# FINAL PROJECT REPORT

*CS699 - DATA MINING*

BY PREETI CHOUBEY

## TABLE OF CONTENTS

## DATASET:

The project-2018-BRFSS-arthritis.csv dataset included 11933 tuples and 108 attributes. Each tuple is a person who participated in the survey and each attribute represents an answer to a survey question. The class attribute is havarth3 and its value is either 1 or 2. 1 means that the individual was diagnosed with arthritis at some point, while 2 means that they haven't.

# DATA PREPROCESSING:

## HANDLING MISSING VALUES

There are no missing values for havarth3 (the label). For other attributes, I used Weka's ReplaceMissingValues filter. It replaces all missing values for the numeric and nominal attributes in a dataset with the mean and mode from the data, respectively. For this, I followed these steps.

From choose button > weka > Filters > unsupervised > attribute > replaceMissingValues > apply > Save.

## STANDARDIZATION

Standardized numeric attributes to zero mean and unit variance.

## REMOVING DUPLICATES

Unnecessary, as there are no duplicate entries.

## FINDING OUTLIERS

From choose button > WEKA > Filters > unsupervised > attribute > interquartile range > apply.

## REMOVE OUTLIERS

choose button > weka > filter > unsupervised > instance > remove with value > click on the filter field (to add outlier attribute indices and nominal indices, I specified it as last value) > apply > remove outliers' attribute.

## RANDOMIZATION

It randomly shuffles the order of the instances passed through it.

## SPLITTING THE DATASET

I split the data into a training set (66% of the data) and a test set (34% of the data).

To do so, I used the RemovePercentage filter (Weka> filter>unsupervised > instance)

For the training set:

1. Load the full dataset
2. Select the RemovePercentage filter in the preprocess panel

3. Set the split to 34%
4. Apply the filter
5. Save the generated data as project-training.arff

For test set:

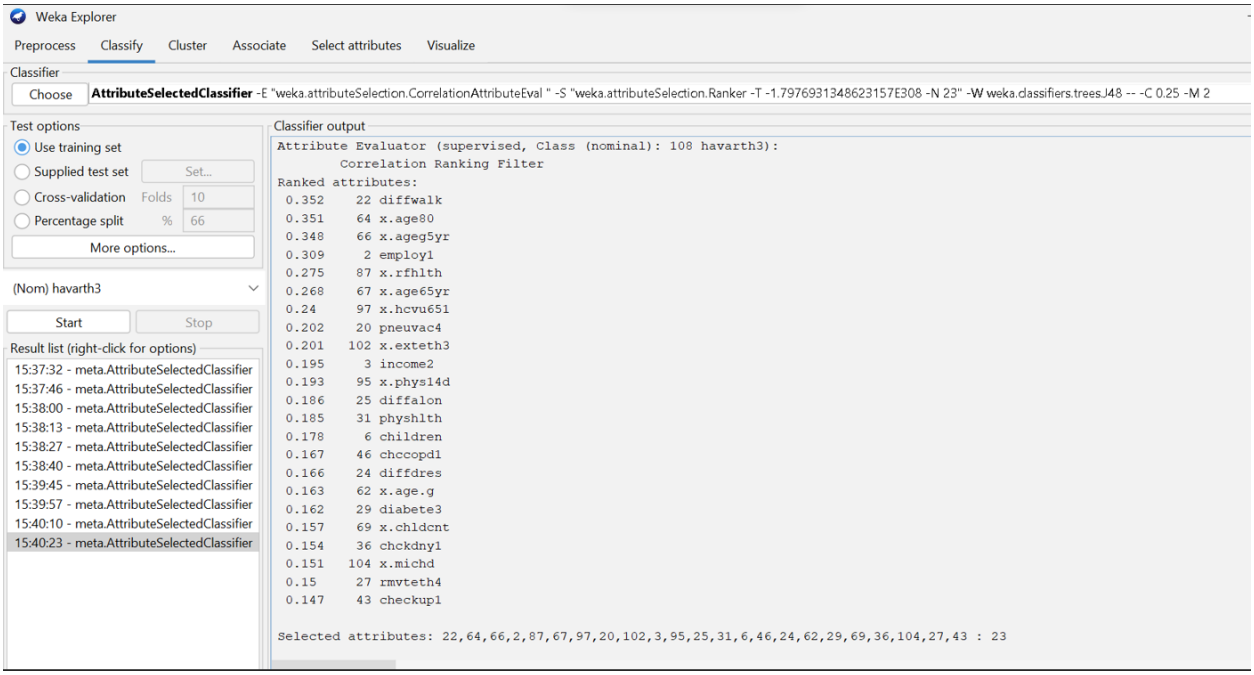
1. Load the full dataset
2. Select the RemovePercentage filter
3. Set the invert Selection property to true
4. Apply the filter
5. Save the generated data as project-test.arff

## FEATURE SELECTION

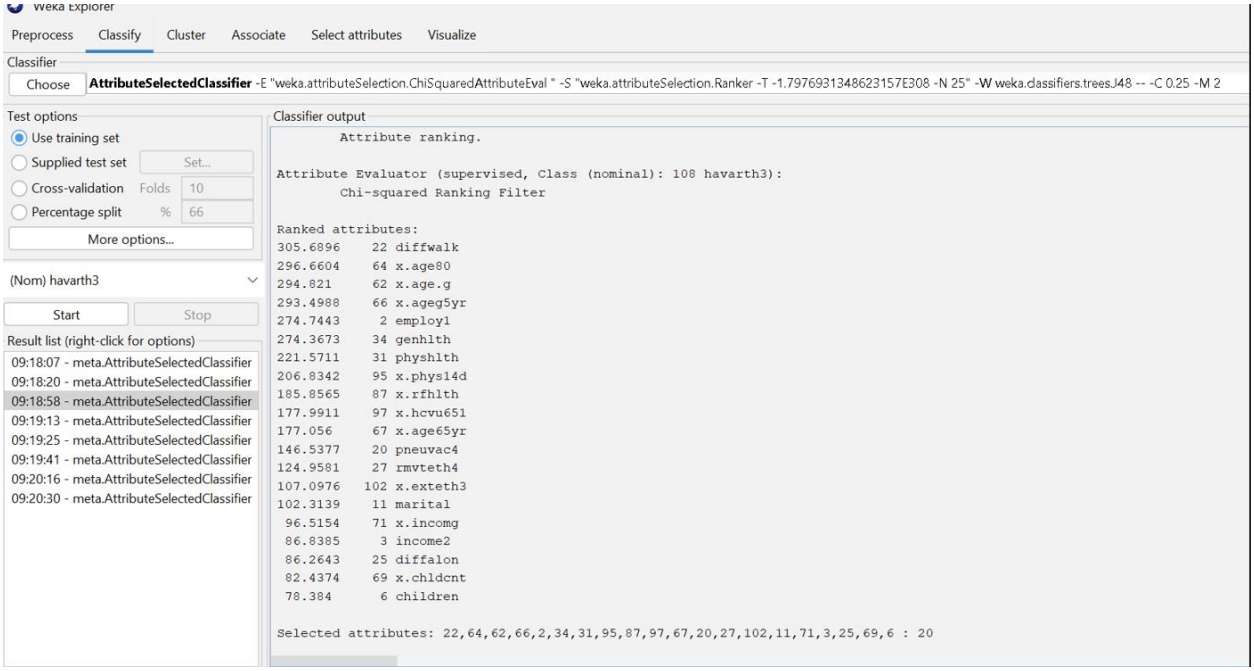
For attribute selection, I used the following attribute selection methods:

- CorrelationAttributeEval: Evaluates the worth of an attribute by measuring the correlation (Pearson's) between it and the class.
- ChiSquaredAttributeEval: Evaluates the worth of an attribute by computing the value of the chi-squared statistic for the class.
- InfoGainAttributeEval: Evaluates the worth of an attribute by measuring the information gain for the class.
- GainRatioAttributeEval: Evaluates the worth of an attribute by measuring the gain ratio for the class
- OneRAttributeEval: Evaluates the worth of an attribute by using the OneR classifier.

