SUMMER 2021 | CS 699 – PROJECT REPORT

Customer Churn Analysis

Abstract

The main contribution of our work is to develop a churn prediction model which assists telecom operators to predict customers who are most likely subject to churn.

Parvathy Sukumaran
Sahil Khanna

Table of Contents

Introduction

Customer churn is a significant problem and one of the most critical concerns for large companies. Due to the direct effect on the companies' revenues, especially in the telecom field, companies seek to develop means to predict potential customers to churn. Therefore, finding factors that increase customer churn is vital to take necessary actions to reduce this churn.

Data Mining Goal

We aim to accomplish the following for this study:

1. First, identify and visualize which factors contribute to customer churn.
2. Build a prediction model that will perform the following:
    a. Classify if a customer is going to churn or not.
    b. Test using multiple models and based on model performance, choose a model that will attach a probability to the churn.

Dataset Details

Dataset Source: IBM

Link to dataset:
https://community.ibm.com/accelerators/catalog/content/Telco-customer-churn

The dataset is from a fictional telecommunications company that includes **33** features about **7000** users such as:

1. Customers who left within the last month –Churn Label, Churn Value, etc.
2. Services that each customer has signed up for – phone, multiple lines, internet, online security, online backup, device protection, tech support, and streaming TV and movies.
3. Customer account information – how long they've been a customer, contract, payment method, paperless billing, monthly charges, and total charges
4. Demographic info about customers – gender, age range, location, and if they have partners and dependents.

Number of attributes = 33
Number of tuples = 7043

1. ***CustomerID:*** A unique ID that identifies each customer.
2. ***Count:*** A value used in reporting/dashboarding to sum up, the number of customers in a filtered set.
3. ***Country:*** The country of the customer's primary residence.
4. ***State:*** The state of the customer's primary residence.

5. ***City:*** The city of the customer's primary residence.
6. ***Zip Code***: The zip code of the customer's primary residence.
7. ***Lat Long:*** The combined latitude and longitude of the customer's primary residence.
8. ***Latitude:*** The latitude of the customer's primary residence.
9. ***Longitude:*** The longitude of the customer's primary residence.
10. ***Gender:*** The customer's gender: Male, Female
11. ***Senior Citizen:*** Indicates if the customer is 65 or older: Yes, No
12. ***Partner:*** Indicate if the customer has a partner: Yes, No
13. ***Dependents:*** Indicates if the customer lives with any dependents: Yes, No. Dependents could be children, parents, grandparents, etc.
14. ***Tenure Months:*** Indicates the total amount of months that the customer has been with the company by the end of the quarter specified above.
15. ***Phone Service:*** Indicates if the customer subscribes to home phone service with the company: Yes, No
16. ***Multiple Lines:*** Indicates if the customer subscribes to multiple telephone lines with the company: Yes, No
17. ***Internet Service:*** Indicates if the customer subscribes to Internet service with the company: No, DSL, Fiber Optic, Cable.
18. ***Online Security:*** Indicates if the customer subscribes to an additional online security service provided by the company: Yes, No
19. ***Online Backup:*** Indicates if the customer subscribes to an additional online backup service provided by the company: Yes, No
20. ***Device Protection:*** Indicates if the customer subscribes to an additional device protection plan for their Internet equipment provided by the company: Yes, No
21. ***Tech Support:*** Indicates if the customer subscribes to an additional technical support plan from the company with reduced wait times: Yes, No
22. ***Streaming TV:*** Indicates if the customer uses their Internet service to stream television programming from a third-party provider: Yes, No. The company does not charge an additional fee for this service.
23. ***Streaming Movies:*** Indicates if the customer uses their Internet service to stream movies from a third-party provider: Yes, No. The company does not charge an additional fee for this service.
24. ***Contract:*** Indicates the customer's current contract type: Month-to-Month, One Year, Two Year.
25. ***Paperless Billing:*** Indicates if the customer has chosen paperless billing: Yes, No
26. ***Payment Method:*** Indicates how the customer pays their bill: Bank Withdrawal, Credit Card, Mailed Check
27. ***Monthly Charge:*** Indicates the customer's current total monthly charge for all their services from the company.
28. ***Total Charges:*** Indicates the customer's total charges, calculated to the end of the quarter specified above.
29. ***Churn Label:*** **Yes** = the **customer left** the company this quarter. **No** = the **customer remained** with the company. They were directly related to Churn Value.

