# Classification: DecisionTable, LWL, RandomTree and RandomForest

* 1. DecisionTable

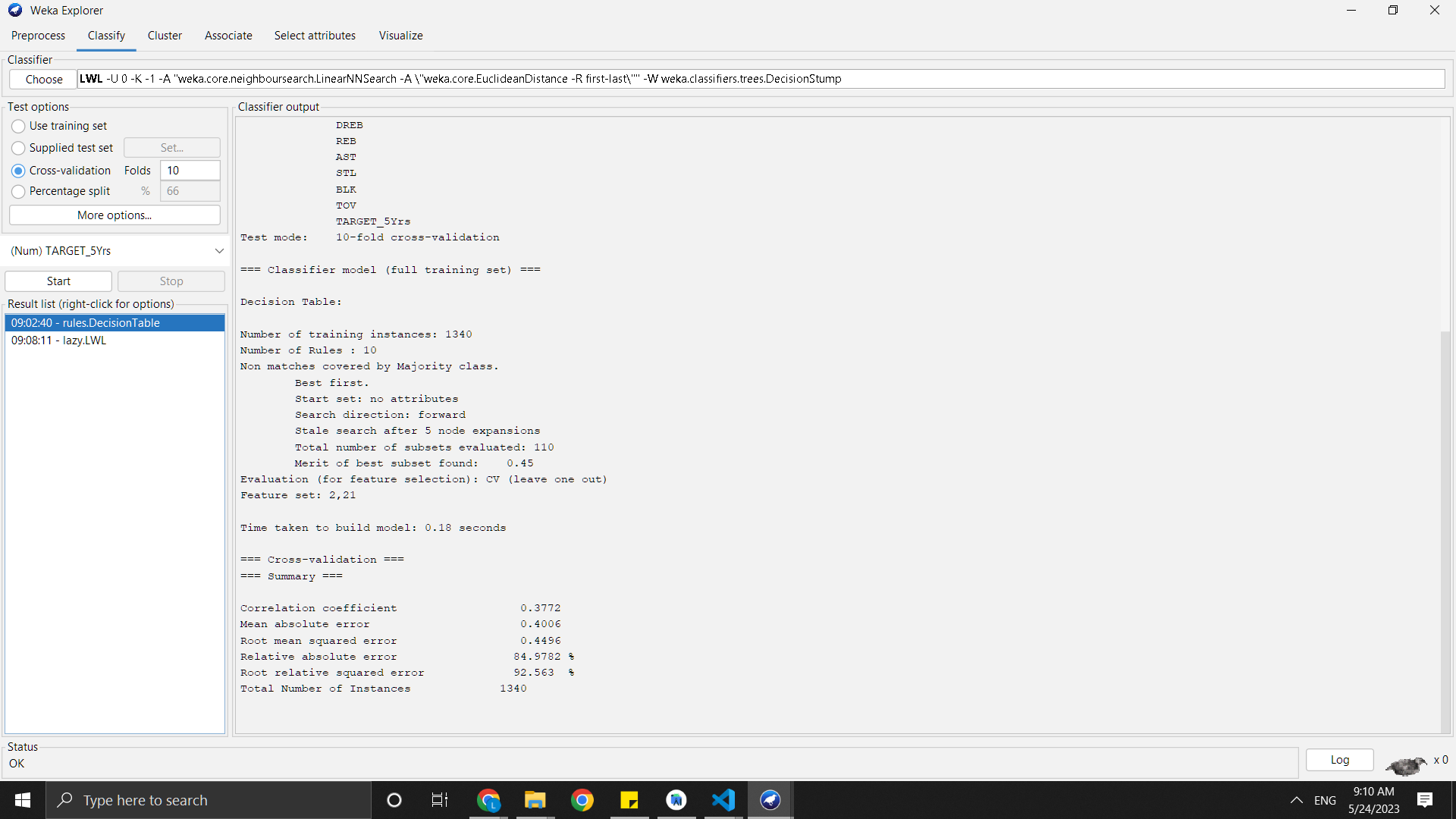


Fig ‎1.1: DecisionTable on Raw Data

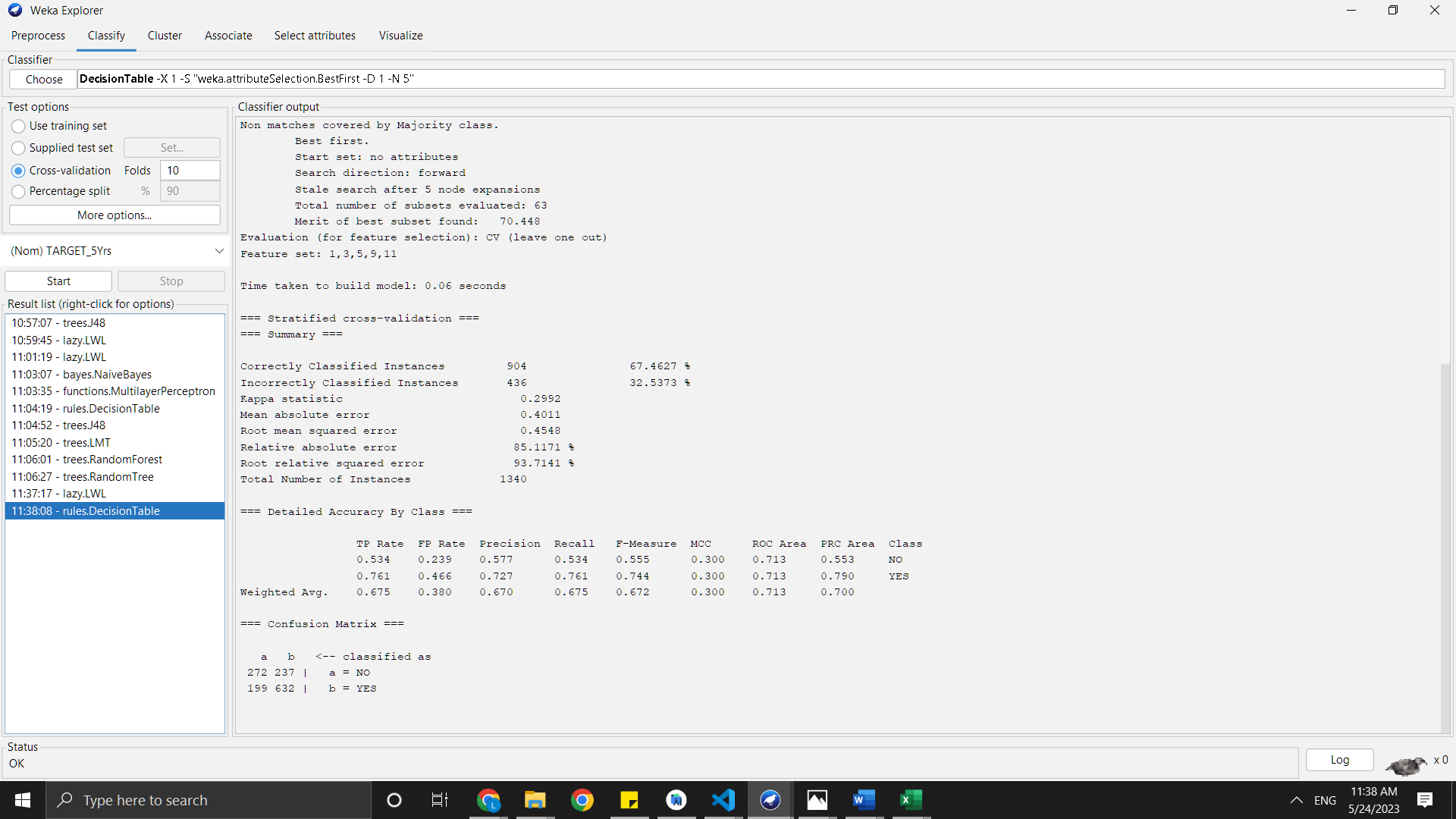


Fig ‎1.2: DecisionTable After Preprocessing

Implemented the DecisionTable on the raw data with Relative absolute error around 85% and after the preprocessing the Relative absolute error went down to around 81% “Accuracy 66”.