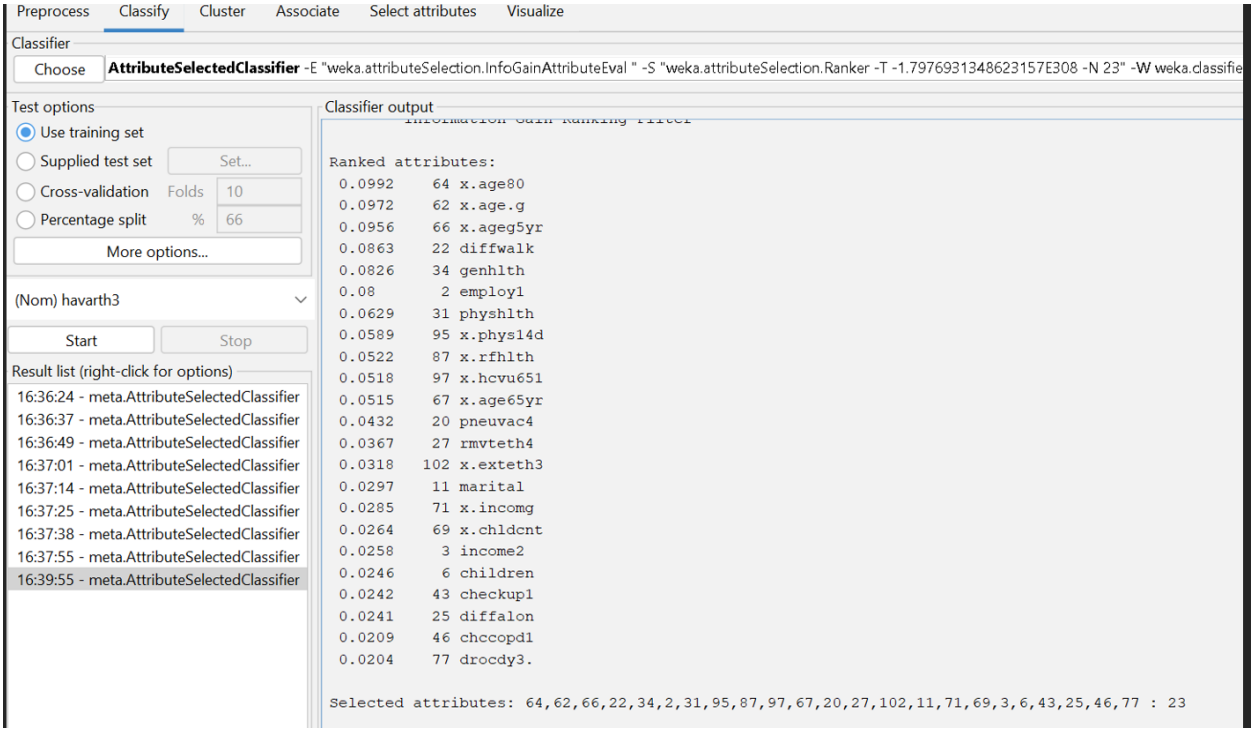
## SELECTED ATTRIBUTE BASED ON CORRELATIONATTRIBUTEEVAL



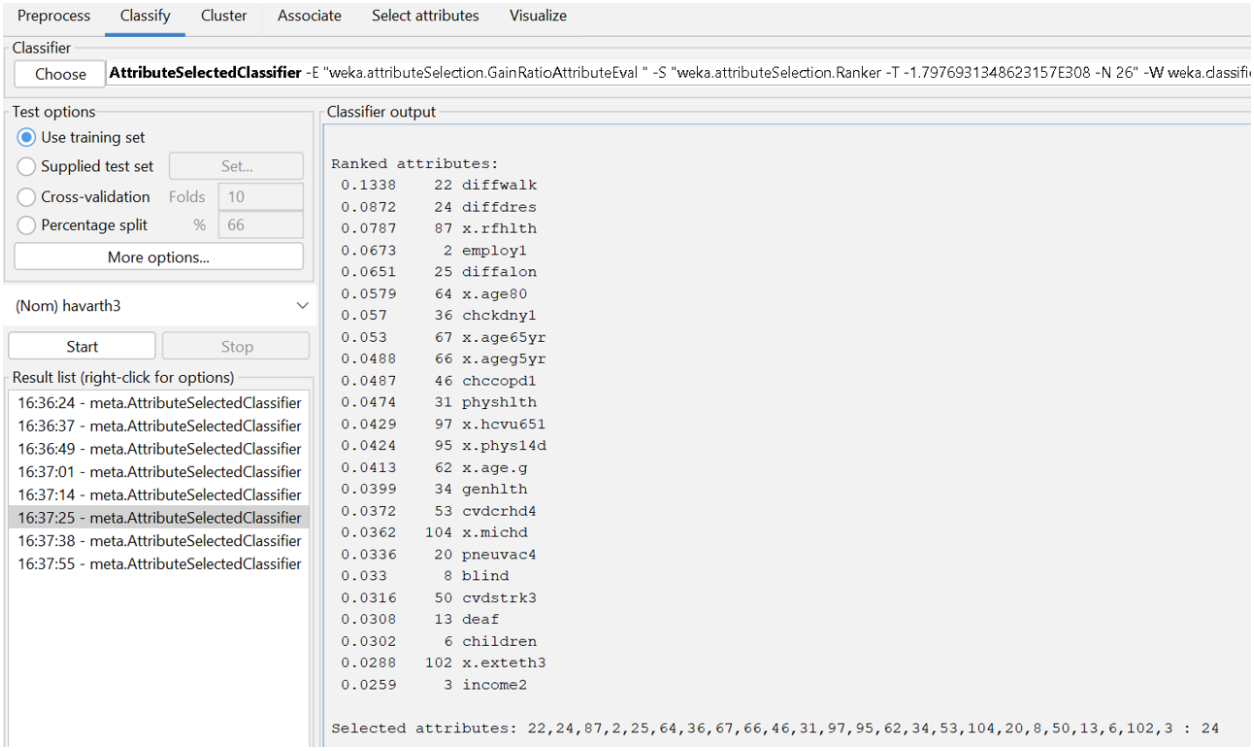

## SELECTED ATTRIBUTE BASED ON CHISQUAREDATTRIBUTEEVAL



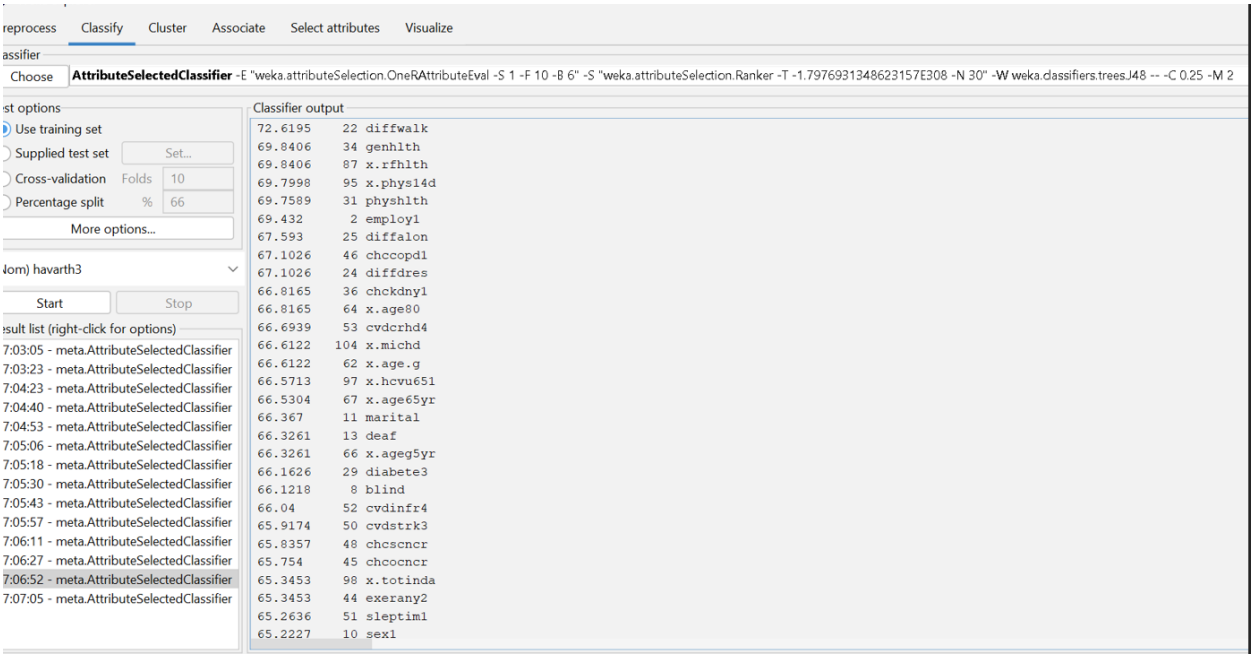
## SELECTED ATTRIBUTE BASED ON INFOGAINATTRIBUTEEVAL




## SELECTED ATTRIBUTE BASED ON GAINRATIOATTRIBUTEEVAL



## SELECTED ATTRIBUTE BASED ON ONERATTRIBUTEEVAL




```
Selected attributes: 22,34,87,95,31,2,25,46,24,36,64,53,104,62,97,67,11,13,66,29,8,52,50,48,45,98,44,51,10 : 29
```

# TRAINING

## CLASSIFIERS USED

- Naive Bayes
- Logistic
- Support Vector Machine
- J48 Decision Tree
- Bagging (Random Forest)

NAIVE BAYES: Naive Bayes is a supervised machine learning algorithm that is based on the Bayes Theorem with an assumption that all the features that predict the target value are independent of each other. It calculates the probability of each class and then picks the one with the highest probability.

LOGISTIC REGRESSION: Logistic regression is a supervised classification algorithm, and it is used in predictive analytics. Logistic regression estimates the probability of an event occurring, based on a given dataset of independent variables. Since the outcome is a probability, the dependent variable is bounded between 0 and 1. In logistic regression, a logit transformation is applied to the odds—that is, the probability of success divided by the probability of failure. This is also commonly known as the log odds or the natural logarithm of odds, and this logistic function is represented by the following formulas:

Logit(pi) = 1/ (1+ exp(-pi))

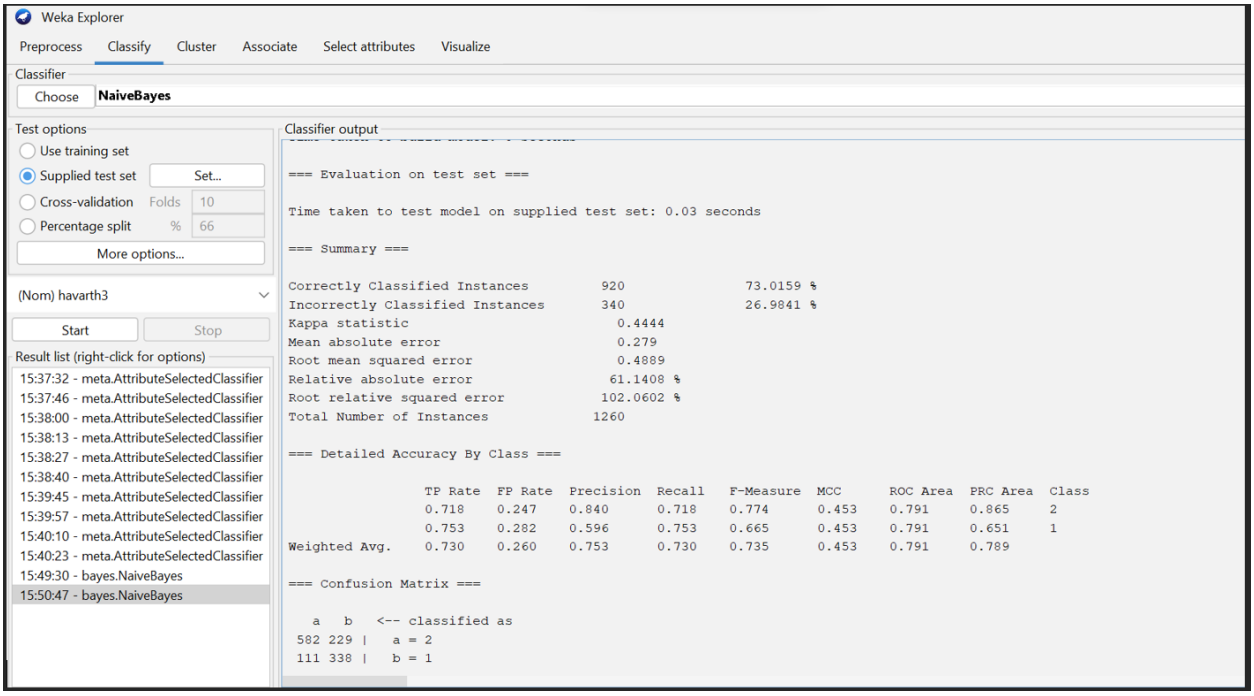
BAGGING: Bagging, also known as Bootstrap aggregating, is an ensemble learning technique that helps to improve the performance and accuracy of machine learning algorithms. It is used to deal with bias-variance trade-offs and reduces the variance of a prediction model. Bagging avoids overfitting of data and is used for both regression and classification models, specifically for decision tree algorithms. Random Forest is an improvement over the bagged decision tree.

SVM: Support Vector Machine or SVM is a Supervised Machine Learning algorithm, which is used for Classification as well as Regression problems. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane. Compared to newer algorithms like neural networks, they have two main advantages: higher speed and better performance with a limited number of samples (in the thousands). This makes the algorithm very suitable for text classification problems, where it's common to have access to a dataset of at most a couple of thousands of tagged samples.

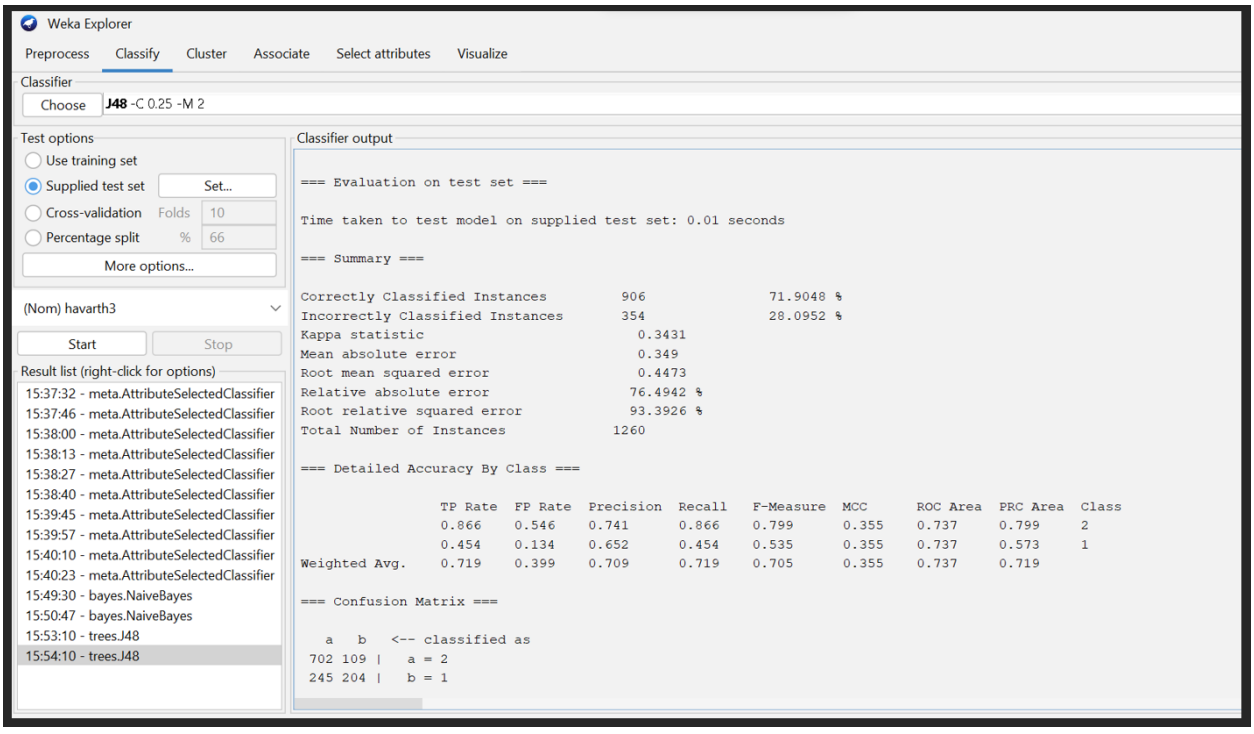
DECISION TREE: A decision tree is a supervised machine learning algorithm that is used for classification. This algorithm generates the outcome as the optimized result based upon the tree structure with the conditions or rules. The decision tree algorithm is associated with three major components as Decision Nodes, Design Links, and Decision Leaves. It operates with the Splitting, pruning, and tree selection process. It supports both numerical and categorical data to construct the decision tree. Decision tree algorithms are efficient for large datasets with less time complexity.

