550000

While our actual value was 550000 when estimating the price in the multilayer perceptron process, the number 680892.991 was found as an estimate, and the error rate was found to be 130892.991. This error rate is the difference between the estimated value and the true value.

```
=== Predictions on test set ===

    inst#     actual  predicted      error
        1 550000     680892.991 130892.991

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0 seconds

=== Summary ===

Correlation coefficient                  0
Mean absolute error                130892.9906
Root mean squared error            130892.9906
Relative absolute error                18.9723 %
Root relative squared error            18.9723 %
Total Number of Instances                1
```

Figure 4.7.4.2 Class Prediction in the Algorithm's Test Data

5. References

https://www.hepsiemlak.com/satilik/daire

https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/

https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning

https://www.javatpoint.com/k-means-clustering-algorithm-in-machine-learning

https://www.researchgate.net/publication/345970825_Precision-Recall_Curve_PRC_Classification_Trees

https://datacadamia.com/data_mining/validation_set

https://machinelearningmastery.com/how-to-tune-a-machine-learning-algorithm-in-weka/

https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc

https://builtin.com/data-science/precision-and-recall

https://en.wikipedia.org/wiki/Precision_and_recall

https://www.pico.net/kb/what-is-a-false-positive-rate/,

http://www.stat.yale.edu/Courses/1997-98/101/linreg.htm

https://www.techopedia.com/definition/20879/multilayer-perceptron-mlp

https://www.sciencedirect.com/topics/mathematics/linear-regression

https://www.sciencedirect.com/topics/engineering/model-tree