* 1. LWL

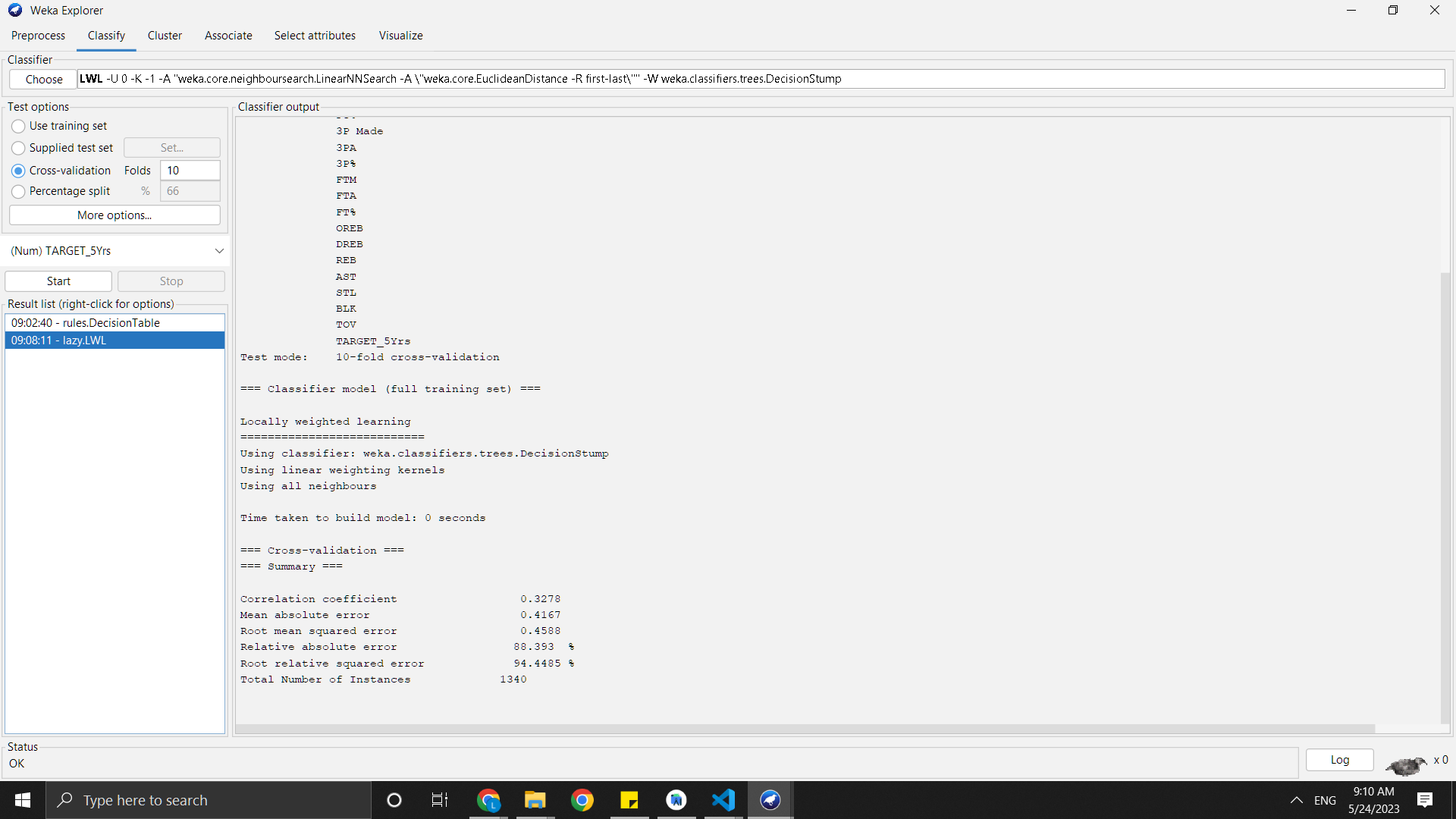
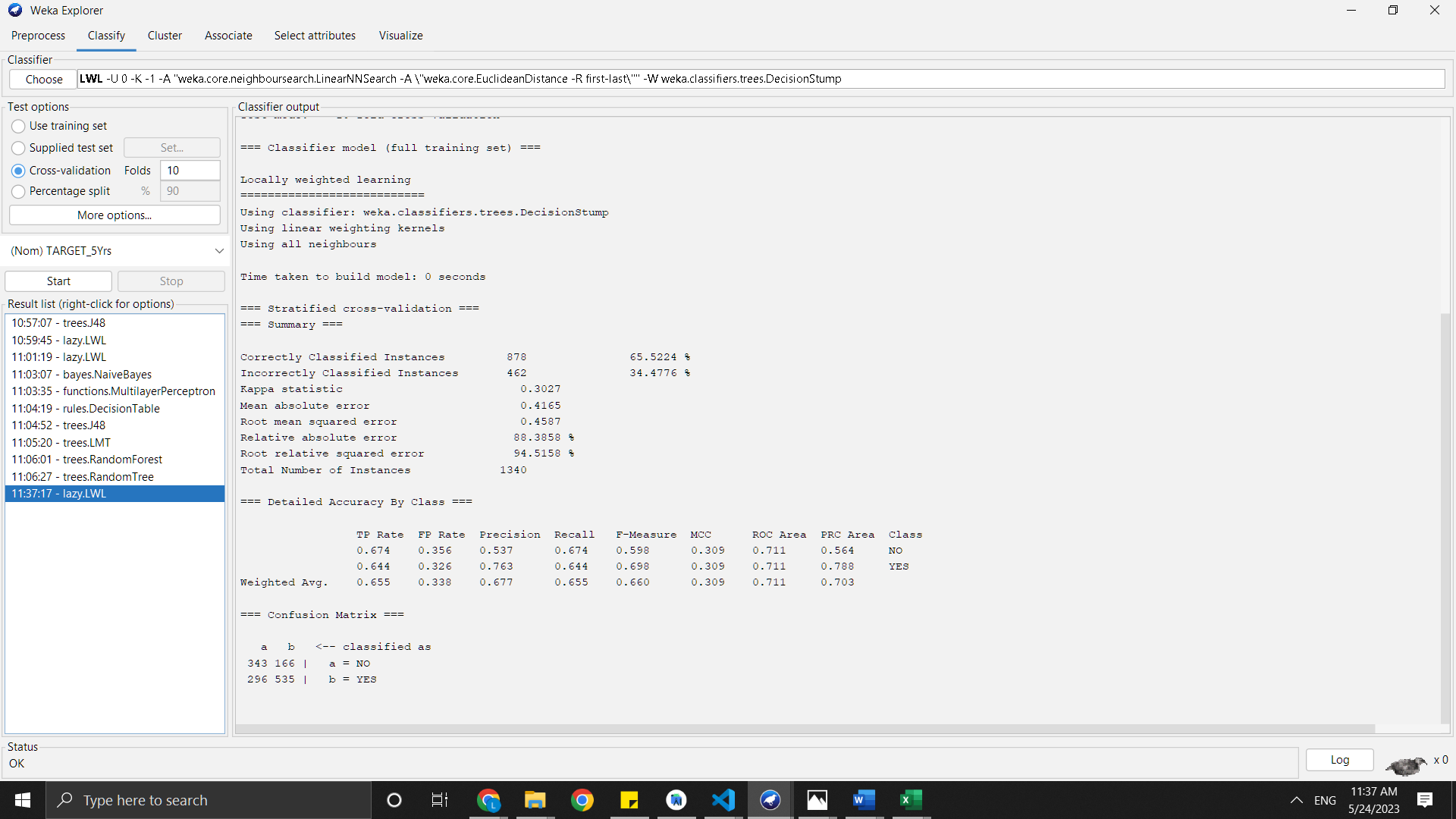
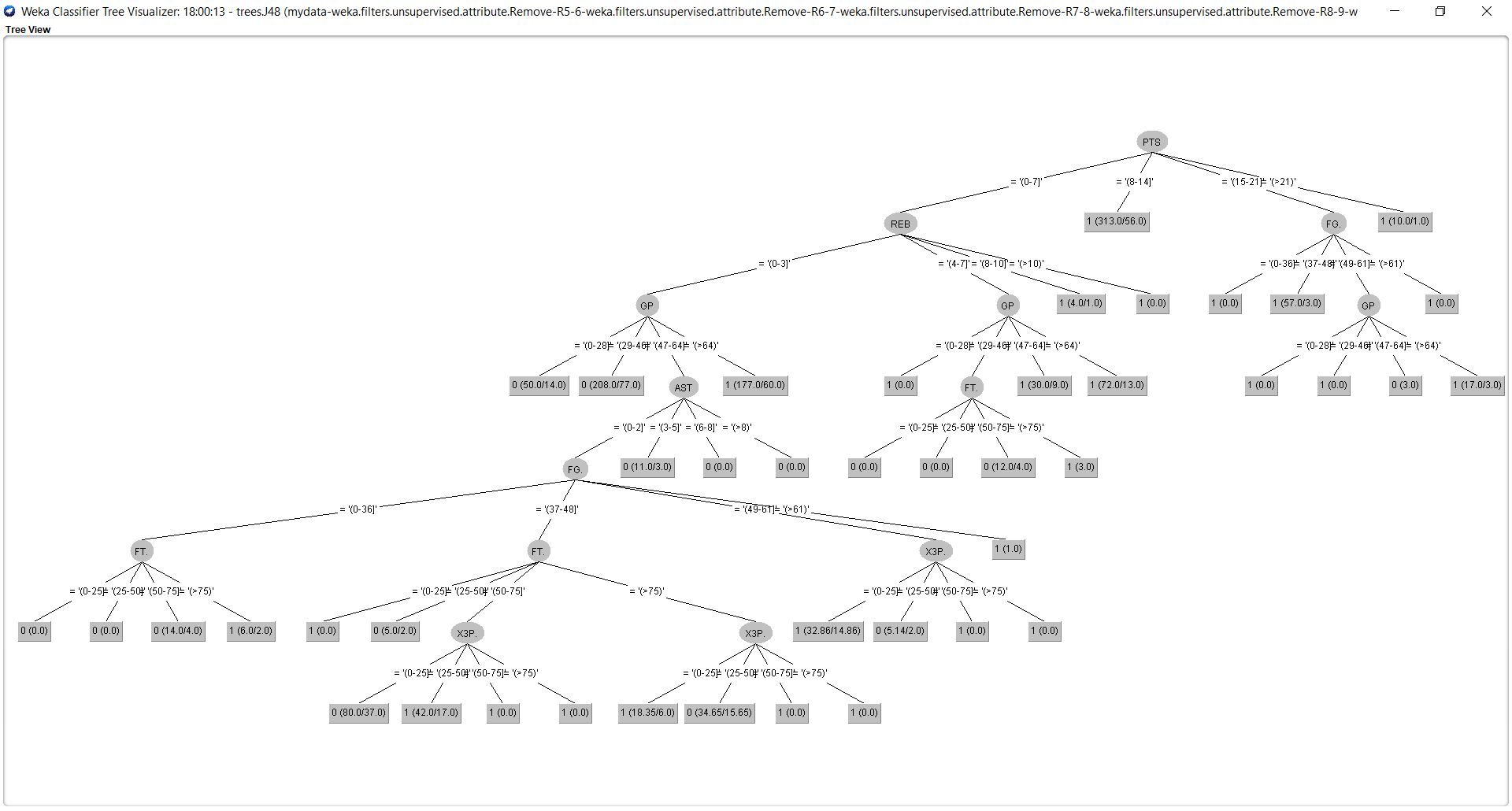


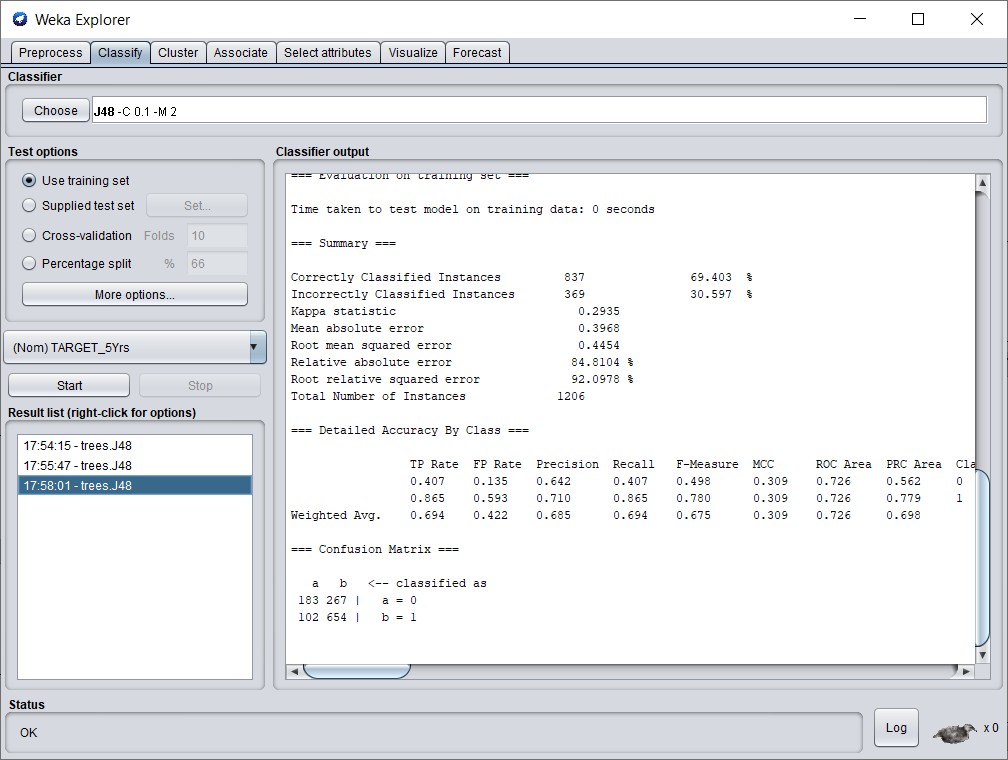
Fig ‎1.3: LWL on Raw data



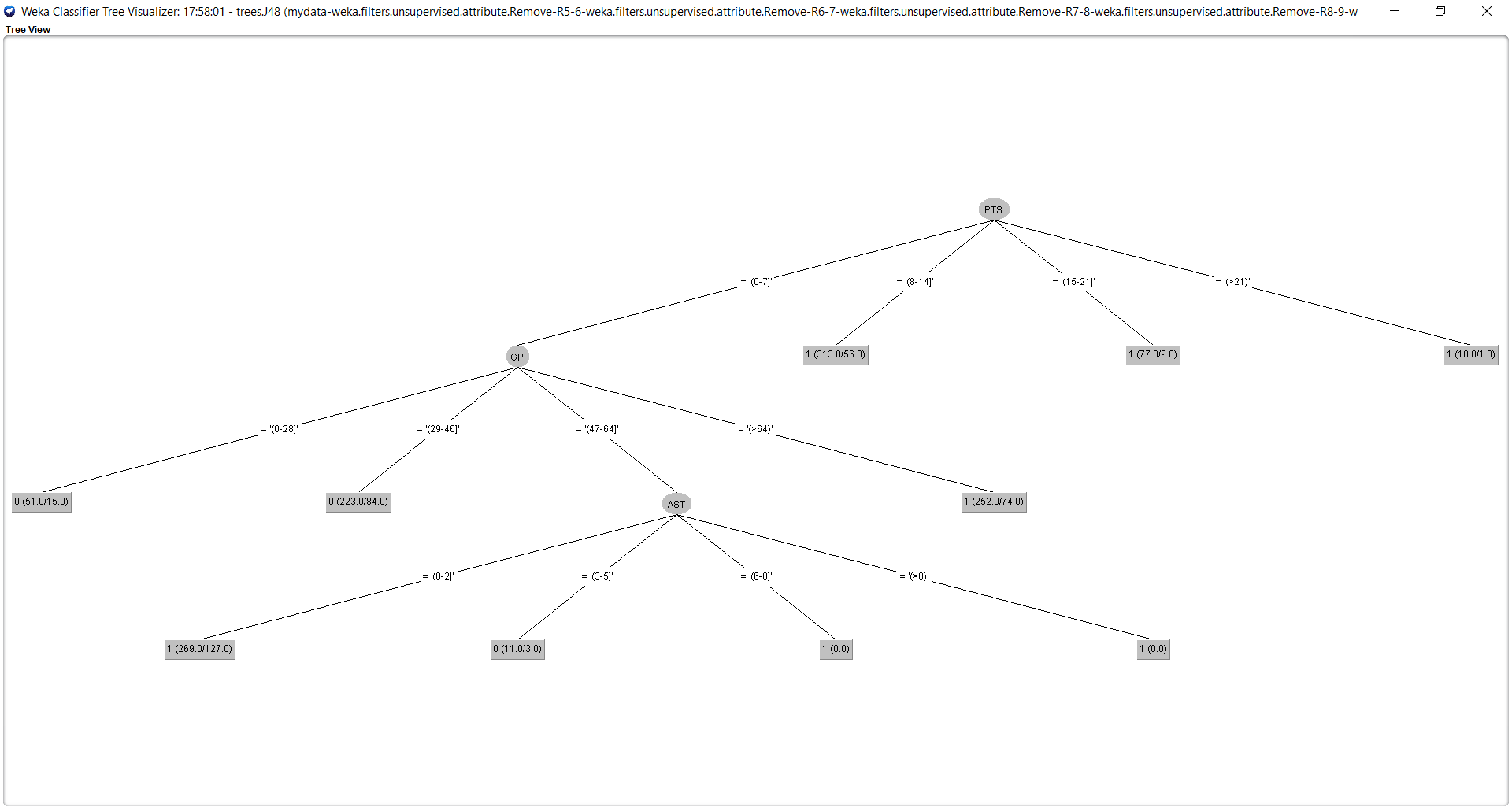


This is the visualisation of the J48 tree for Test 2 . The tree has too many branches and even though detailed is tougher to grasp. It seems even more complicated than the previous test. We run the next test by decreasing the confidence value.