## TEST RESULTS OF ALL 25 MODELS.

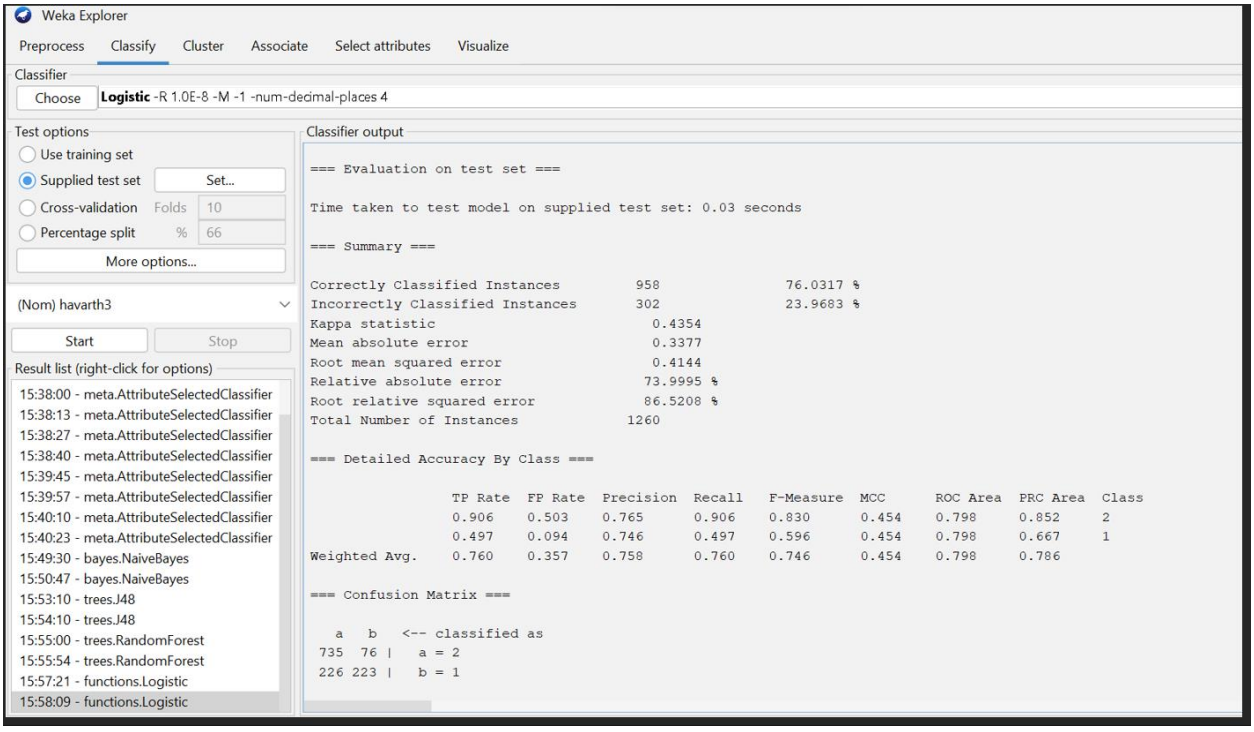
### CORRELATION ATTRIBUTE SELECTION WITH NAIVE BAYES ALGORITHM



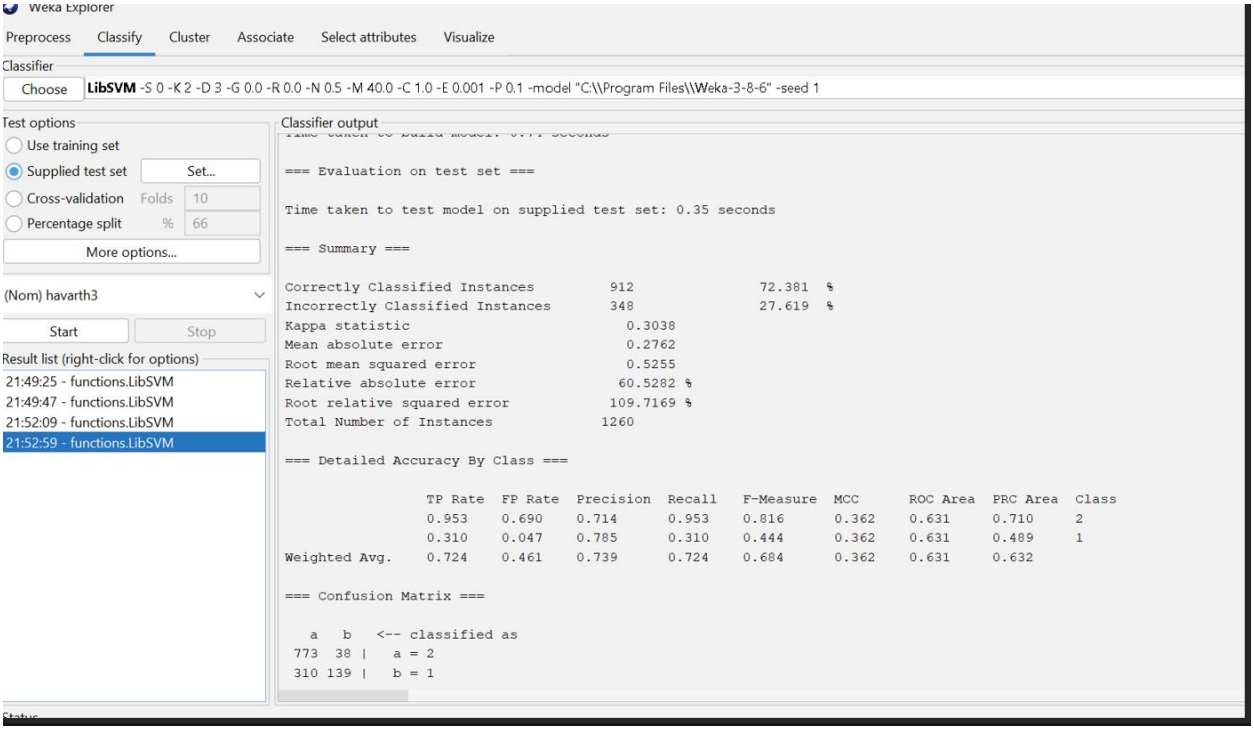

## CORRELATION ATTRIBUTE SELECTION WITH J48 DECISION TREE ALGORITHM.



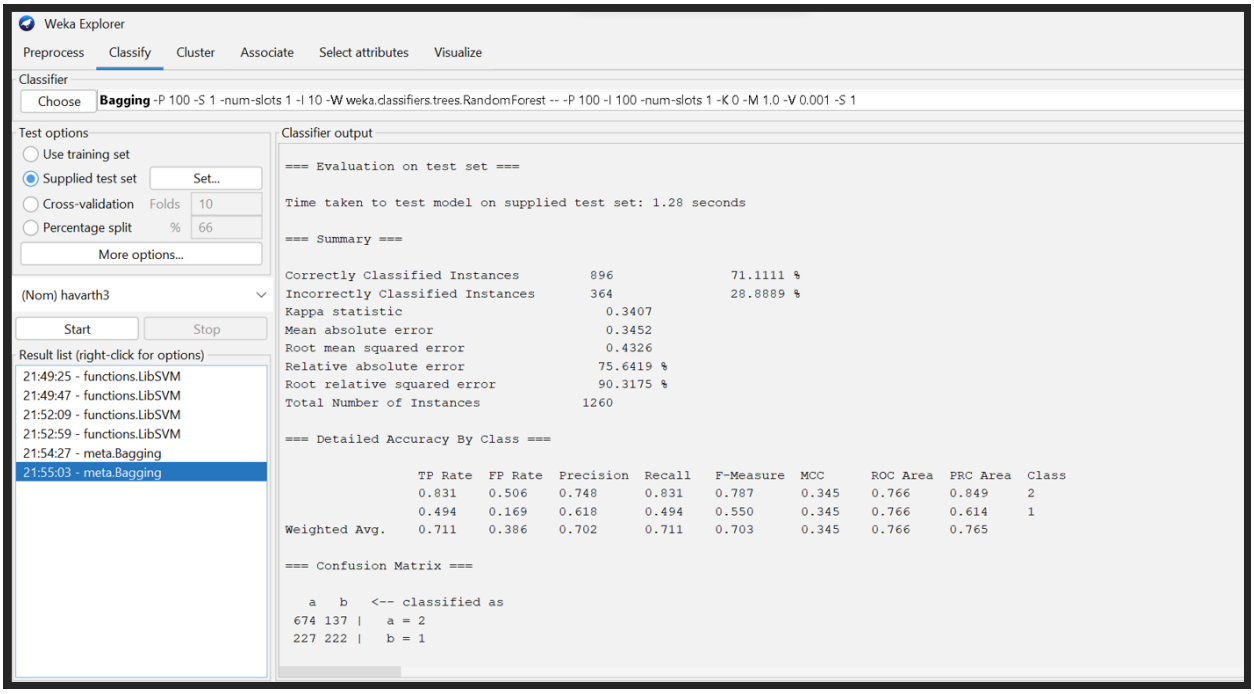

## CORRELATION ATTRIBUTE SELECTION WITH LOGISTIC ALGORITHMS.



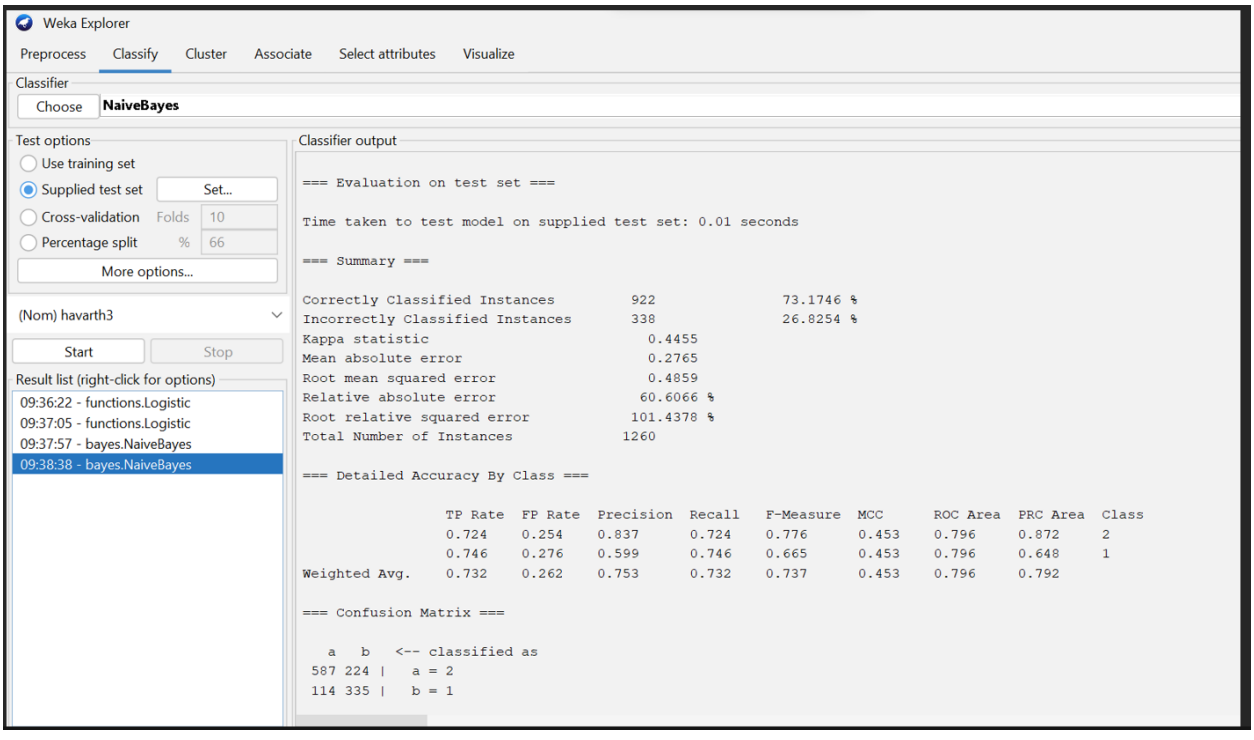

## CORRELATION ATTRIBUTE SELECTION WITH SUPPORT VECTOR MACHINE ALGORITHM.



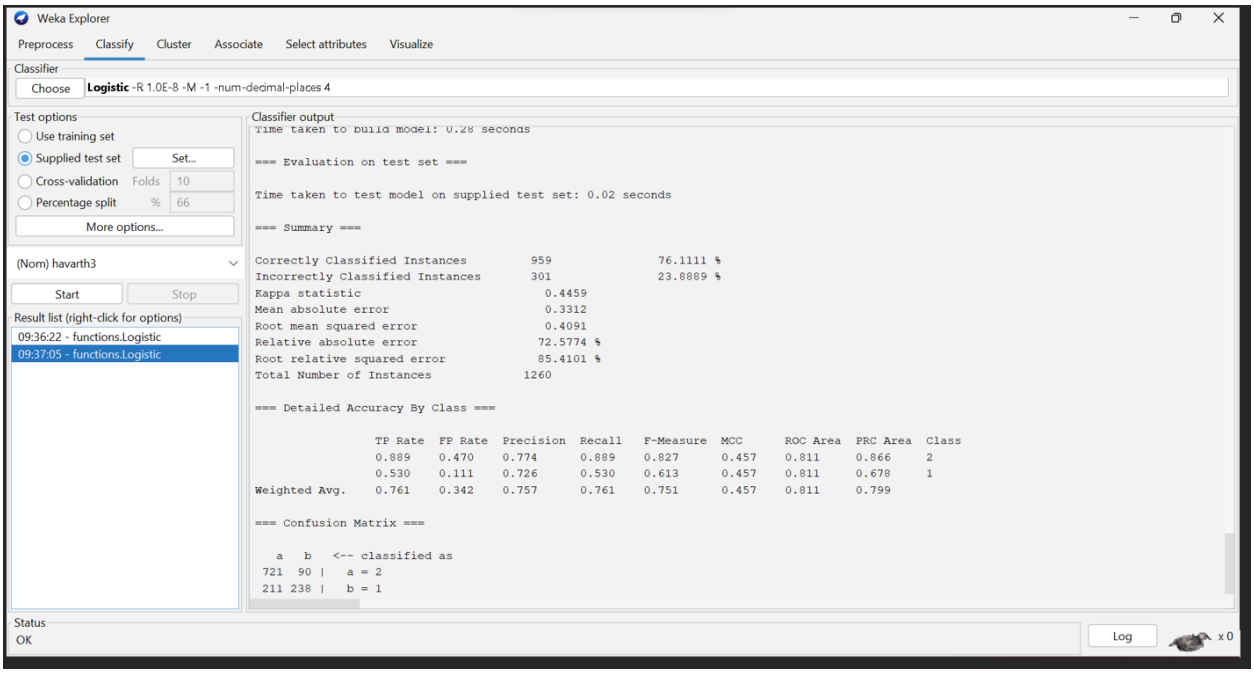

## CORRELATION ATTRIBUTE SELECTION WITH BAGGING (RANDOM FOREST) ALGORITHM.



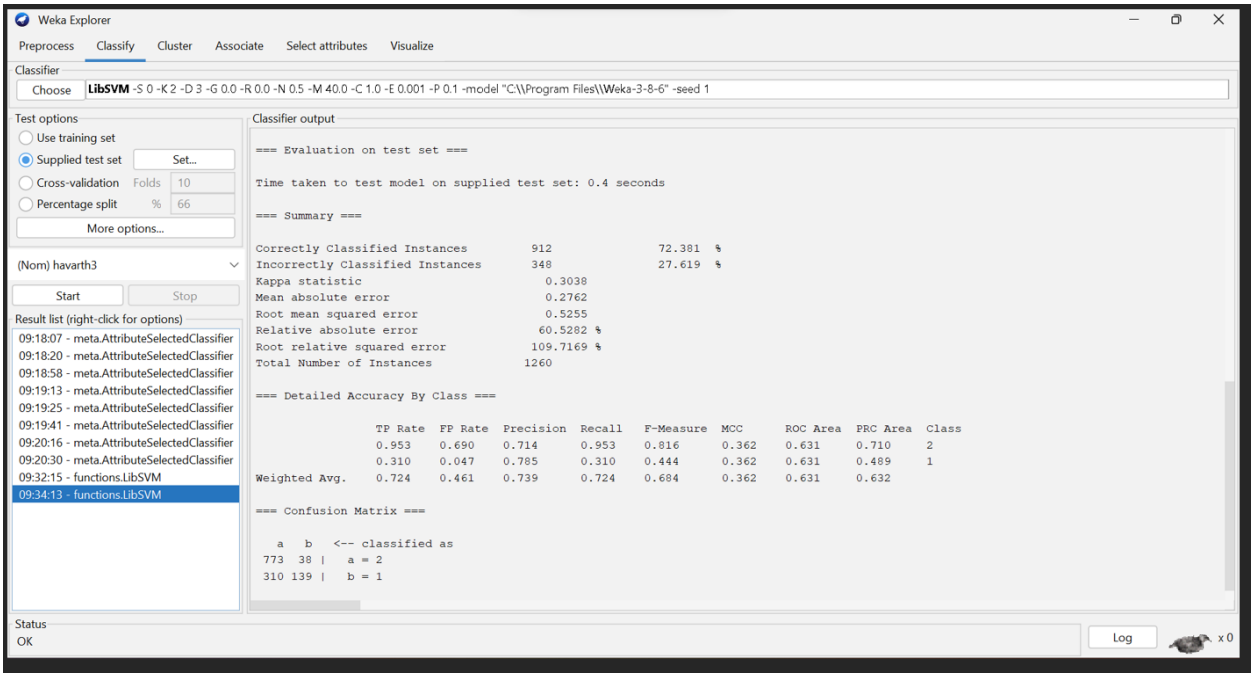

## CHI-SQUARED ATTRIBUTE SELECTION WITH NAIVE BAYES ALGORITHM:



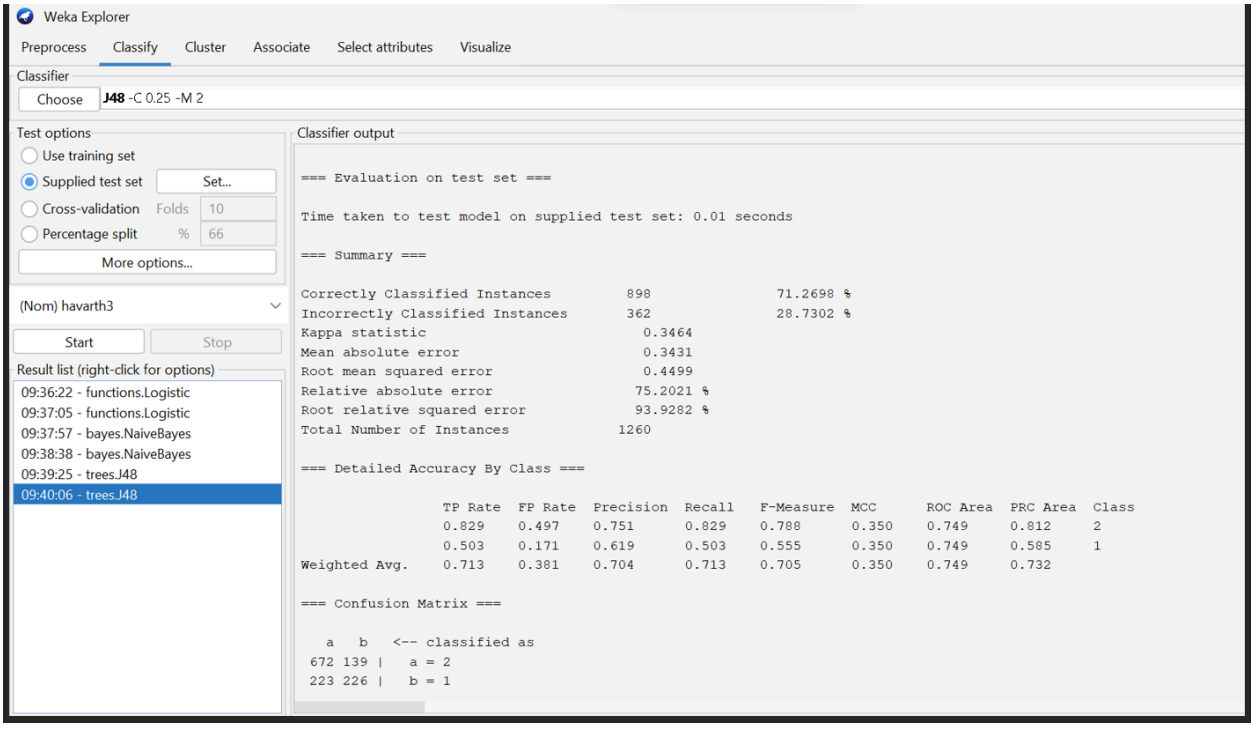

## CHI-SQUARED ATTRIBUTE SELECTION WITH LOGISTIC ALGORITHM


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
Logistic -R 1.0E-8 -M -1 -num-decimal-places 4
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) havarth3
Start
Stop
Result list (right-click for options)
09:36:22 - functions.Logistic
09:37:05 - functions.Logistic
Classifier output
Time taken to build model: 0.28 seconds
=== Evaluation on test set ===
Time taken to test model on supplied test set: 0.02 seconds
=== Summary ===
Correctly Classified Instances 959 76.1111 %
Incorrectly Classified Instances 301 23.8889 %
Kappa statistic 0.4459
Mean absolute error 0.3312
Root mean squared error 0.4091
Relative absolute error 72.5774 %
Root relative squared error 85.4101 %
Total Number of Instances 1260
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
0.889 0.470 0.774 0.889 0.827 0.457 0.811 0.866 2
0.530 0.111 0.726 0.530 0.613 0.457 0.811 0.678 1
Weighted Avg. 0.761 0.342 0.757 0.761 0.751 0.457 0.811 0.799
=== Confusion Matrix ===
a b <-- classified as
721 90 | a = 2
211 238 | b = 1
Status
OK
Log
x 0