30. ***Churn Value:*** **1** = the **customer left** the company this quarter. **0** = the **customer remained** with the company. They were directly related to Churn Label.
31. ***Churn Score:*** A value from 0-100 calculated using the predictive tool IBM SPSS Modeler. The model incorporates multiple factors known to cause churn. The higher the score, the more likely the customer will churn.
32. ***CLTV:*** Customer Lifetime Value. A predicted CLTV is calculated using corporate formulas and existing data. The higher the value, the more valuable the customer. High-value customers should be monitored for churn.
33. ***Churn Reason:*** A customer's specific reason for leaving the company. It is directly related to Churn Category.

Data Mining Tools

We used **Python** for Exploratory Data Analysis (EDA) and Data preprocessing.
We used ***Weka*** for the data mining tasks, including attribute selection and classification.

Data Pre-processing and Exploratory Data Analysis

Data Pre-processing

We used *Python in the Jupyter* environment for Pre-processing and EDA and performed data (dimensionality) reduction. We selected the most suitable attributes and dropped others. Thus, reducing our dataset to 23 attributes.

```
In [3]: df.drop(['CustomerID', 'Count', 'Country', 'State','Lat Long', 'Latitude', 'Longitude',
               'Churn Score', 'CLTV'],axis=1,inplace = True)
        df.info()
```

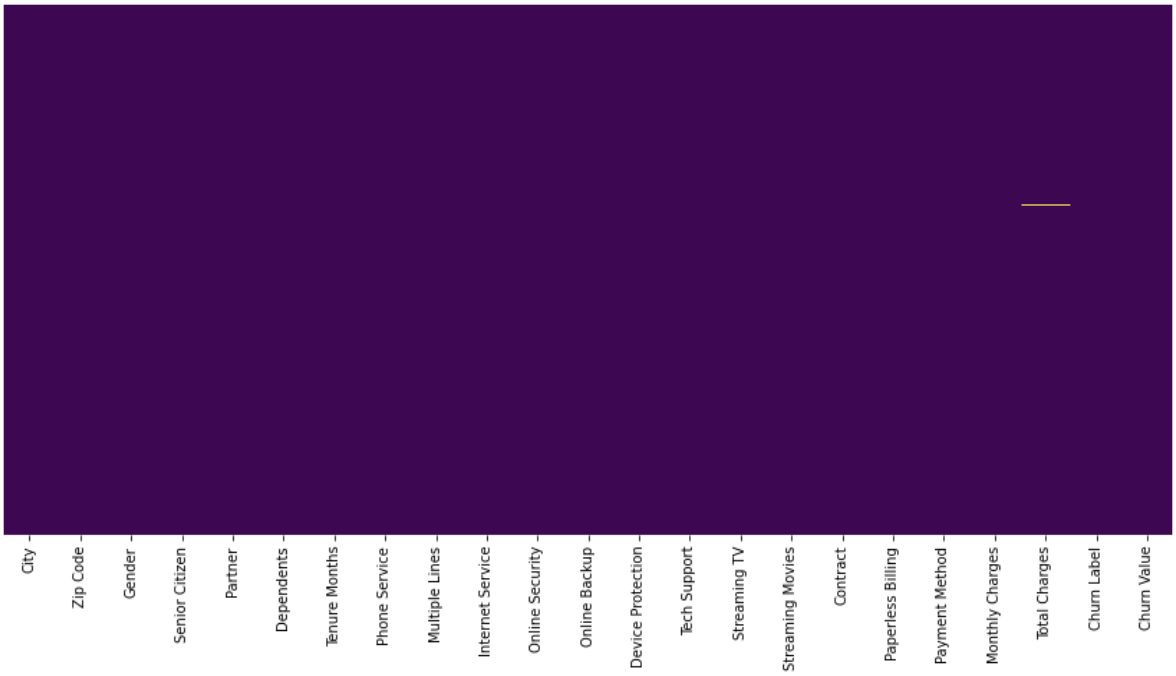
Exploratory Data Analysis

Before we start our analysis, we'll need to convert categorical features to dummy variables using pandas. Otherwise, our data analysis won't be able to take in those features as inputs directly.
Note: However, in weka, we will use the categorical features as is.