1. Test 3



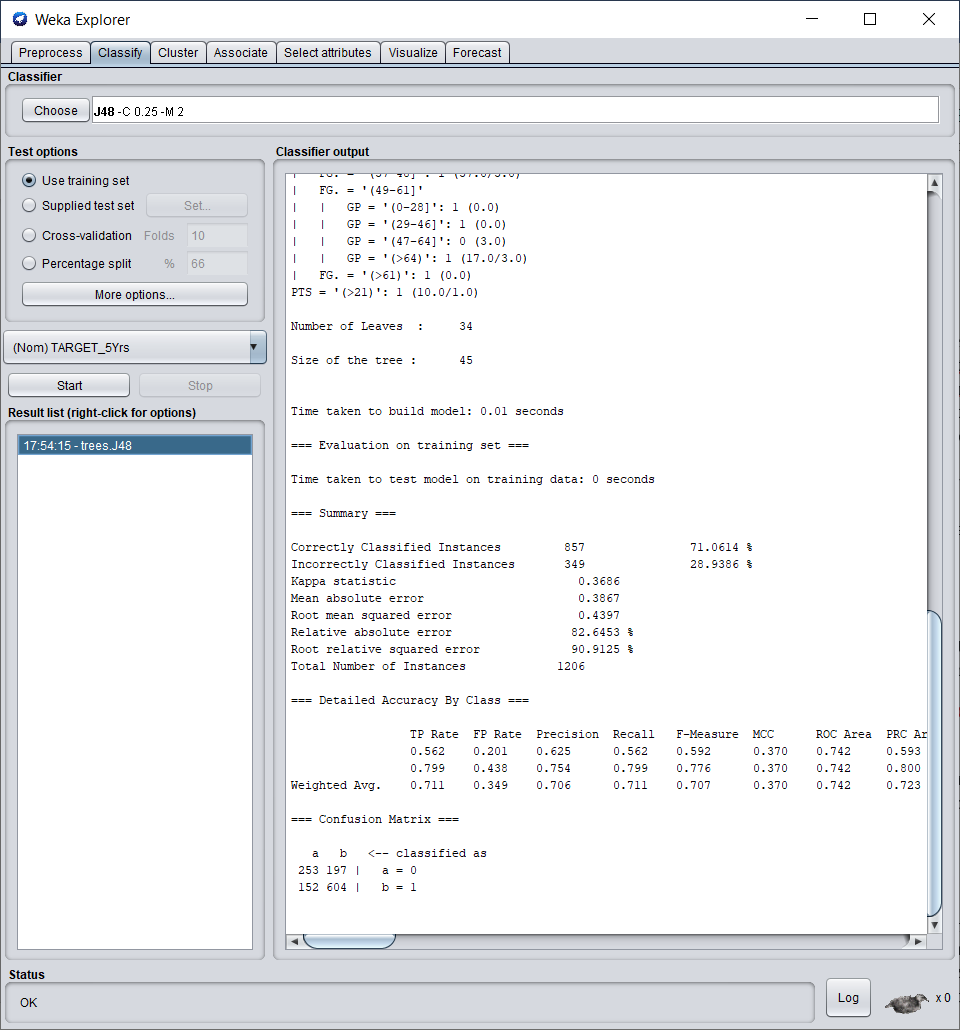
We run the J48 algorithm on the training dataset with a confidence of 0.1 and a minimum no of objects as 2. We get around 69.4% correctly classified instances. The instances are classified slightly worse than the previous two tests.

–

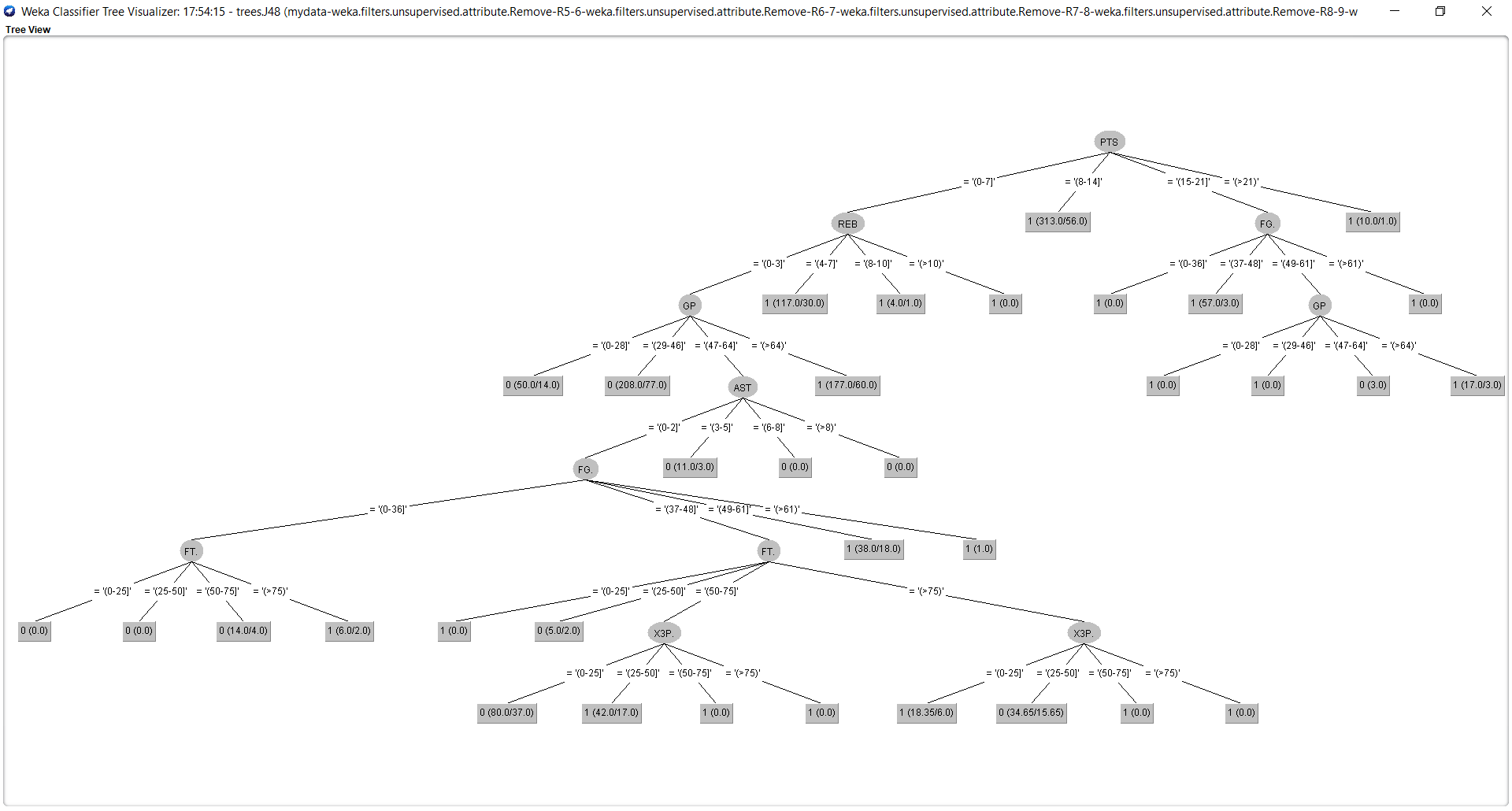
This is the visualisation of the J48 tree for Test 3. The tree seems easy to understand and looks better than the previous 2 tests.

Testing model:

1. Test 4

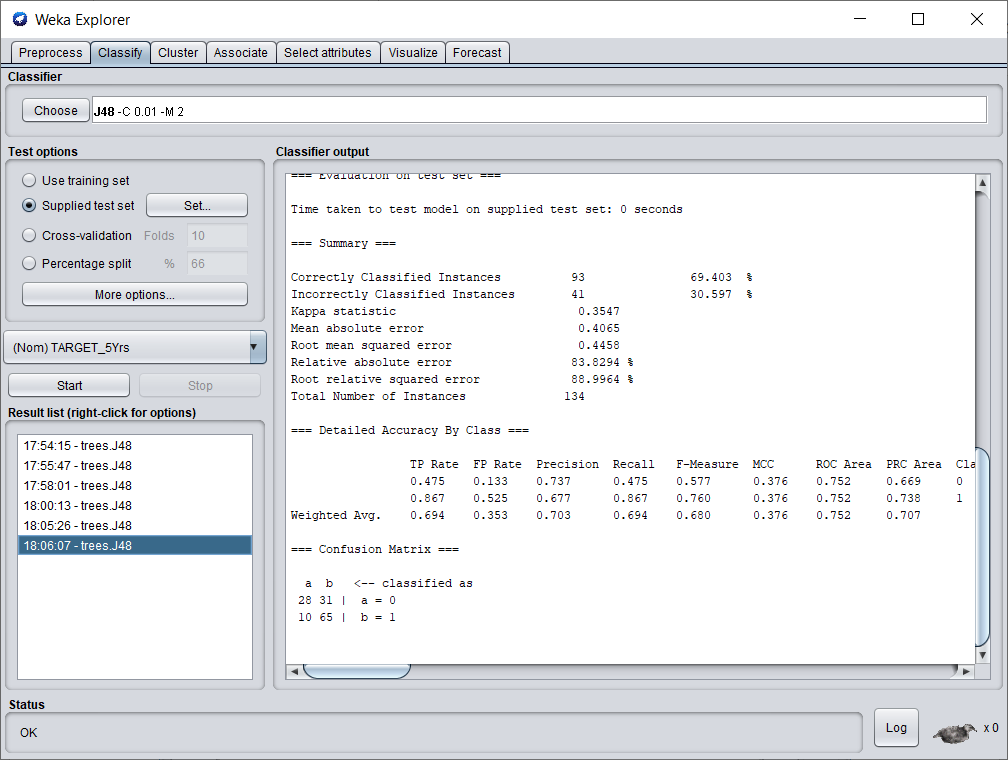


We run the J48 algorithm on the testing dataset with a confidence of 0.25 and a minimum no of objects as 2. We get around 71% correctly classified instances.

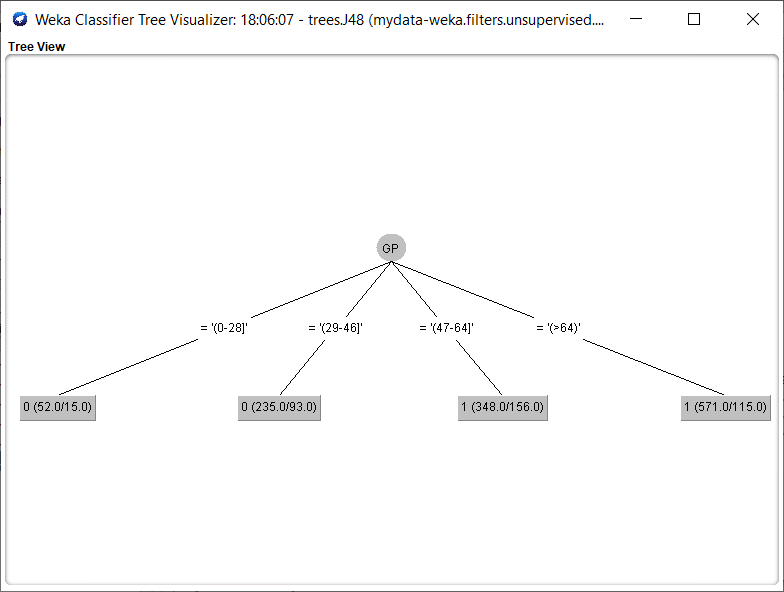


This is the visualisation of the J48 tree for Test 4. The tree is similar to the tree from the training test with the same confidence value.The tree has too many branches and even though detailed is tougher to grasp. We run the next test by decreasing the confidence value.