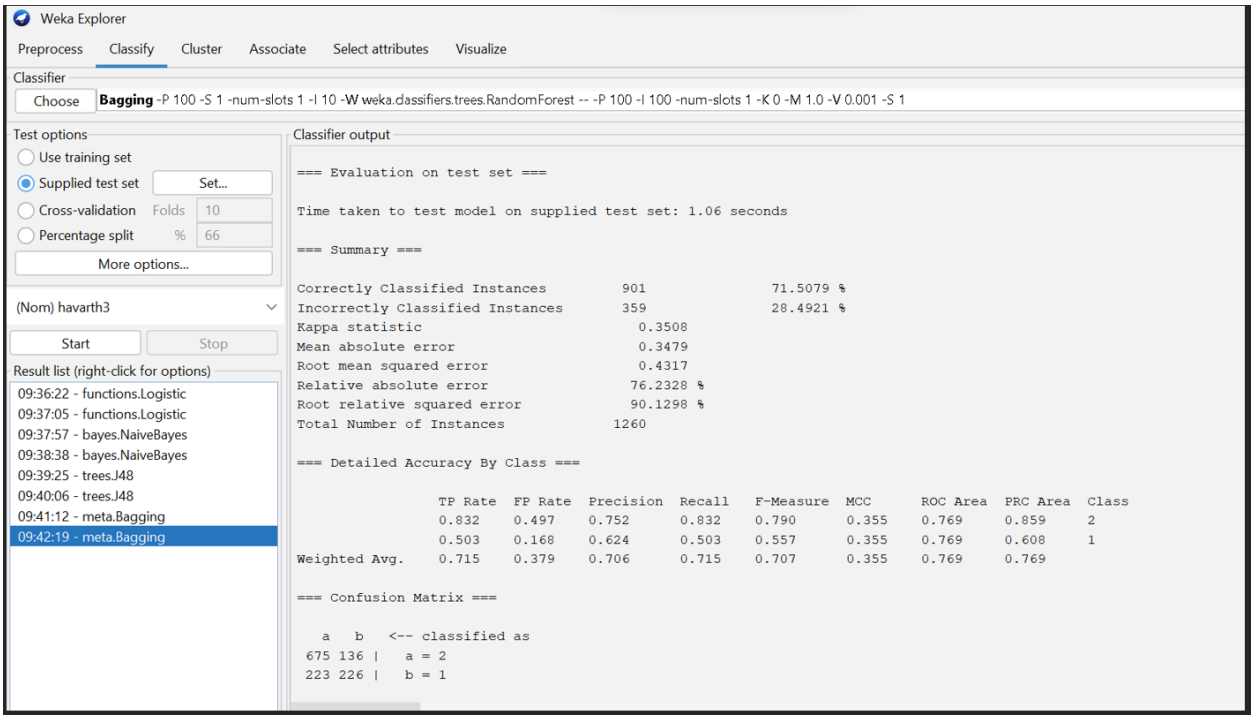
## CHI-SQUARED ATTRIBUTE SELECTION WITH SUPPORT VECTOR MACHINE ALGORITHM



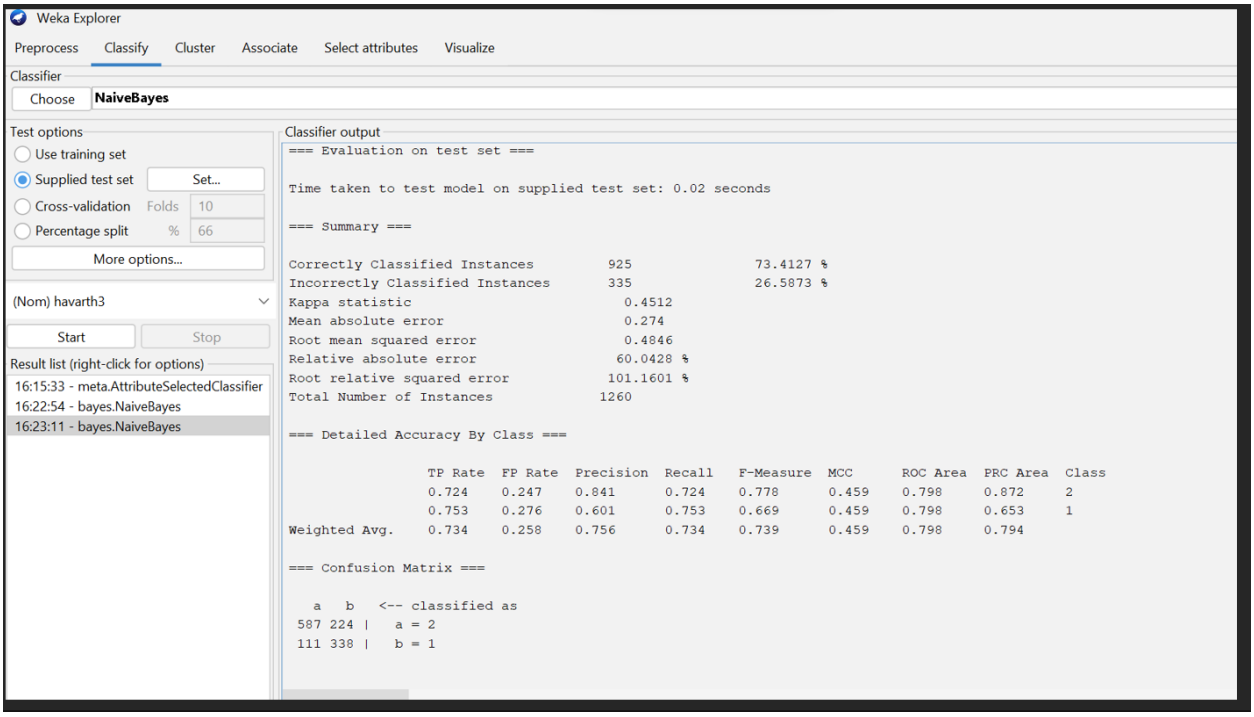

## CHI-SQUARED ATTRIBUTE SELECTION WITH J48 DECISION TREE ALGORITHM



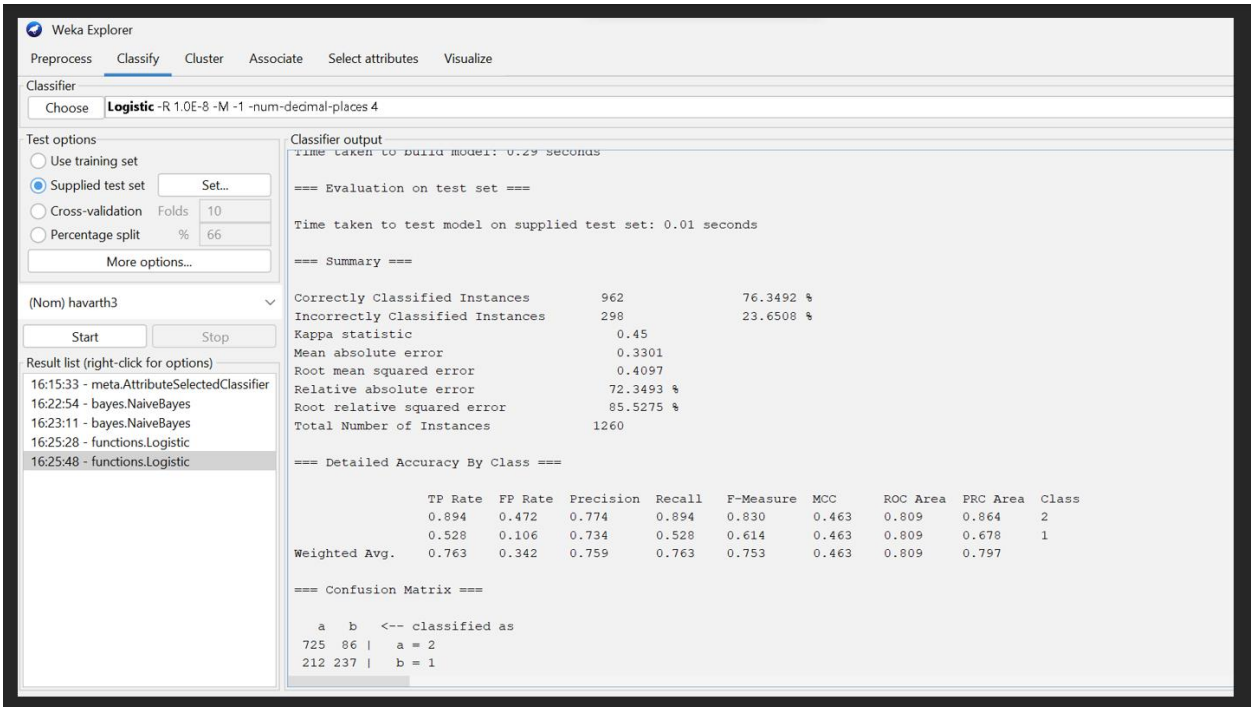
## CHI-SQUARED ATTRIBUTE SELECTION WITH BAGGING (RANDOM FOREST) ALGORITHM



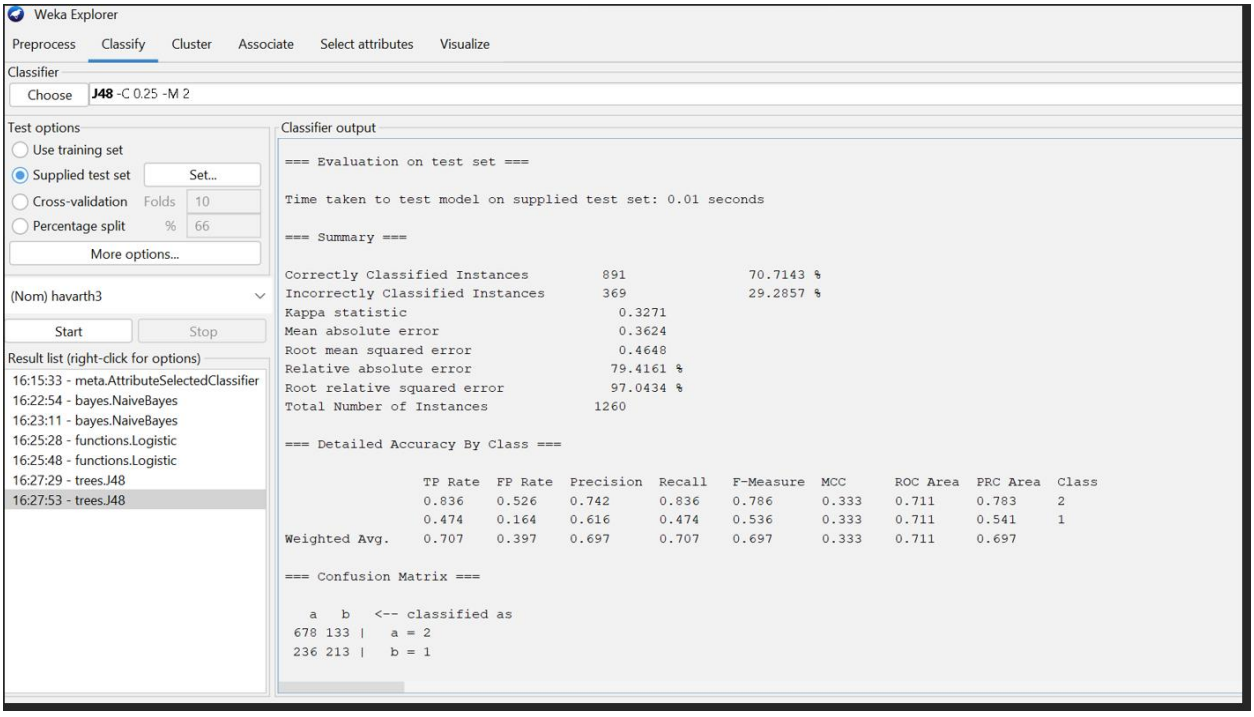

## INFO GAIN ATTRIBUTE SELECTION WITH NAIVE BAYES ALGORITHM.



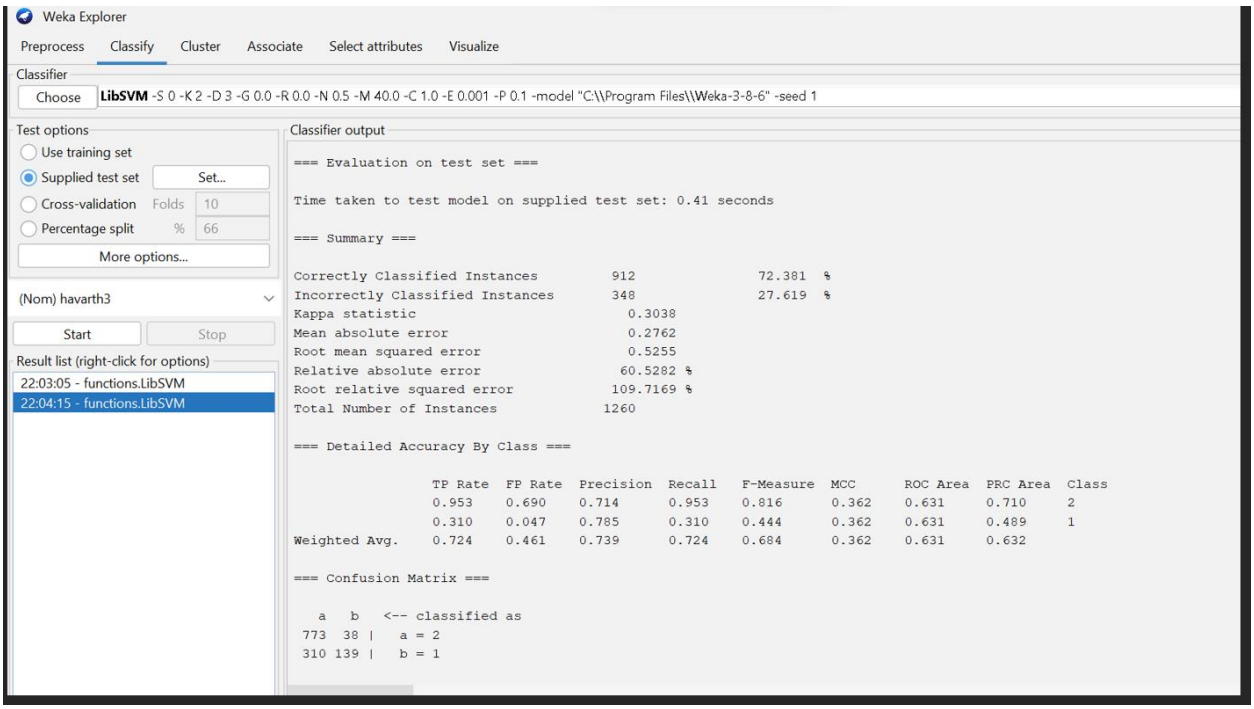
## INFO GAIN ATTRIBUTE SELECTION WITH LOGISTIC ALGORITHM.