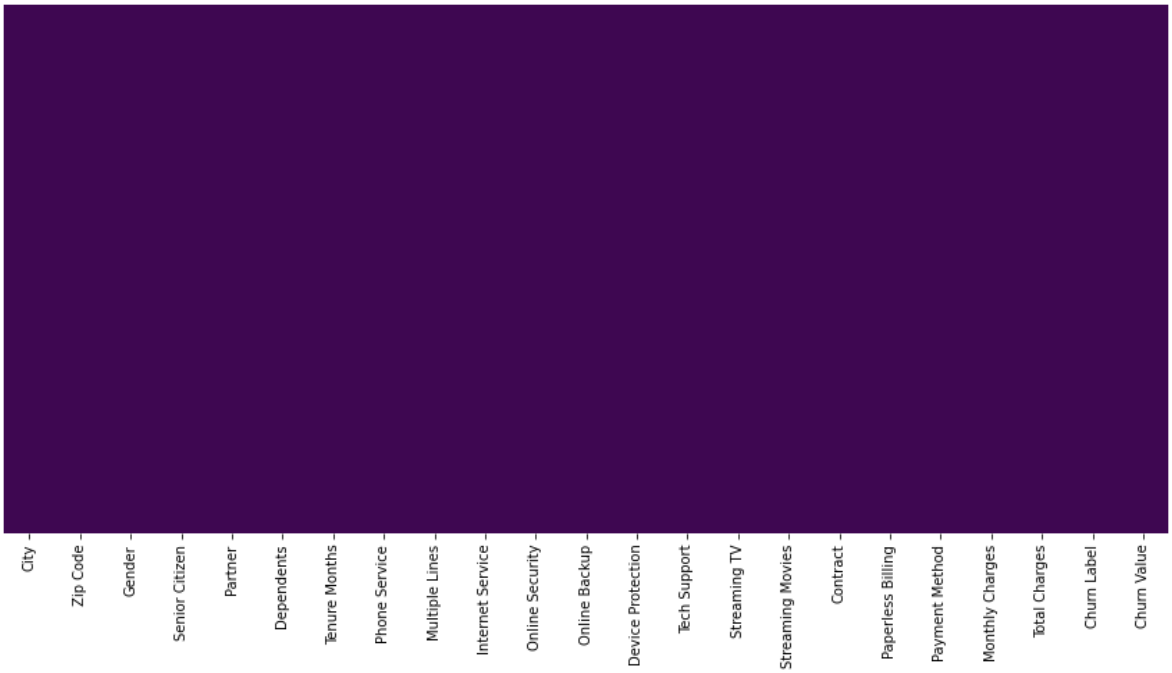
```
In [4]: df["Senior Citizen"] = np.where(df["Senior Citizen"] == "Yes",1,0)
        df["Partner"] = np.where(df["Partner"] == "Yes",1,0)
        df["Dependents"] = np.where(df["Dependents"] == "Yes",1,0)
        df["Phone Service"] = np.where(df["Phone Service"] == "Yes",1,0)
        df["Multiple Lines"] = np.where(df["Multiple Lines"] == "Yes",1,0)
        df["Online Security"] = np.where(df["Online Security"] == "Yes",1,0)
        df["Online Backup"] = np.where(df["Online Backup"] == "Yes",1,0)
        df["Device Protection"] = np.where(df["Device Protection"] == "Yes",1,0)
        df["Tech Support"] = np.where(df["Tech Support"]  == "Yes",1,0)
        df["Streaming TV"] = np.where(df["Streaming TV"] == "Yes",1,0)
        df["Streaming Movies"] = np.where(df["Streaming Movies"] == "Yes",1,0)
        df["Paperless Billing"] = np.where(df["Paperless Billing"] == "Yes",1,0)
        df["Total Charges"] = pd.to_numeric(df['Total Charges'],errors='coerce')
```

Missing Data

We used the seaborn package to create a simple heatmap to look for the missing data.