1. Test 5

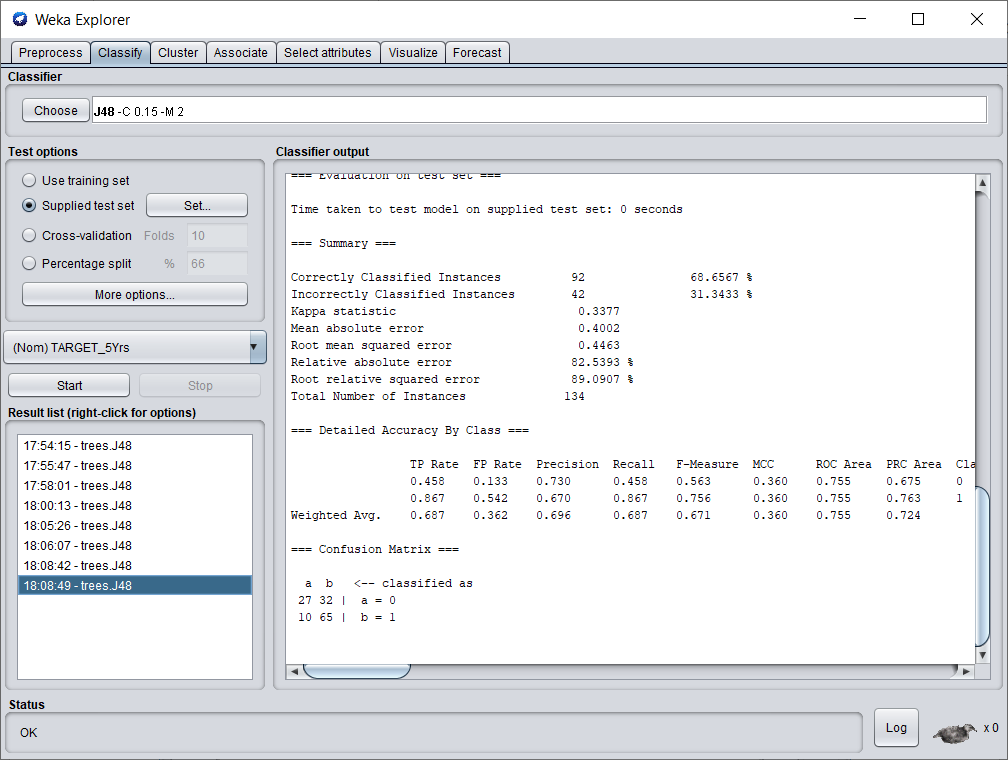


We run the J48 algorithm on the testing dataset with a confidence of 0.01 and a minimum no of objects as 2. We get around 69.4% correctly classified instances which is lesser than the previous test.

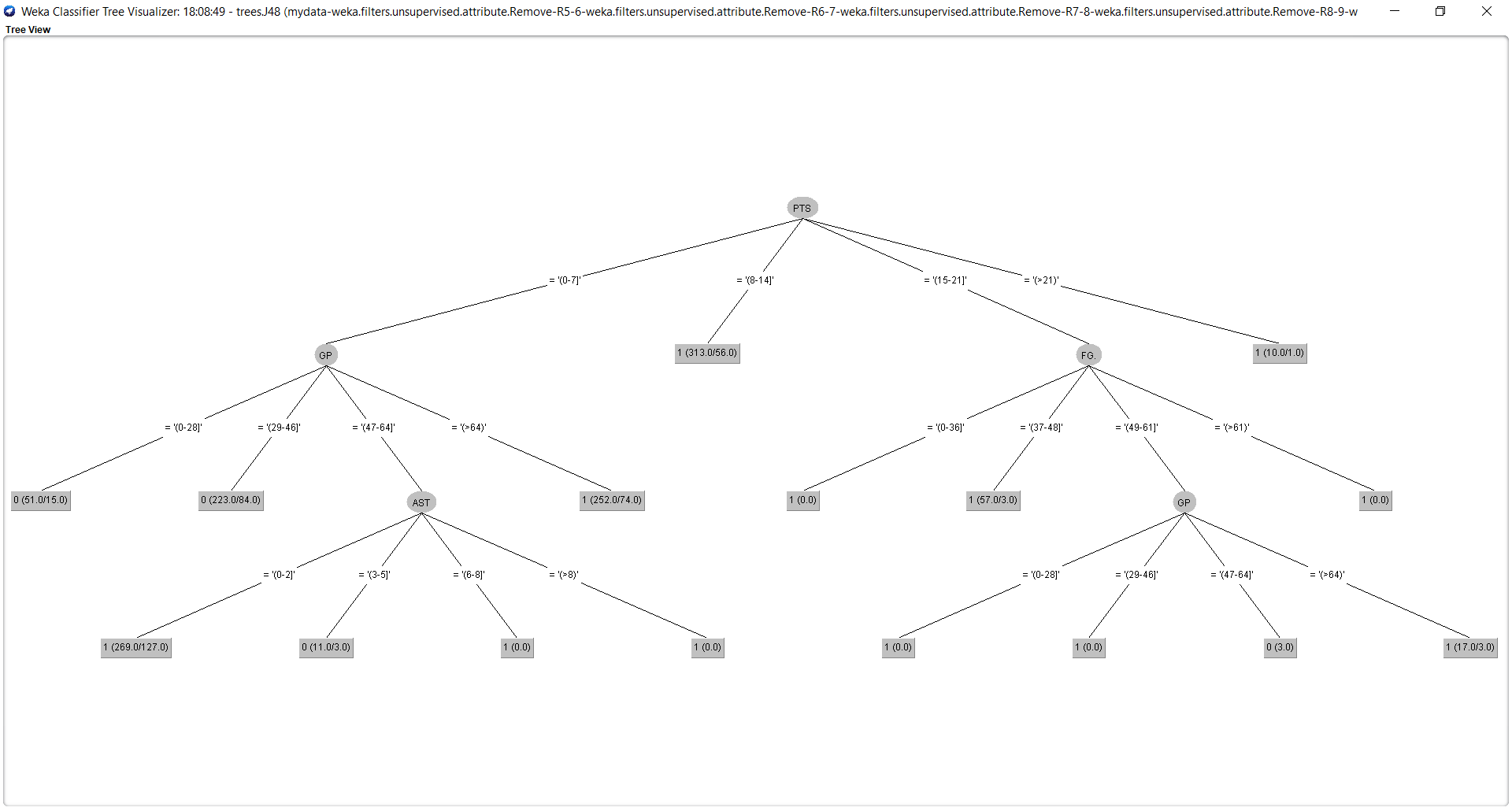


This is the visualisation of the J48 tree for Test 5. Because of setting the confidence level to 0.01, we get an overtly simple tree with very few branches. The tree doesn’t seem to be very useful. We run the next test by increasing the confidence value.

1. Test 6



We run the J48 algorithm on the testing dataset with a confidence of 0.15 and a minimum no of objects as 2. We get around 68.6% correctly classified instances which is lesser than the previous test.

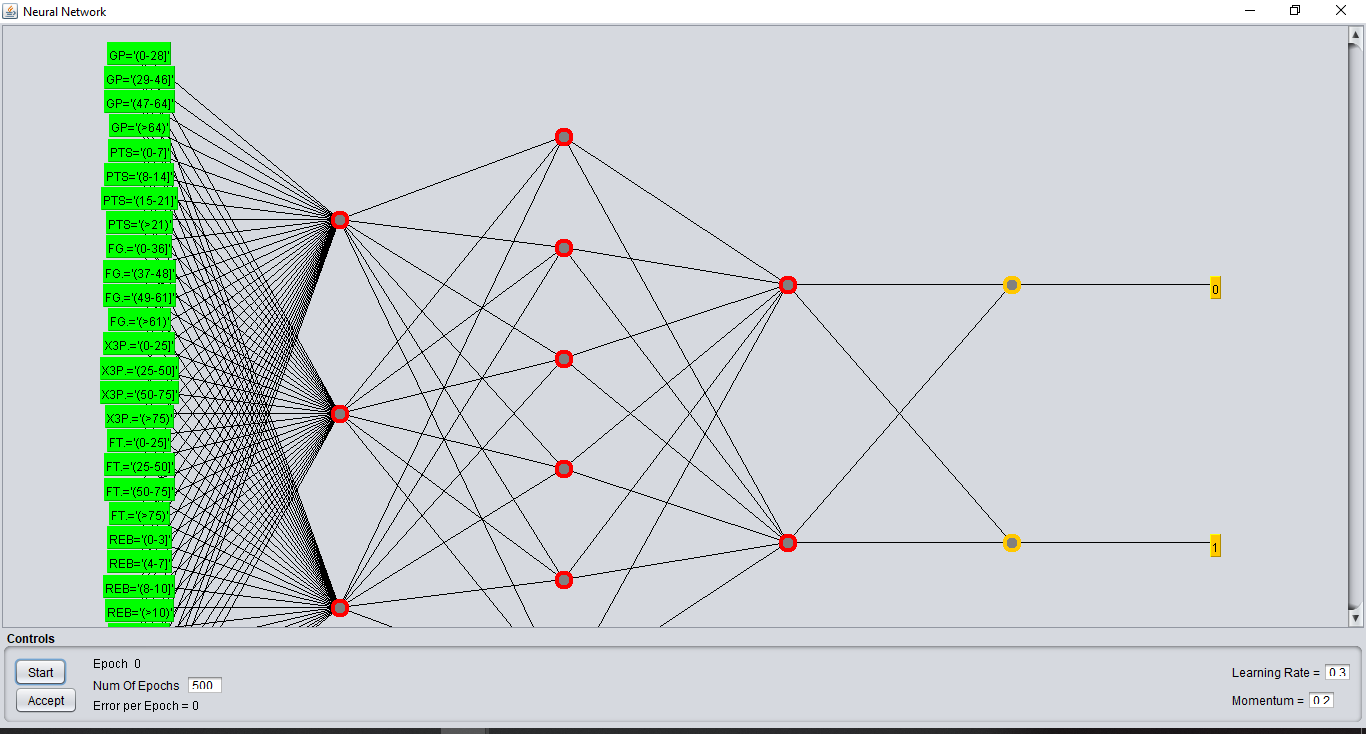


This is the visualisation of the J48 tree for Test 6. The tree seems easy to understand and looks better than the previous 2 tests.

**5. Classification: MLP or a similar advanced technique from Weka – 15%**

Steps :

1. Initially, we load the training set in the weka.
2. Then we open the classify tab and select the Multilayer Perceptron in the functions section.
3. By clicking on the tab, next to the choose button we configure the multilayer perceptron.
4. Here , we have used three hidden layers namely having 3,6,2 nodes respectively.
5. We set the value of GUI to true.
6. Next we run the algorithm.
7. The following image is the neural network that is obtained.



1. As we have many attributes, the layer in green is the layer of inputs.
2. We have the outputs 0(no) and 1(yes) in yellow.
3. The result obtained in weka is as follows:

=== Evaluation on training set ===

Time taken to test model on training data: 0 seconds

=== Summary ===

Correctly Classified Instances 860 71.3101 %

Incorrectly Classified Instances 346 28.6899 %

Kappa statistic 0.4065

Mean absolute error 0.3734

Root mean squared error 0.4329

Relative absolute error 79.8186 %

Root relative squared error 89.5157 %

Total Number of Instances 1206

=== Detailed Accuracy By Class ===

TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class

0.698 0.278 0.599 0.698 0.645 0.410 0.761 0.598 0

0.722 0.302 0.801 0.722 0.759 0.410 0.761 0.826 1

Weighted Avg. 0.713 0.293 0.725 0.713 0.717 0.410 0.761 0.741

=== Confusion Matrix ===

a b <-- classified as

314 136 | a = 0

210 546 | b = 1

Summary of findings: We have achieved an accuracy level of 71% which is similar to J48 classsification.