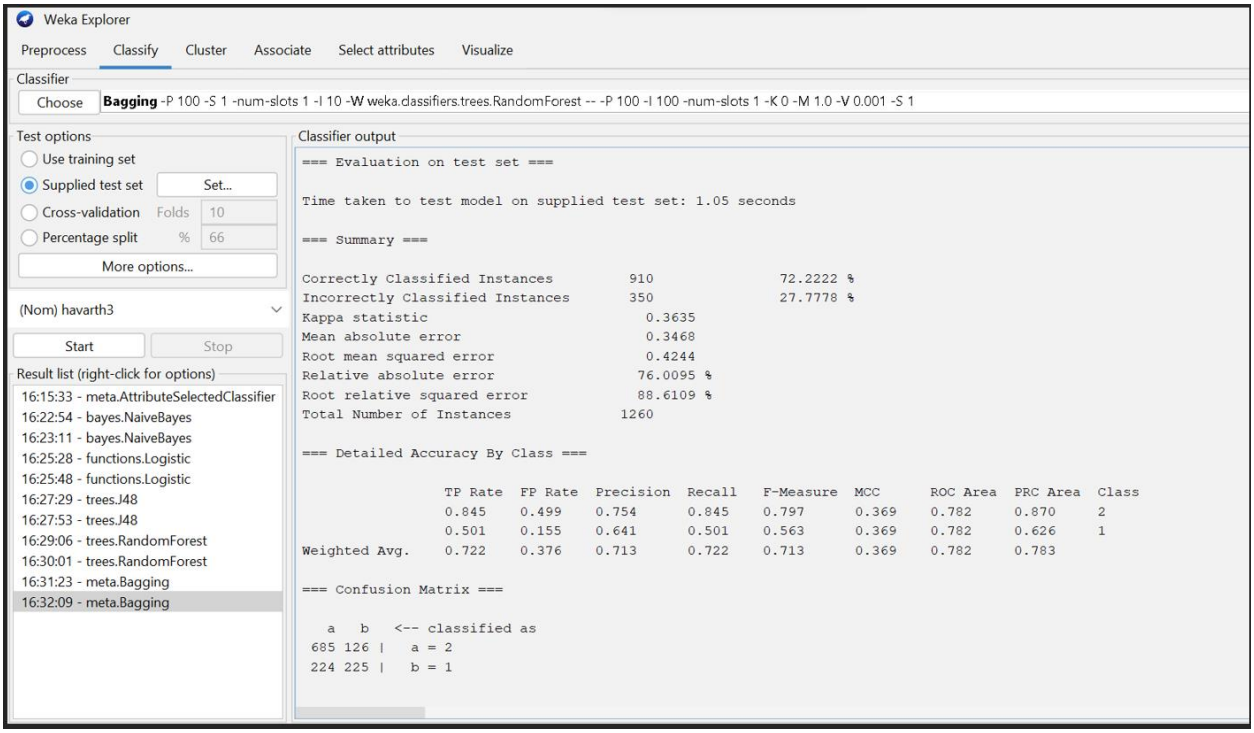

## INFO GAIN ATTRIBUTE SELECTION WITH J48 DECISION TREE ALGORITHM.



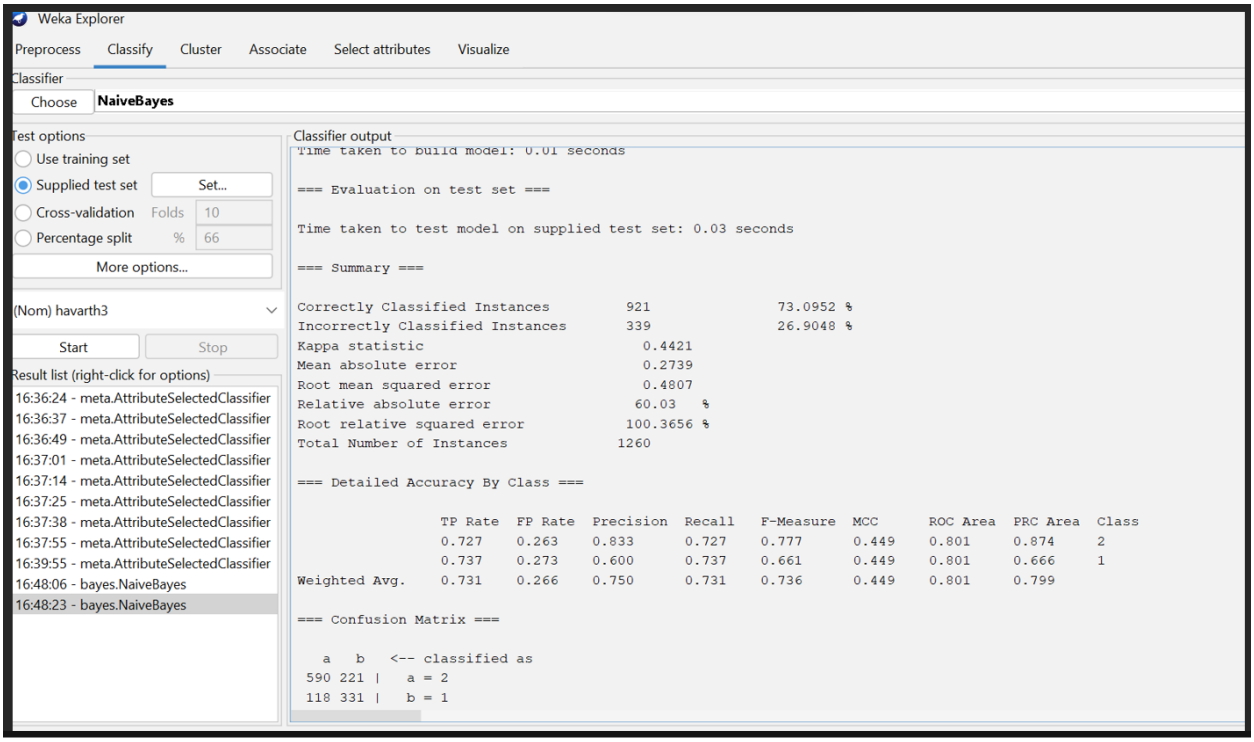
## INFO GAIN ATTRIBUTE SELECTION WITH SUPPORT VECTOR MACHINE ALGORITHM.



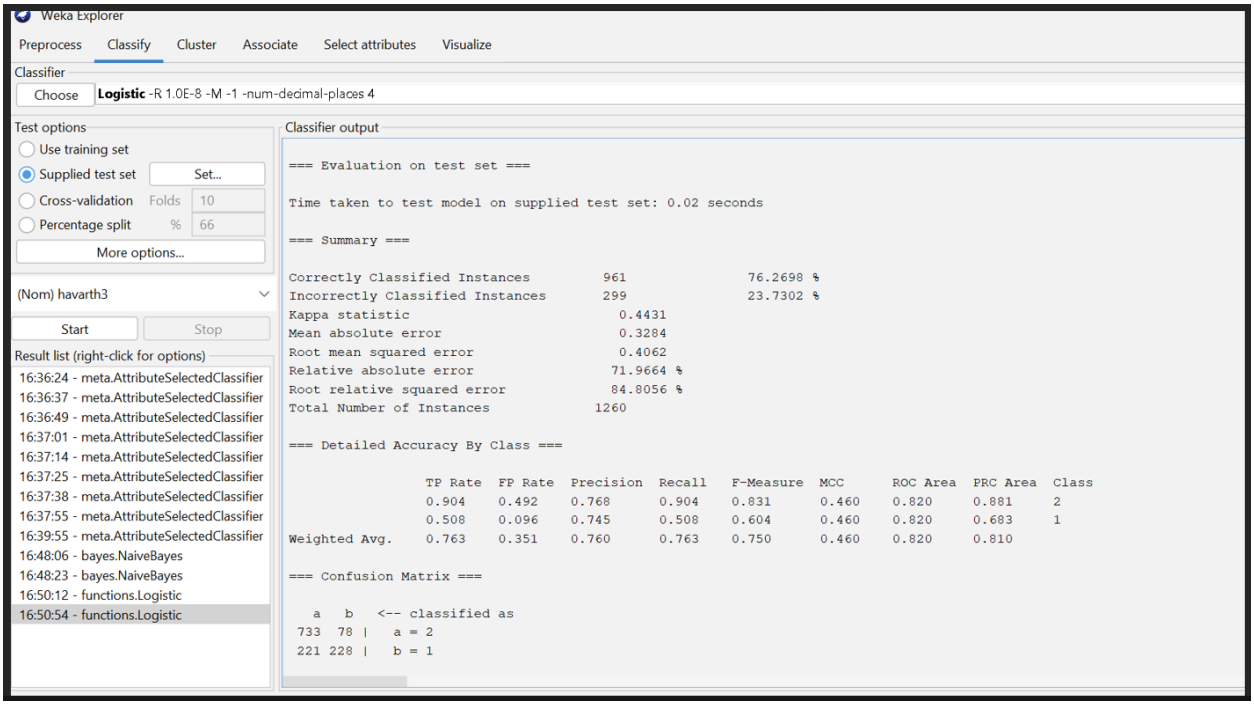

## INFO GAIN ATTRIBUTE SELECTION WITH BAGGING (RANDOM FOREST) ALGORITHM.



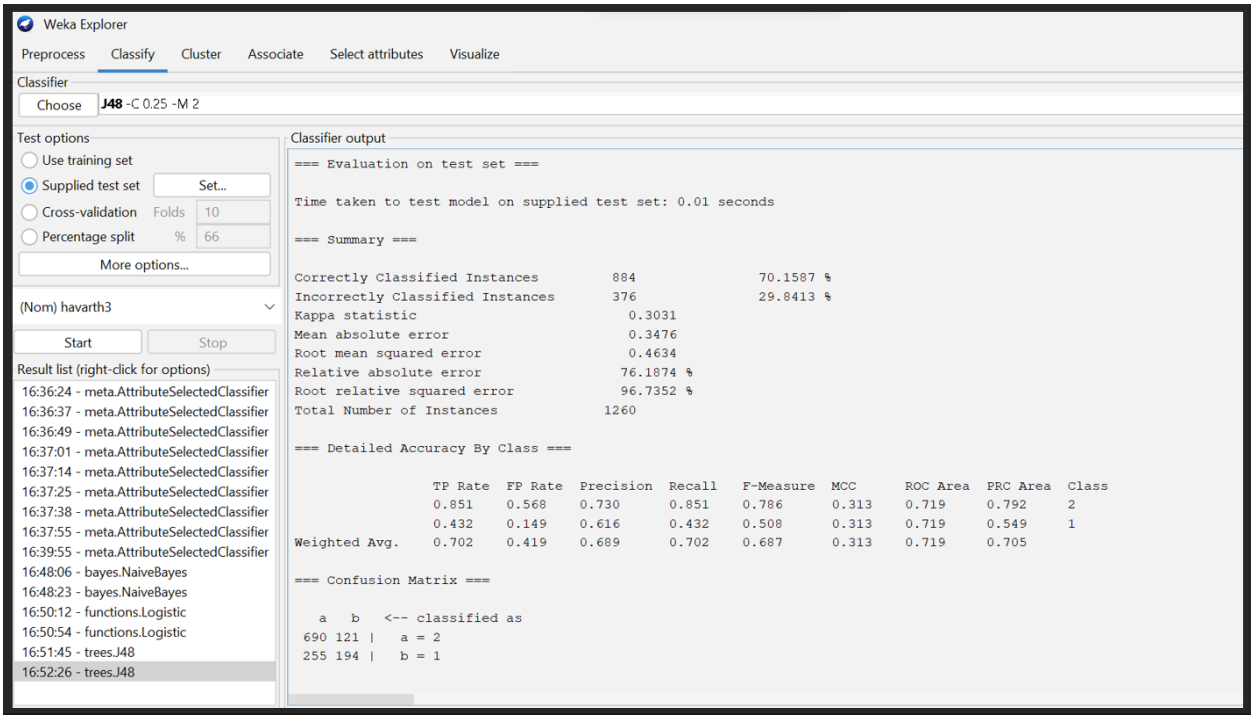
## GAIN RATIO ATTRIBUTE SELECTION WITH NAIVE BAYES ALGORITHM:



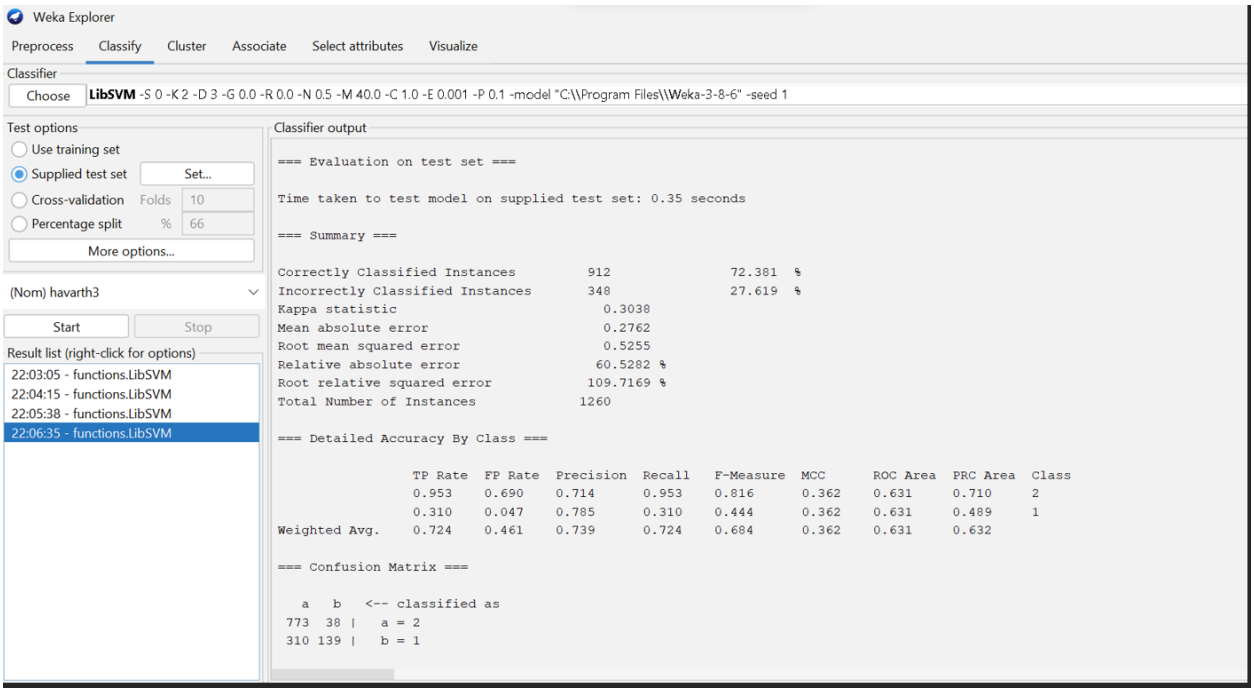

## GAIN RATIO ATTRIBUTE SELECTION WITH LOGISTIC ALGORITHM:



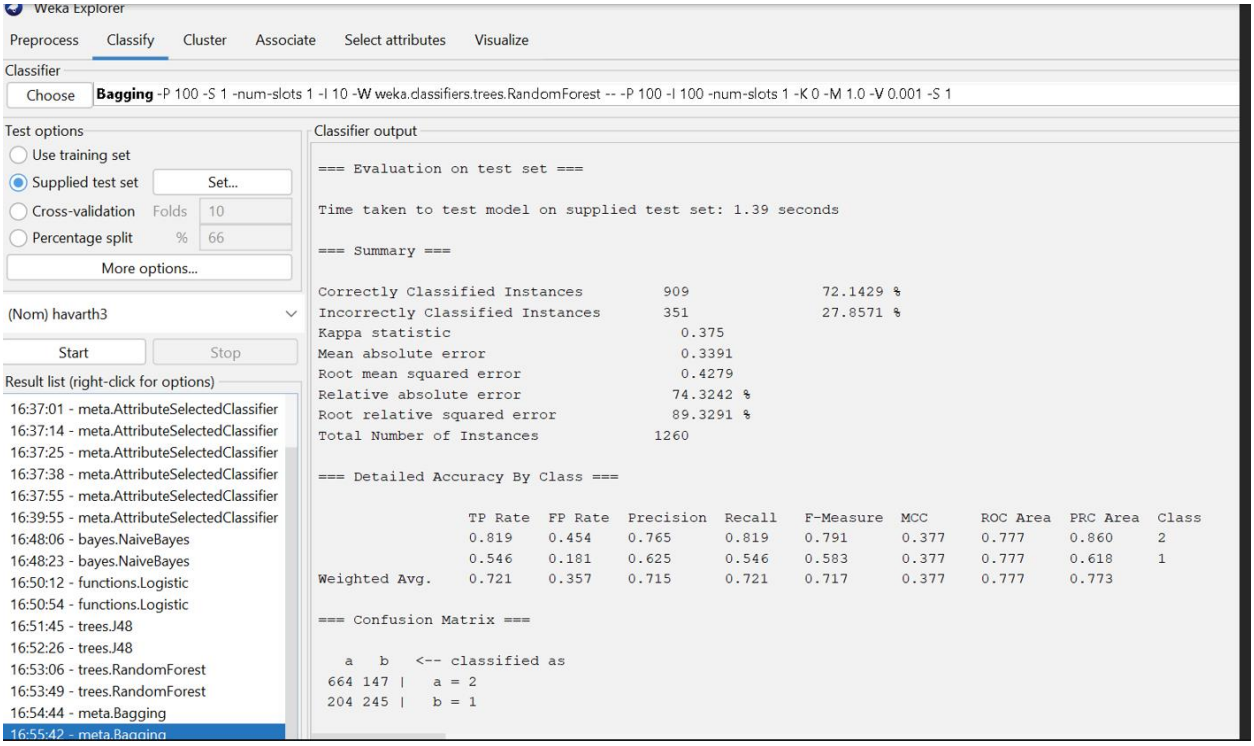
## GAIN RATIO ATTRIBUTE SELECTION WITH J48 DECISION TREE ALGORITHM:



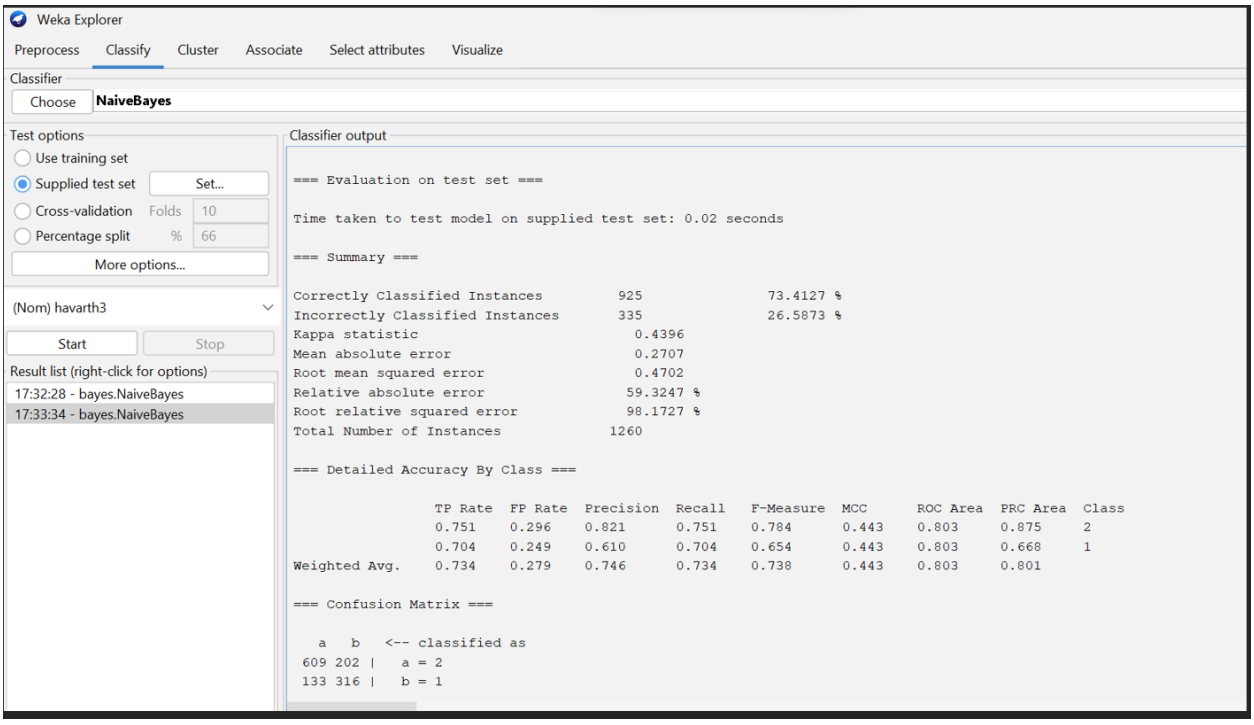

## GAIN RATIO ATTRIBUTE SELECTION WITH SUPPORT VECTOR MACHINE ALGORITHM:



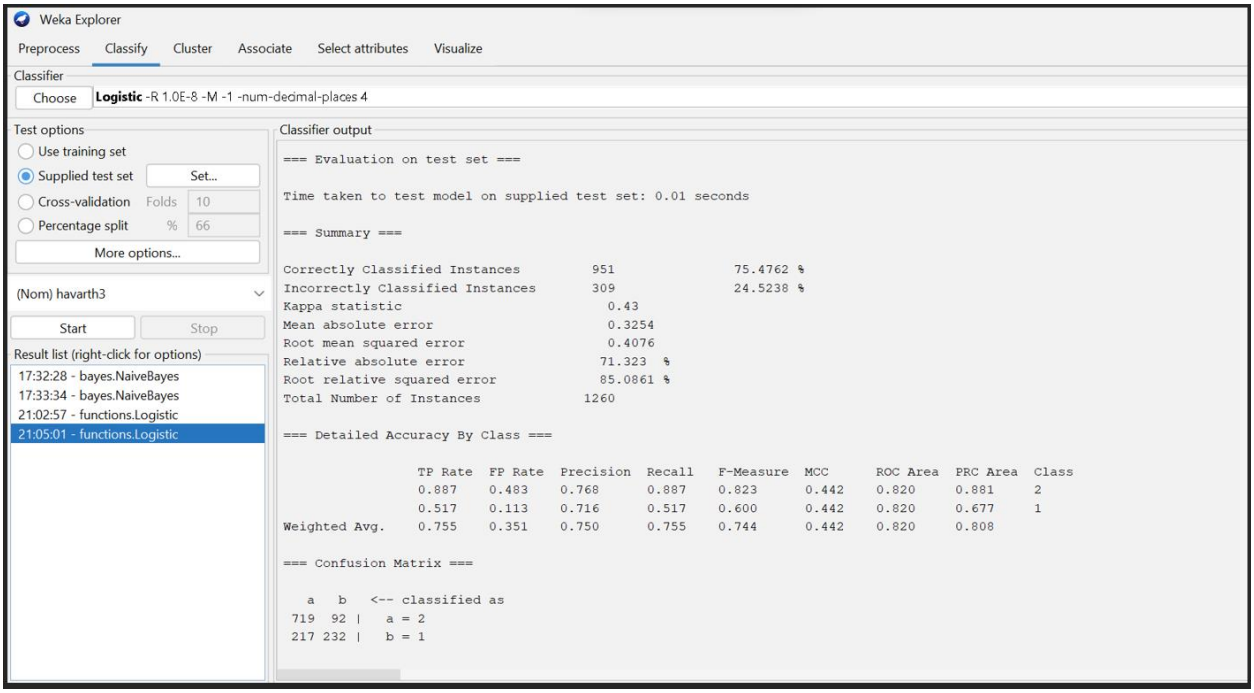
## GAIN RATIO ATTRIBUTE SELECTION WITH BAGGING (RANDOM FOREST) ALGORITHM:



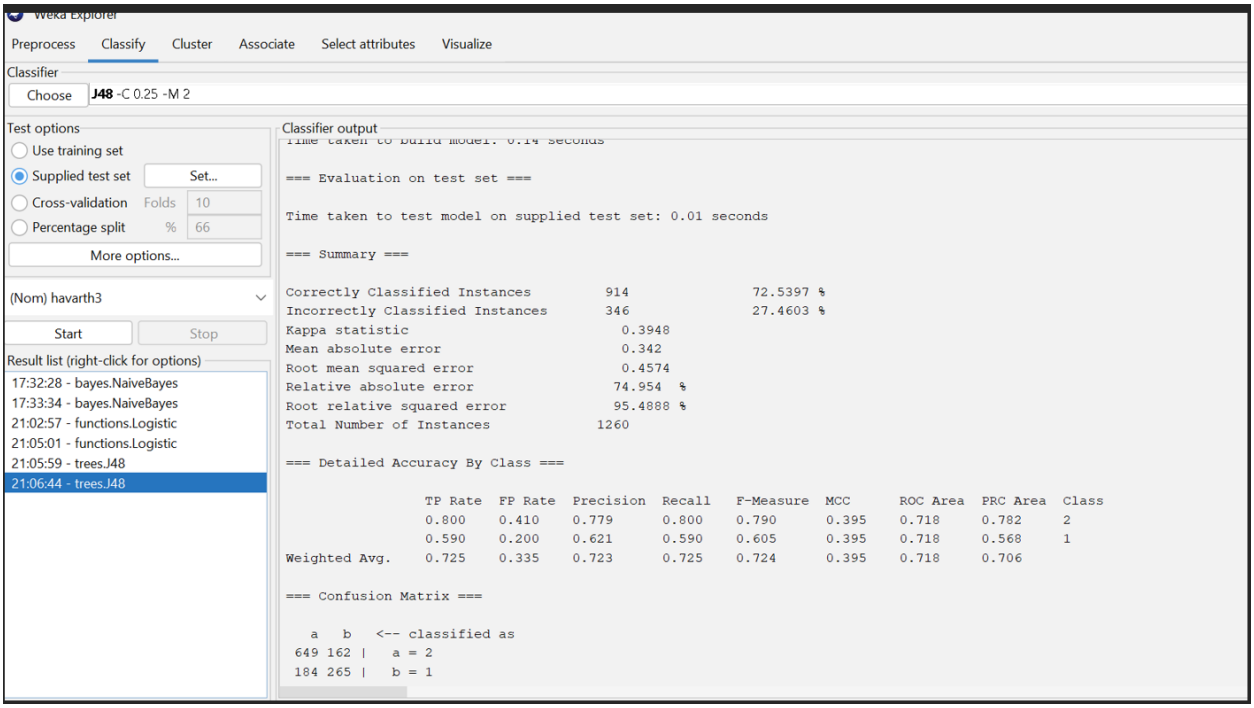

## ONE R ATTRIBUTE SELECTION WITH NAIVE BAYES ALGORITHM.



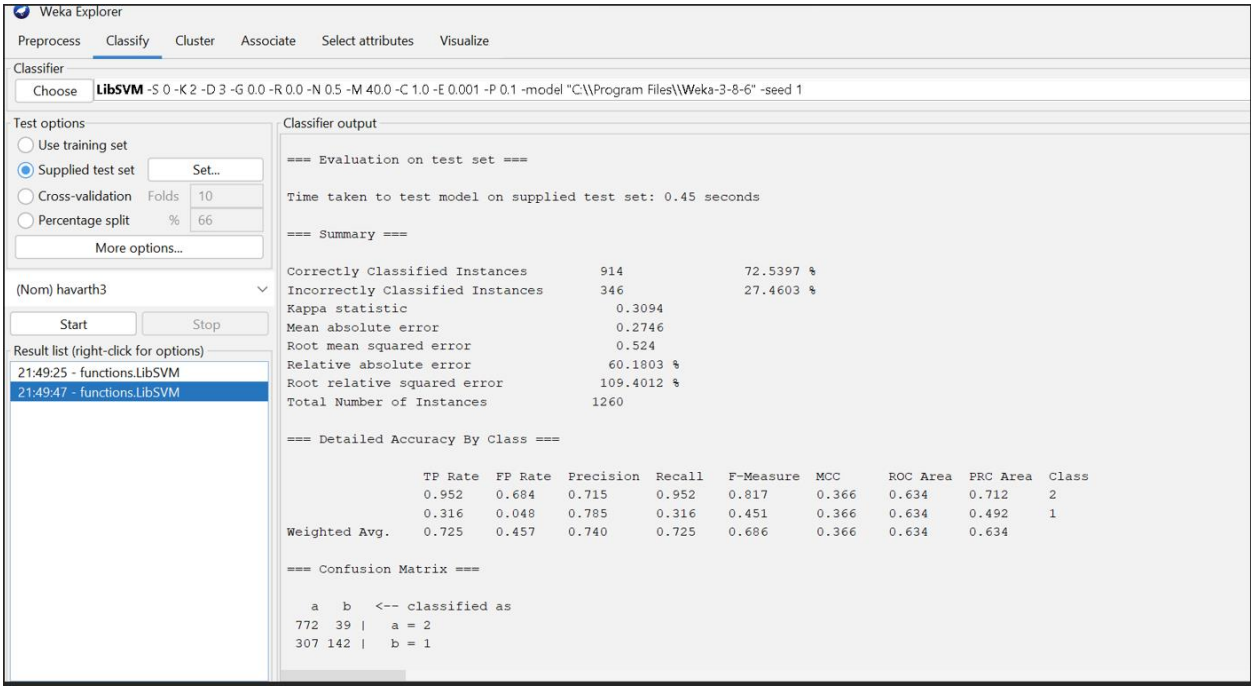
## ONE R ATTRIBUTE SELECTION WITH LOGISTIC ALGORITHM.



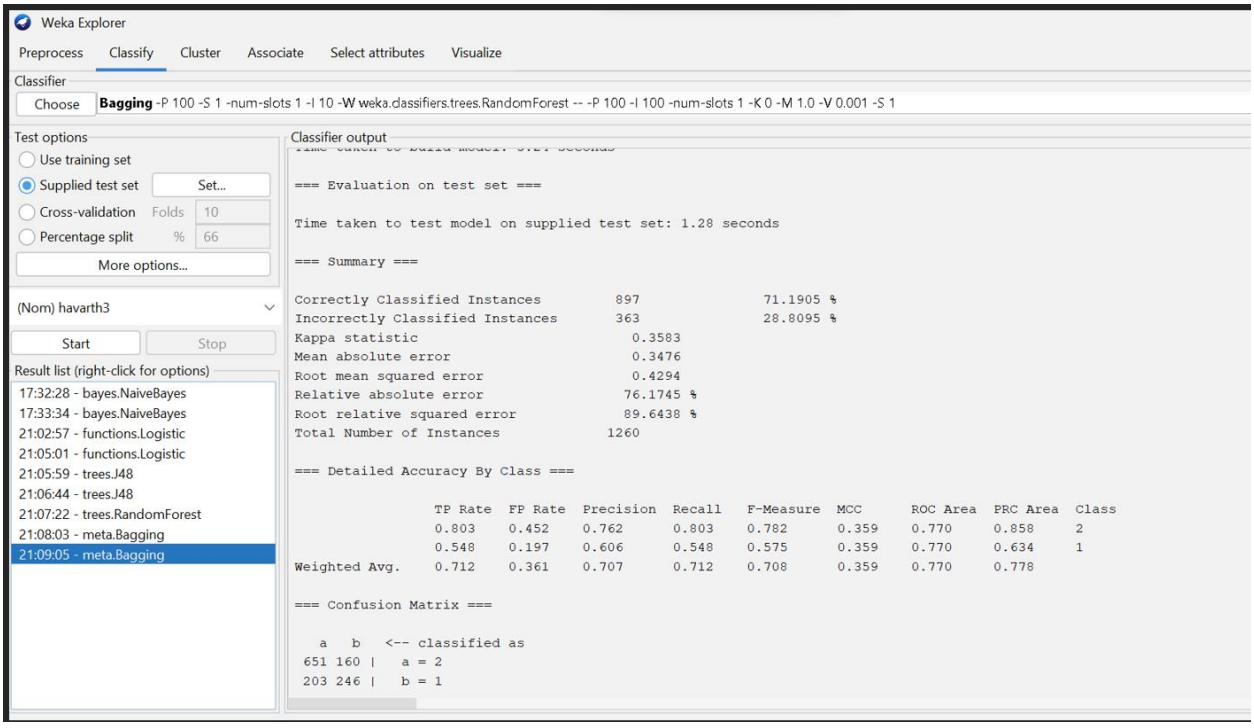

## ONE R ATTRIBUTE SELECTION WITH J48 DECISION TREE ALGORITHM.



## ONE R ATTRIBUTE SELECTION WITH SUPPORT VECTOR MACHINE ALGORITHM.




## ONE R ATTRIBUTE SELECTION WITH BAGGING (RANDOM FOREST) ALGORITHM.



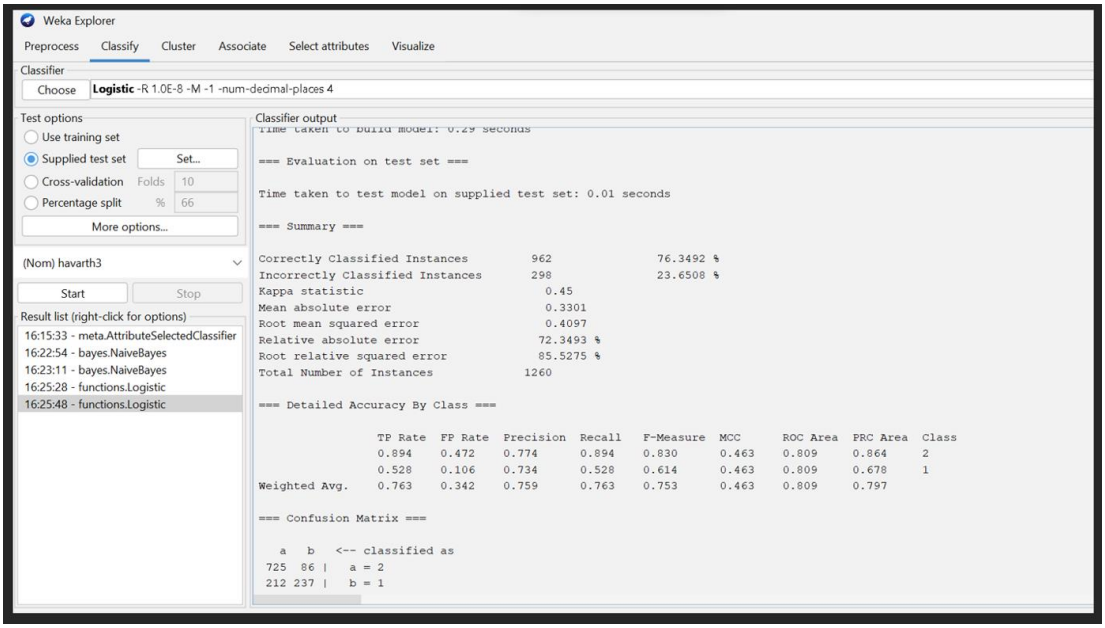
| **Attribute selection algorithm** | **Classification algorithm** | **Accuracy** | **ROC** |
|---|---|---|---|
| Correlation attribute selection | Naive Bayes | 73.0159 | 0.791 |
| | Logistic | 76.0317 | 0.798 |
| | J48 Decision Tree | 71.9048 | 0.737 |
| | SVM | 72.381 | 0.631 |
| | Bagging (Random Forest) | 71.1111 | 0.766 |
| | **Classification algorithm** | **Accuracy** | **ROC** |
| Chi Squared Attribute selection | Naive Bayes | 73.1746 | 0.796 |
| | Logistic | 76.111 | 0.811 |
| | J48 Decision Tree | 72.381 | 0.631 |
| | SVM | 71.2698 | 0.749 |
| | Bagging (Random Forest) | 71.5079 | 0.769 |
| | **Classification algorithm** | **Accuracy** | **ROC** |
| Info Gain attribute selection | Naive Bayes | 73.4127 | 0.798 |
| | Logistic | 76.3492 | 0.809 |
| | J48 Decision Tree | 70.7143 | 0.711 |
| | SVM | 72.381 | 0.631 |
| | Bagging (Random Forest) | 72.222 | 0.782 |
| | **Classification algorithm** | **Accuracy** | **ROC** |
| Gain Ratio Attribute selection | Naive Bayes | 73.0952 | 0.801 |
| | Logistic | 76.2698 | 0.820 |
| | J48 Decision Tree | 70.1587 | 0.719 |
| | SVM | 72.381 | 0.631 |
| | Bagging (Random Forest) | 72.1429 | 0.777 |
| | **Classification algorithm** | **Accuracy** | **ROC** |
| One R attribute selection | Naive Bayes | 73.4127 | 0.803 |
| | Logistic | 75.4762 | 0.820 |
| | J48 Decision Tree | 72.5397 | 0.718 |
| | SVM | 72.5397 | 0.634 |
| | Bagging (Random Forest) | 71.1905 | 0.770 |

# CONCLUSION

Info Gain attribute selection method with Logistic regression algorithm gives me the best results. Logistic regression gave me the best results for all attribute selection methods, but it achieved the highest accuracy when used with Info Gain. For getting this model, I used ranker = 23. This model used the following attributes:

```
 64 x.age80
 62 x.age.g
 66 x.ageg5yr
 22 diffwalk
 34 genhlth
  2 employ1
 31 physhlth
 95 x.phys14d
 87 x.rfhlth
 97 x.hcvu651
 67 x.age65yr
 20 pneuvac4
 27 rmvteth4
102 x.exteth3
 11 marital
 71 x.incomg
 69 x.chldcnt
  3 income2
  6 children
 43 checkup1
 25 diffalon
 46 chccopd1
 77 drocdy3.
```

And achieved the following performance on the test data:




## FIVE ATTRIBUTES THAT ARE MOST RELEVANT TO THE CLASS ATTRIBUTES:

- x.age80 (Age > 80)
- Diffwalk (Difficulty in walking)
- Genhlth (General Health)
- Chckdny (kidney diseases)
- chccopd1 (chronic obstructive pulmonary disease, emphysema, or chronic bronchitis)

I selected these attributes because all five attributes selection methods suggest them. In general, we also know that age, heart disease, kidney disease, health in general, and difficulty in walking are connected to arthritis.

## WHAT I LEARNED FROM THIS PROJECT:

From this project, I got hands-on experience in data mining. I learned how to:

- Handle missing values
- Remove duplicate values
- Standardize data
- Handle outliers
- Split the dataset for training and testing
- Select features
- Implement different classification algorithms
- Compare models