**Clustering**

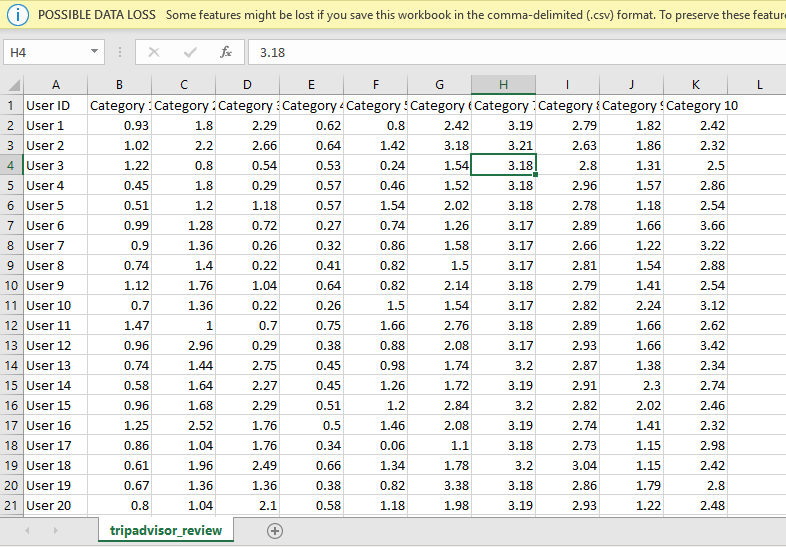
**Initial Tasks**

**1. Description of your dataset and findings – 20%**

* **Title**: Using the Simple K-means clustering algorithm to cluster user reviews on different types of tourists places in East Asia. The user ratings range from 0 to 4 with 0 being the lowest and 4 being the highest.
* **Data description:** A description of the data in detail under the following subheadings:
  + The problem domain: The dataset includes 11 attributes and 980 tuples. The user reviews are numeric values ranging from 0 to 4. This is a social media dataset and is built from trip advisor data. The problem definition is to group types of tourists places based on user reviews.
  + The source of the data: The dataset was obtained from UCI Machine LearningRepository. [https://archive.ics.uci.edu/ml/datasets/Travel+Reviews#](https://archive.ics.uci.edu/ml/datasets/Travel+Reviews)
  + The agencies working with the data : This dataset was used by the professors of Cochin University of Science and Technology(CUSAT), Kochi, India as a part of research.
  + The intended use of the data : This data is to used to test various clustering algorithms like K- means, K-mediods, CLARA etc.
  + The attribute types of the data : The dataset has 11 attributes and they are as follows :

1. User id : Unique user id (Nominal)
2. Category 1 : Average user feedback on art galleries (Numeric)
3. Category 2 : Average user feedback on dance clubs (Numeric)
4. Category 3 : Average user feedback on juice bars (Numeric)
5. Category 4 : Average user feedback on restaurants (Numeric)
6. Category 5 : Average user feedback on museums (Numeric)
7. Category 6 : Average user feedback on resorts (Numeric)
8. Category 7 : Average user feedback on park/picnic spots (Numeric)
9. Category 8 : Average user feedback on beaches (Numeric)
10. Category 9 : Average user feedback on theaters (Numeric)
11. Category 10 : Average user feedback on religious institutions.(Numeric)

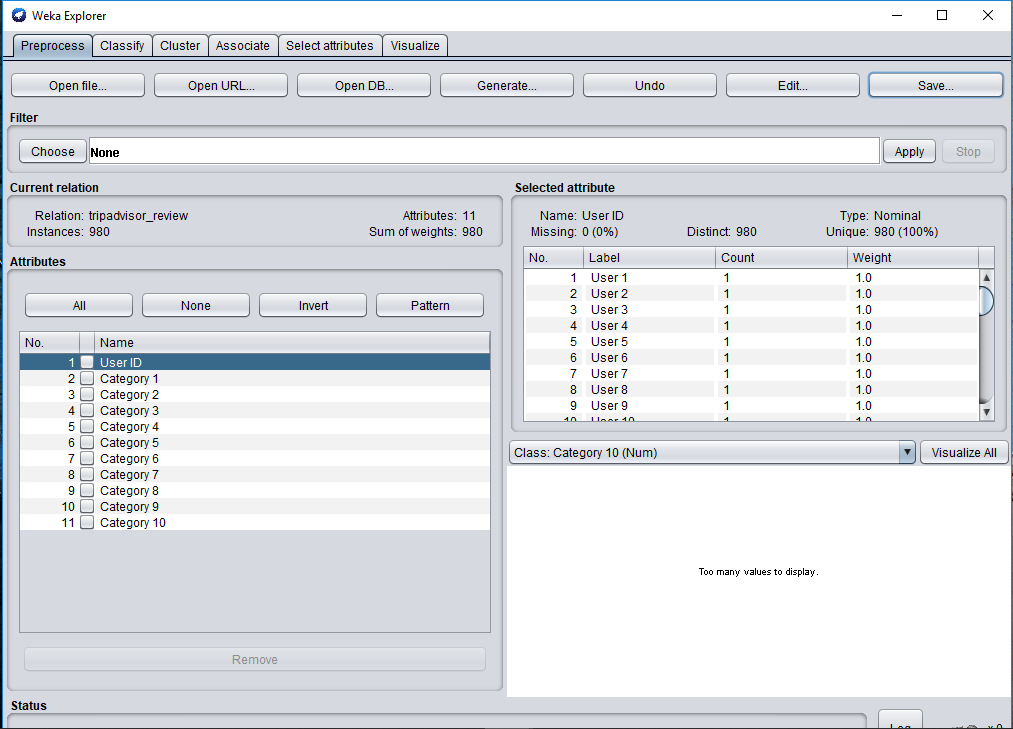
The following image shows the csv view of the dataset.



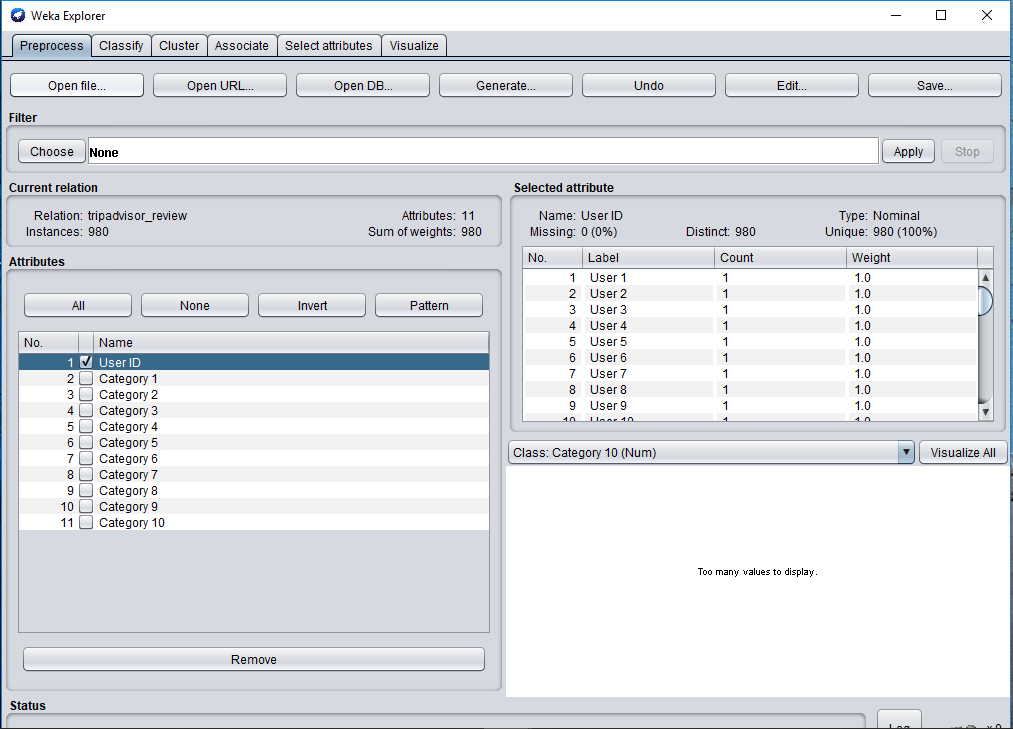
* **Objective**: The technique used on this dataset is K-Means clustering . Two of the numeric attributes will be selected that are most suitable for this algorithm. The main objective is to find clusters of user reviews(based on value of ratings) of the selected 2 types of attributes.

**2. Preprocessing – 10%**

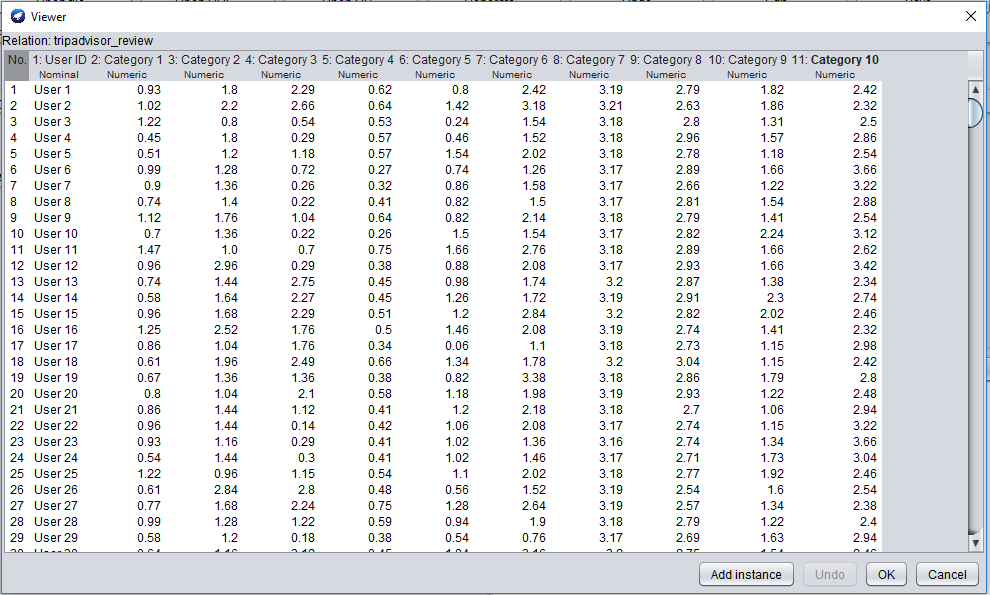
We will be using the Weka software for performing K-means clustering. The following image shows the view of the data in weka.



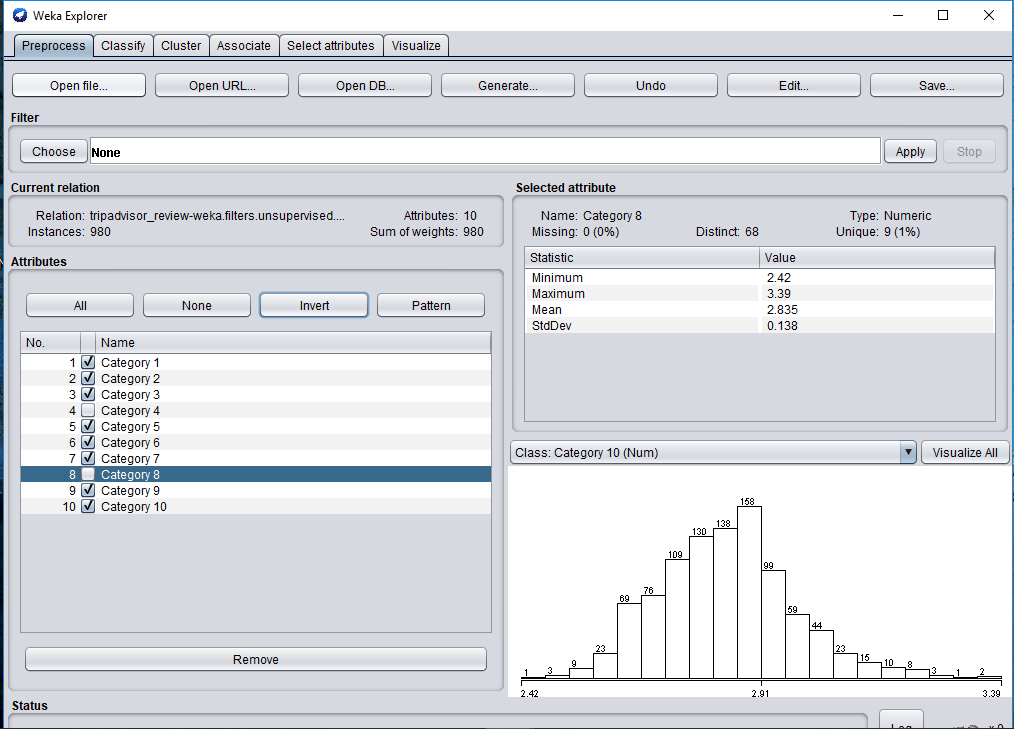
The user id attribute is a nominal attribute and has a different value for every tuple. Hence it is unnecessary for this technique we remove this attribute as shown below:



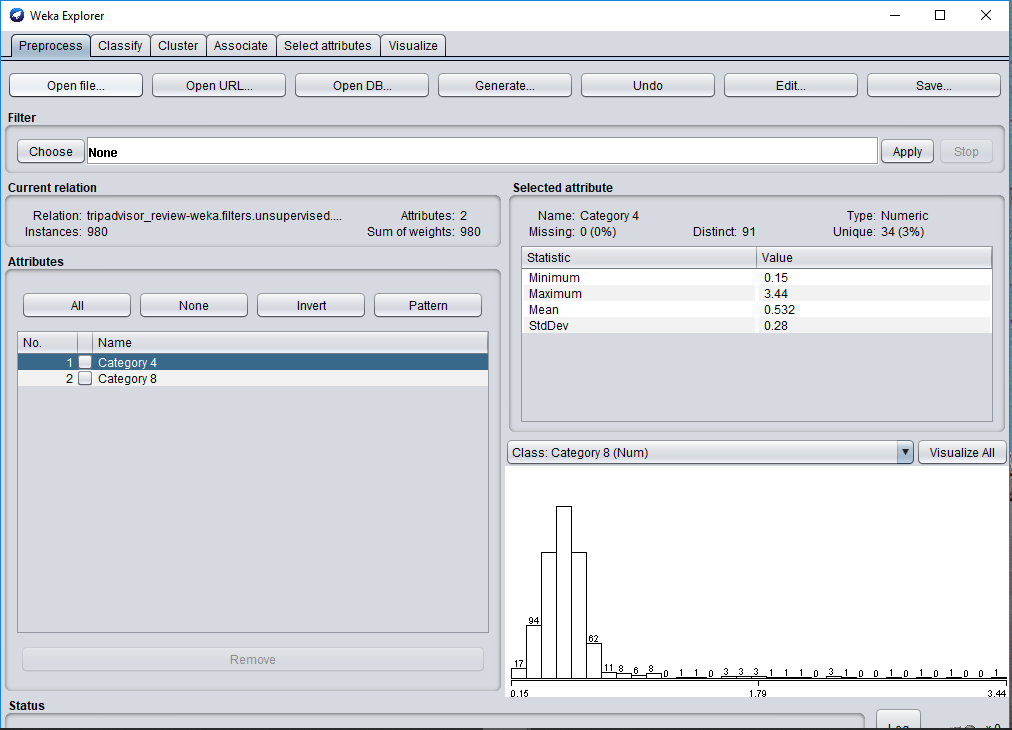
This dataset has no missing values and hence does not require any specific technique for removing the missing values. This can be seen from the following image.



The two attribute that are selected for this are Category 4 and Category 8 . Hence we remove all other attributes as shown below for accurate results. We save the resultant file as dataset-clustering.arff.



dataset-clustering.arff :

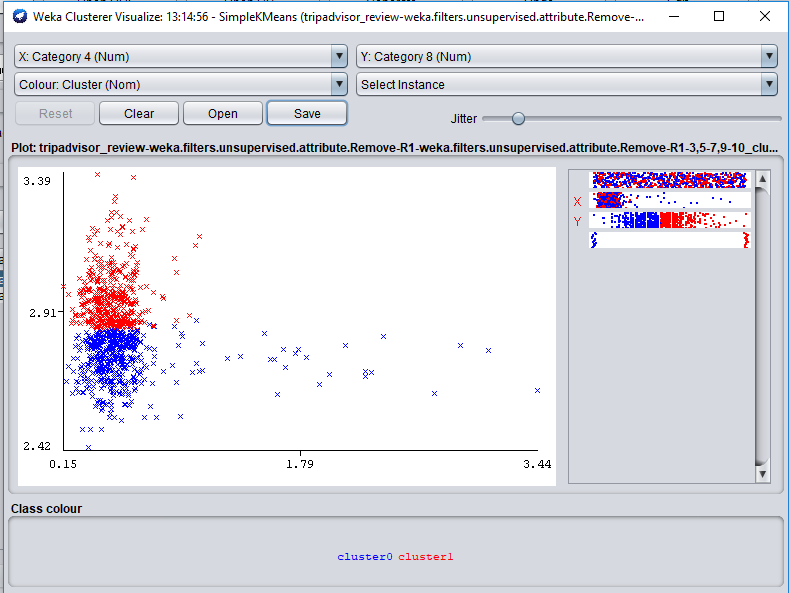


1. **Clustering: K-Means or DBSCAN – 10%**

**The technique used is Simple K-means clustering in Weka.**

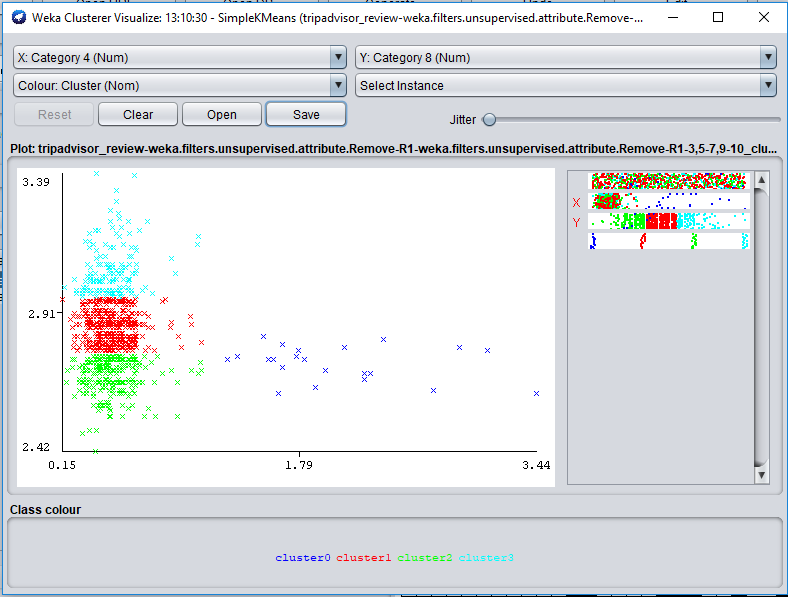
Steps :

1. We choose the cluster tab in weka and then choose the Simple K- means algorithm from the choose section . We initially set the value of cluster to 2 clusters and then run the algorithm. The following cluster assignments are obtained:



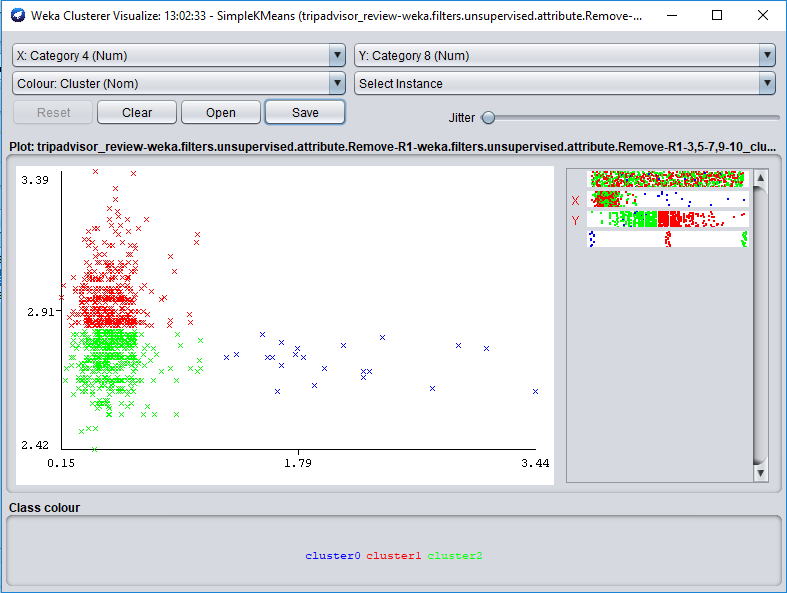
The two clusters are generated in red and blue . However, from the graph we can see that the points that are far away from cluster in blue are also considered in the same cluster.

So now we change the value of cluster to 4 and then the following cluster assignments are obtained :



By comparing to the previous clusters, we see that the two large clusters are now converted in 3 clusters. The points of these clusters are close by and can be grouped in fewer clusters.

Hence , now we set the value of clusters to 3 and run the algorithm.



The above result is accurate as compared to the above two. The two main clusters are in red and green and the points that are far away from the cluster are in blue . Hence we select this as our final result.

Result of clustering :

1. The attributes used for clustering are Category 4 ( Average rating for Juice bars) and Category 8 (Average rating for resorts).
2. We set them as follows :

Category 4 : Atrribute on X axis of the graph

Category 8 : Attribute on Y axis of the graph

1. The results obtained after running the experiment in weka are as follows :

=== Run information ===

Scheme: weka.clusterers.SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 3 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1 -S 10

Relation: tripadvisor\_review-weka.filters.unsupervised.attribute.Remove-R1-weka.filters.unsupervised.attribute.Remove-R1-3,5-7,9-10

Instances: 980

Attributes: 2

Category 4

Category 8

Test mode: evaluate on training data

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

kMeans

======

Number of iterations: 8

Within cluster sum of squared errors: 9.891700192070802

Initial starting points (random):

Cluster 0: 0.64,2.79

Cluster 1: 0.53,2.82

Cluster 2: 0.45,2.69

Missing values globally replaced with mean/mode

Final cluster centroids:

Cluster#

Attribute Full Data 0 1 2

(980.0) (22.0) (452.0) (506.0)

========================================================

Category 4 0.5325 2.0536 0.493 0.5016

Category 8 2.8351 2.725 2.9513 2.736

Time taken to build model (full training data) : 0 seconds

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

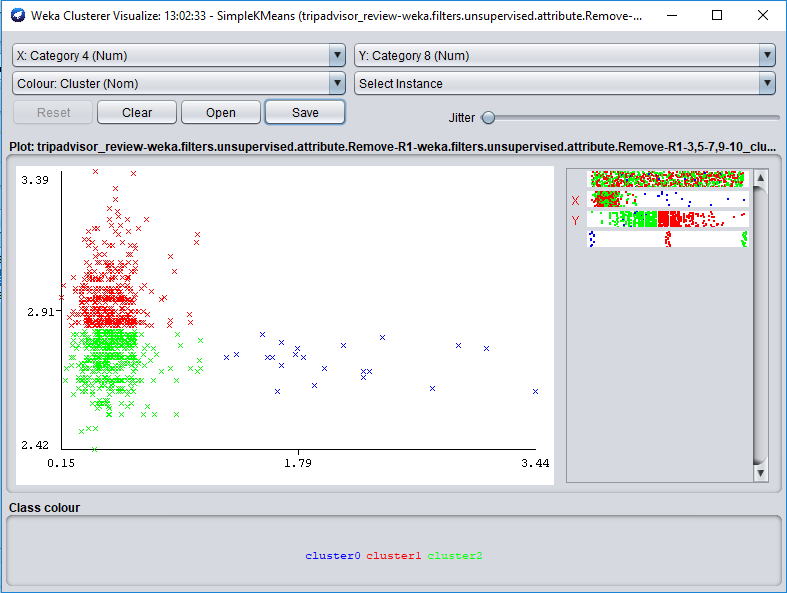
Clustered Instances

0 22 ( 2%)

1 452 ( 46%)

2 506 ( 52%)

Visualizing cluster assignments :



Summary of findings :

1. Namely 3 clusters are formed :

Cluster 0 22 ( 2%)(cluster in blue)

Cluster 1 452 ( 46%)(cluster in red)

Cluster 2 506 ( 52%)(Cluster in green)

1. Very few of the user ratings for juice bars are higher than 1. That is juice bars have very low ratings.
2. Comparatively, user ratings for resorts are mostly high ranging from 2.5 to 3.5.
3. Most of the user ratings lie between the range of 2.7 to 3.1 approximately.
4. Finally, user rating for resorts are better than those for juice bars.
5. The graph gets denser as it moves up vertically and it gets sparse as it moves right horizontally.

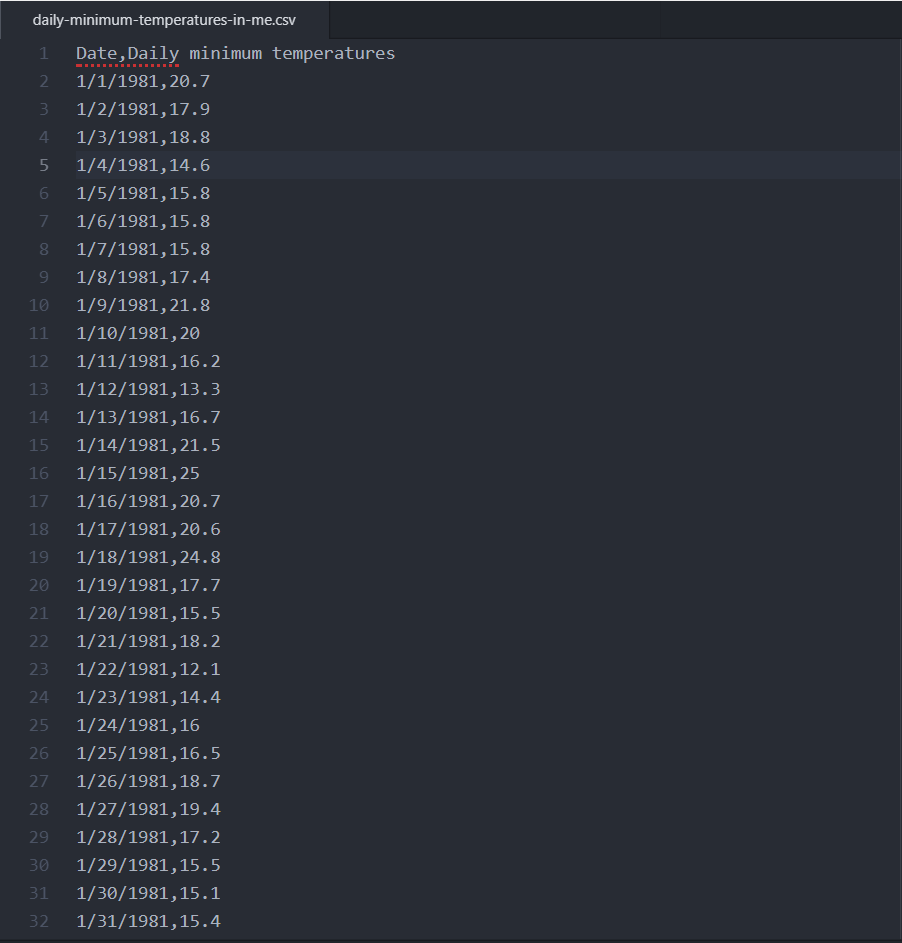
**Time Series Forecasting**

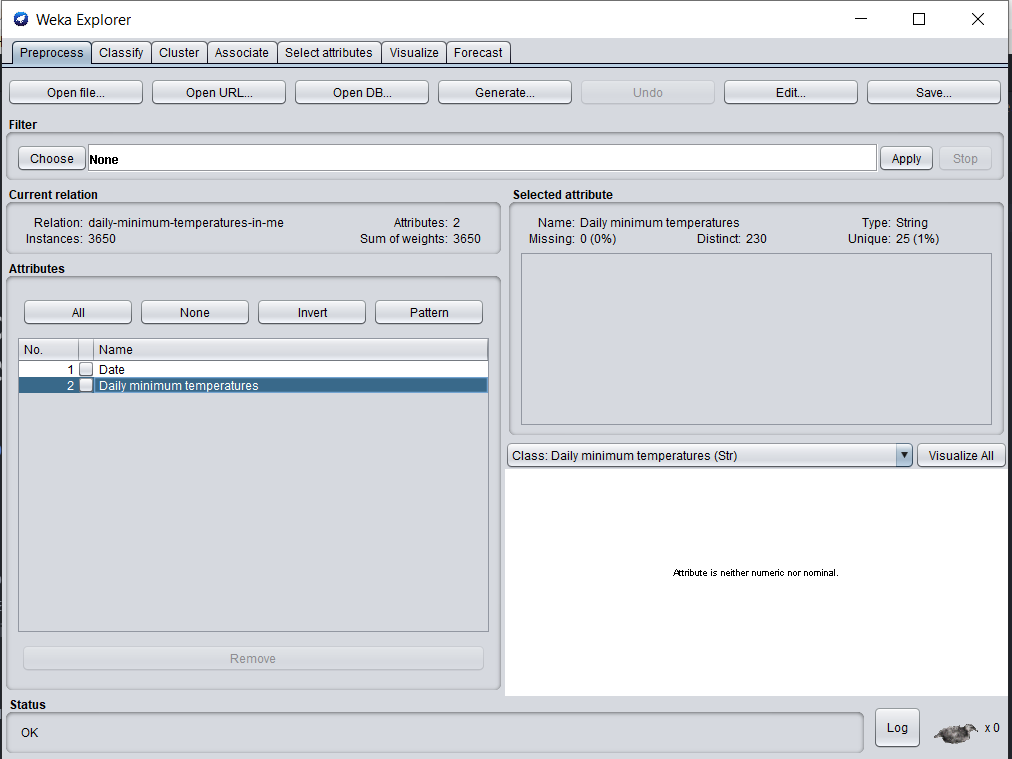
1. **Time Series Forecasting – 15%**

* We use the daily-minimum-temperatures-in-me dataset which has the Daily minimum temperatures in Melbourne, Australia from 1981-1990.
* **Source :** <https://www.kaggle.com/shenba/time-series-datasets>

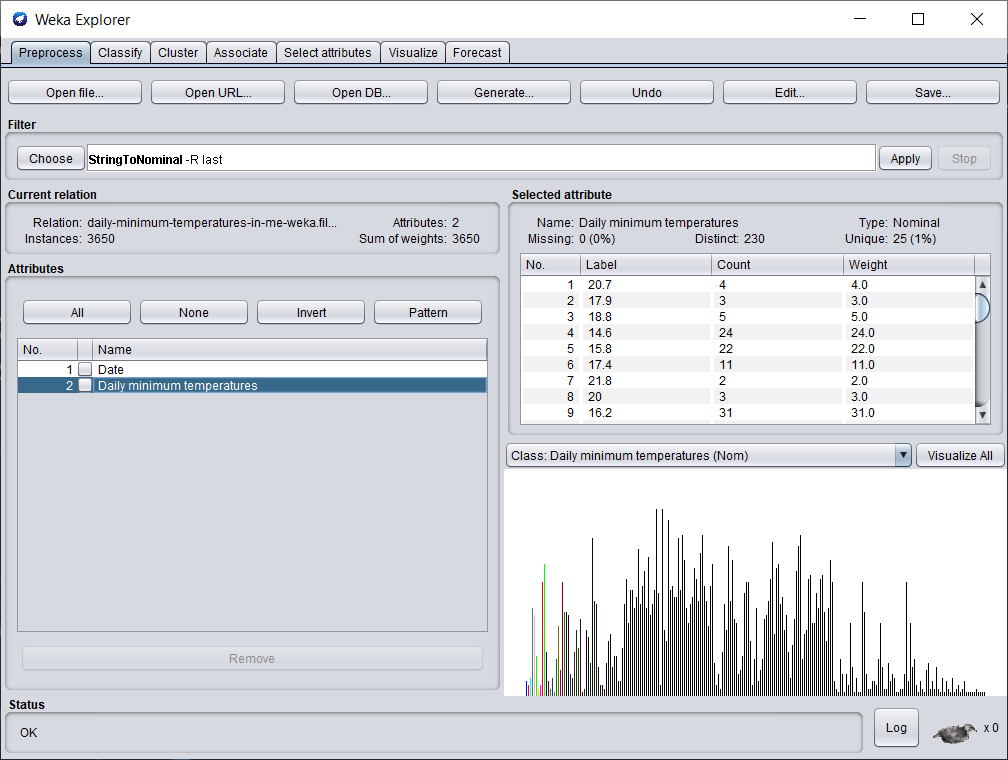
<https://www.kaggle.com/paulbrabban/daily-minimum-temperatures-in-melbourne>

* **Objective**: To predict the minimum temperatures in Melbourne for the succeeding two years using historical data.
* **Original dataset:** The dataset has 2 attributes, Date and Daily minimum temperatures. There are 3650 instances and no missing values.



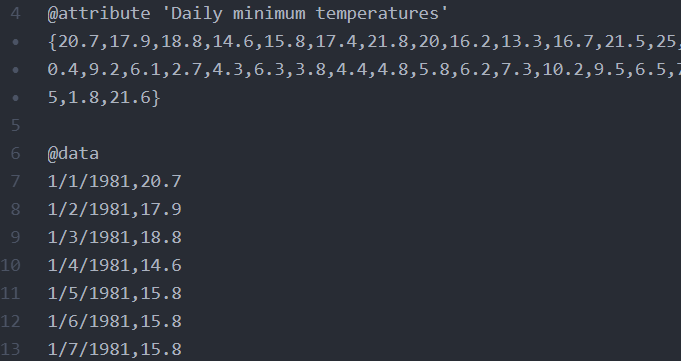
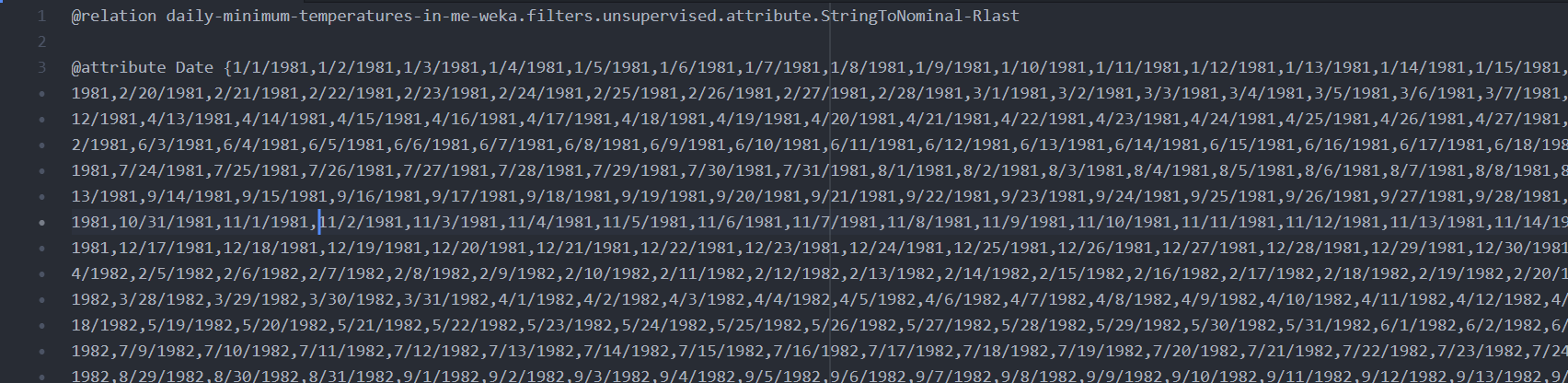
* The Date attribute is of the type Nominal and Daily minimum temperatures is in String format. 
* **Preprocessing**

1. We initially need to convert the Date attribute to date type and Daily minimum temperatures to numeric type. We use StringToNominal filter to initially convert Daily minimum temperatures to the nominal type.

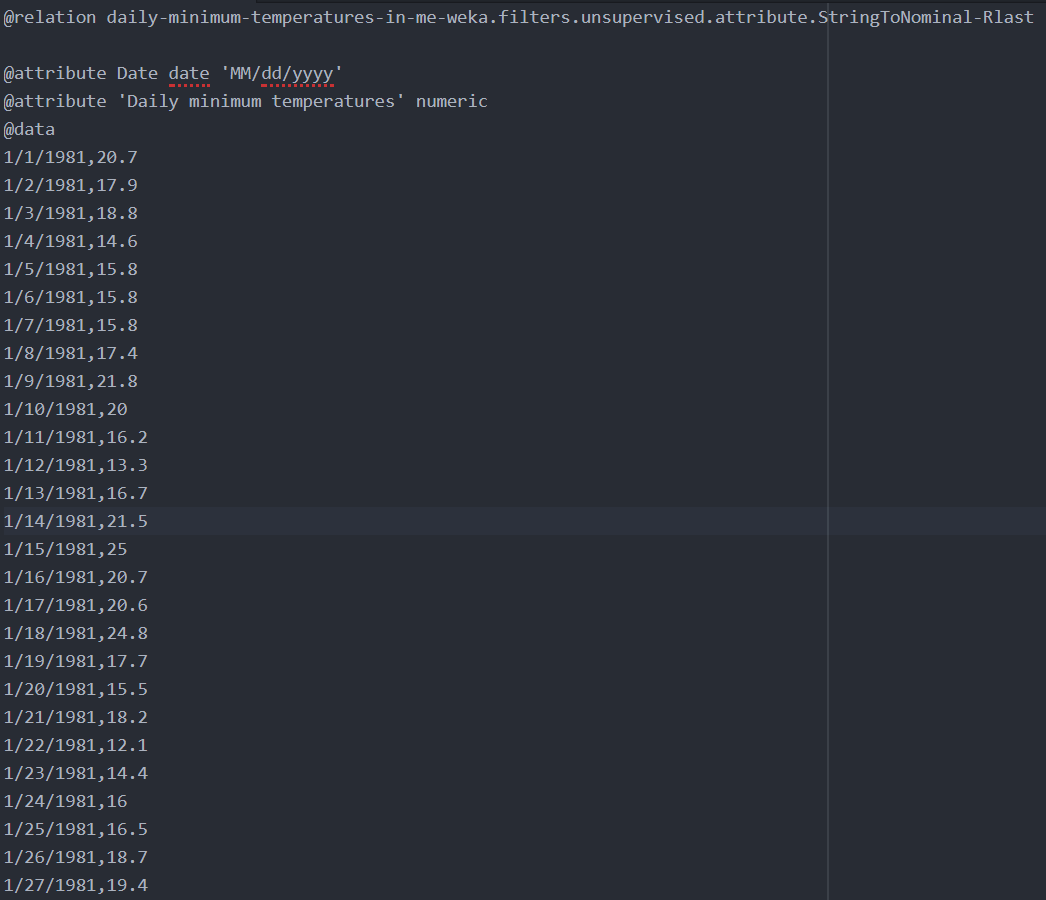


1. Next, we use a text editor to convert the attributes to their desired data type.

* Initial data type

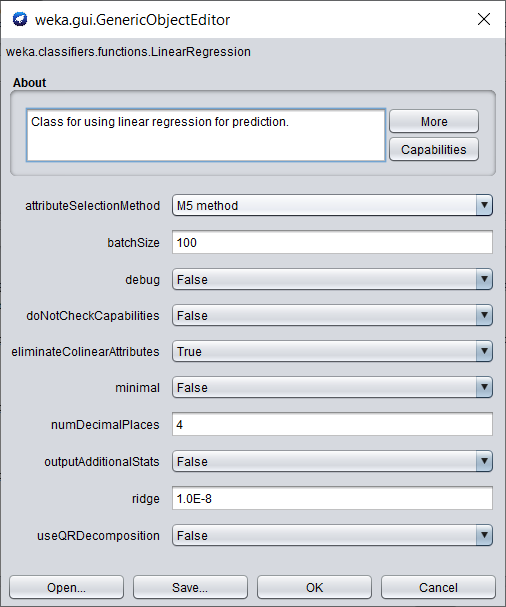


* Converted data type:

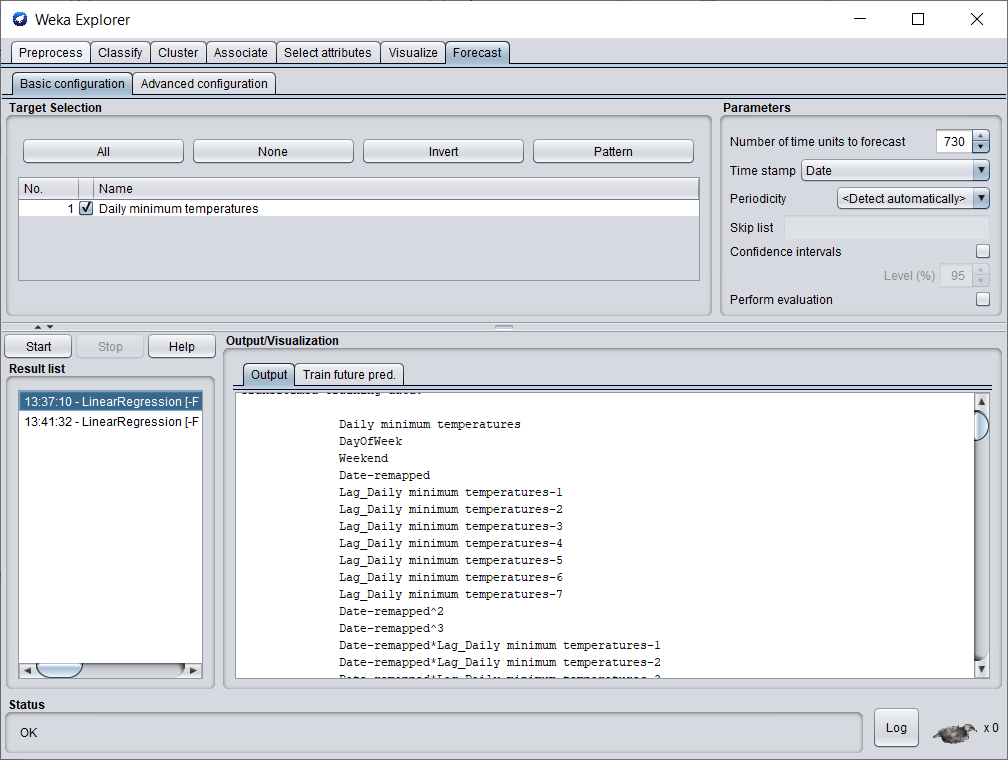


1. Write one or two paragraph analyzing the results of the forecasting. Support this analysis with screenshots of
   1. The regression equation
   2. Diagram of the historical values
   3. Diagram of the predictions

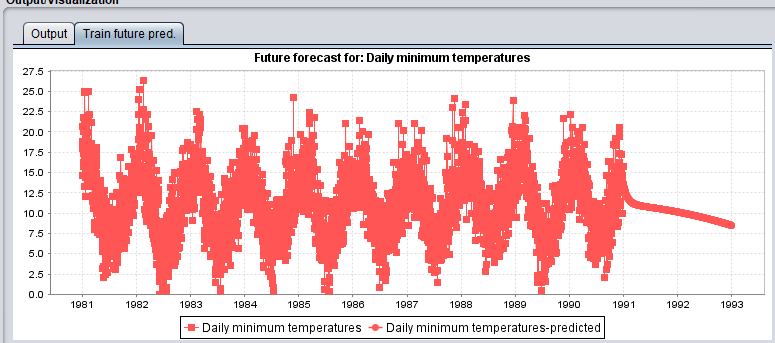
We use daily minimum temperatures as our target selection and date as the time stamp. Since we want to predict the daily minimum temperatures for the next two years, we set number of time units to forecast as 730. We use the default settings without making any changes.



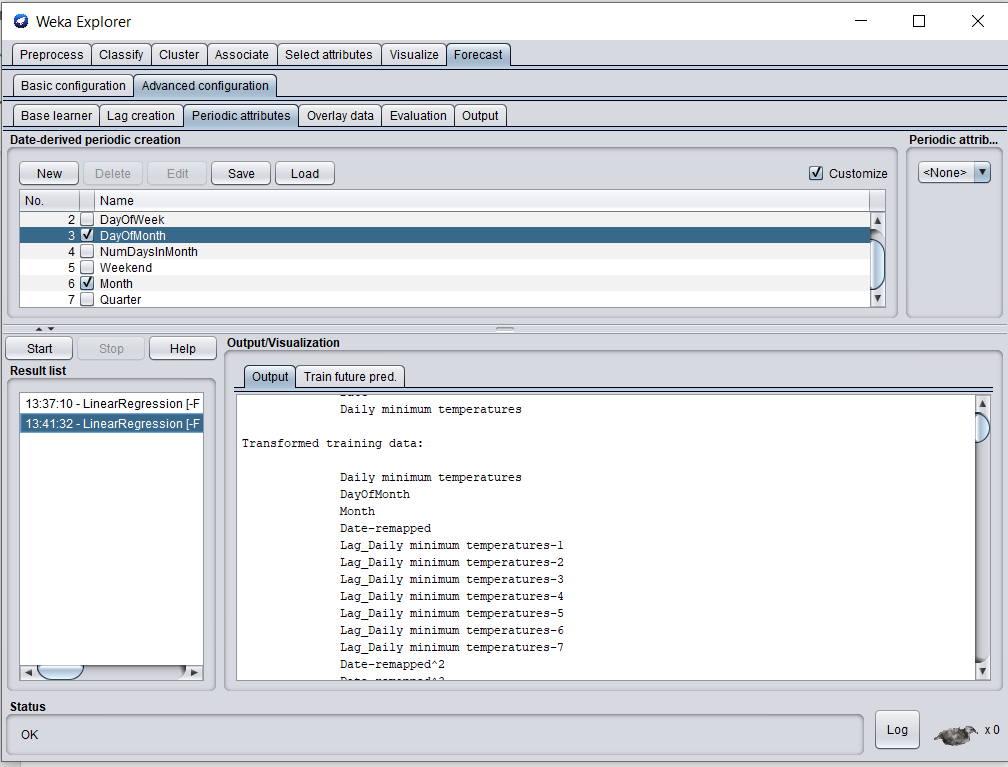
The default algorithm for prediction is Linear Regression which uses the Akaike criterion for model selection. The Akaike information criterion (AIC) is an estimator of the relative quality of statistical models for a given set of data. The evaluation metrics chosen are Mean absolute error (MAE) and Root mean squared error (RMSE).



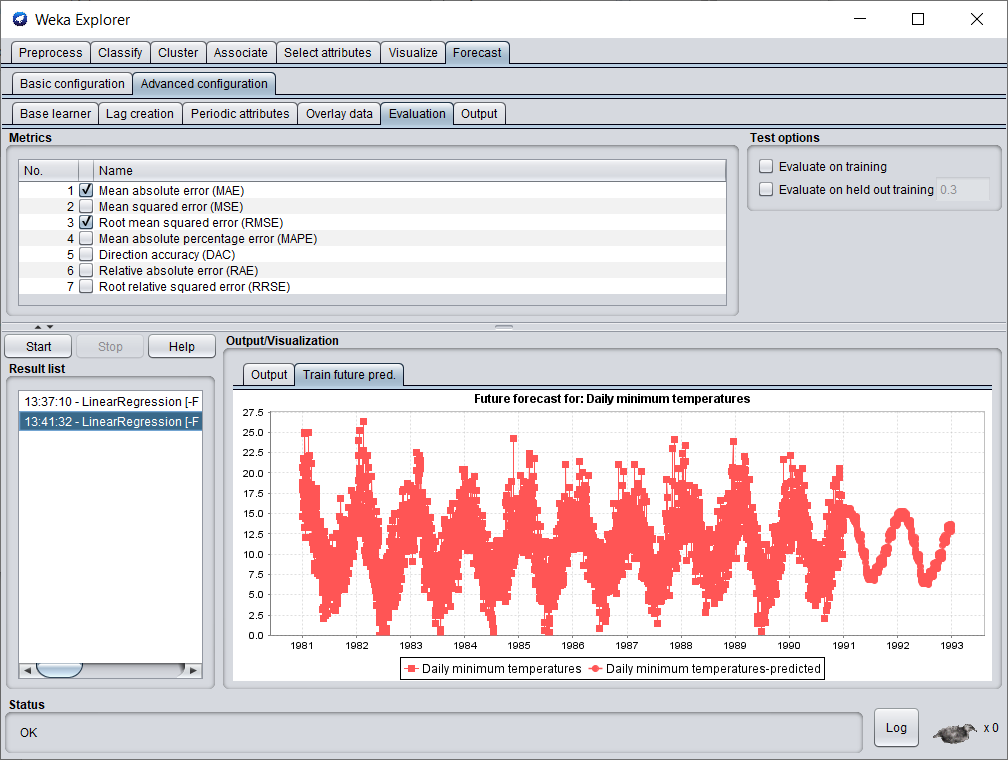
On choosing default values, DayOfWeek and Weekend are set as periodic attributes.



The forecast from the start of 1981 to the end of 1990 are historical values and from the start of 1991 to the end of 1992 are the predicted values. We can observe that the predictions seem to be incorrect as the predicted temperatures seem to be decreasing daily. The incorrect values could possibly be because the periodic attributes are set to DayOfWeek and Weekend. Hence, we change the periodic attributes and run the test again.



We set the Periodic attributes as DayOfMonth and Month because temperatures can change based on the month. We leave the other attributes as default and use the same algorithm i.e. Linear Regression.



The forecast from the start of 1981 to the end of 1990 are historical values and from the start of 1991 to the end of 1992 are the predicted values. By changing the periodic attributes, we get a better prediction of the daily temperatures which seem to emulate the same pattern as the historical values.

Conclusion: Using historical data, we predicted the minimum temperature in Melbourne for the succeeding two years.