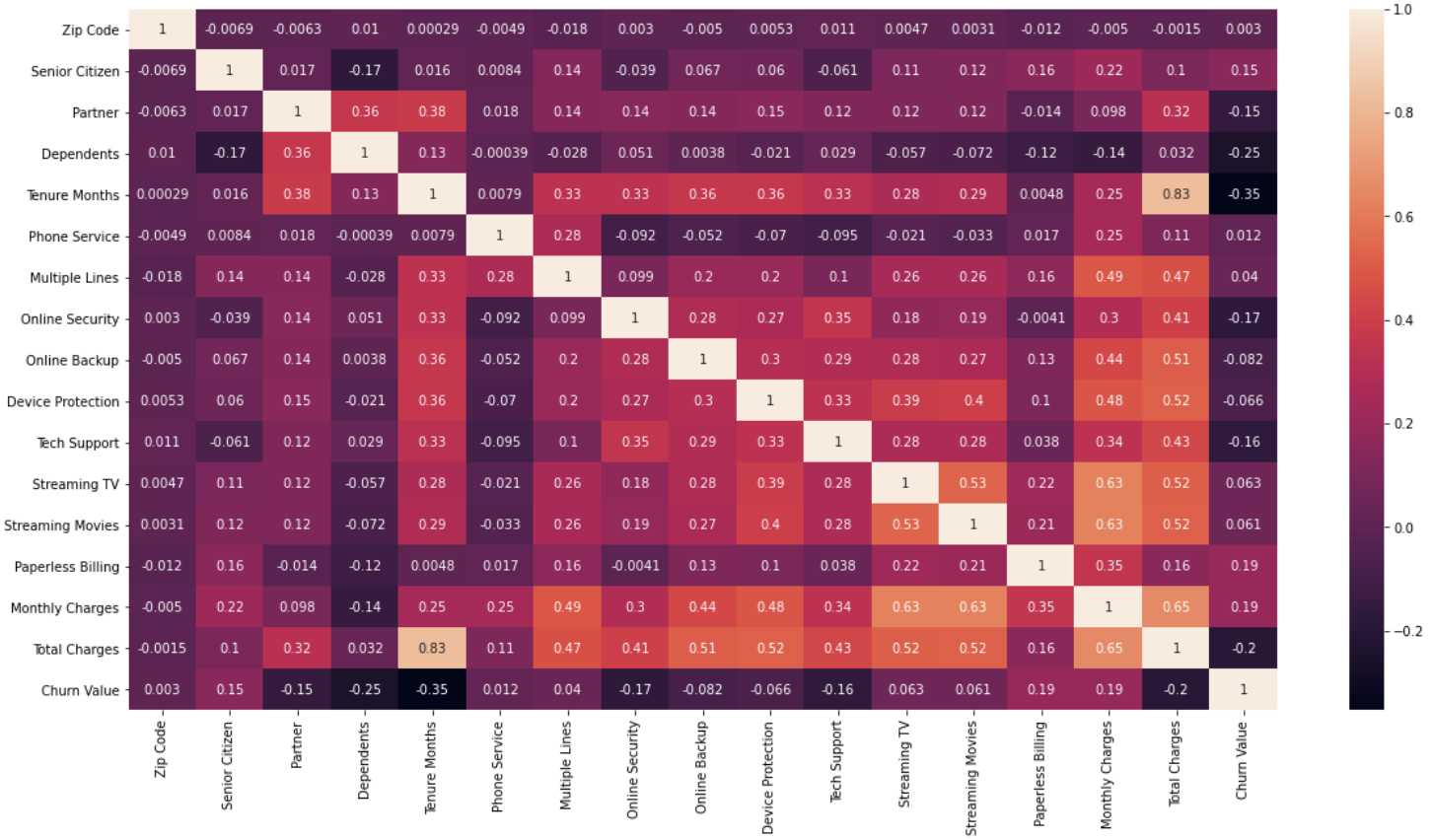


Dataset after removing one row of missing data



Correlation

Using seaborn, we can create a correlation heatmap between data columns.



Summary of Numerical attributes

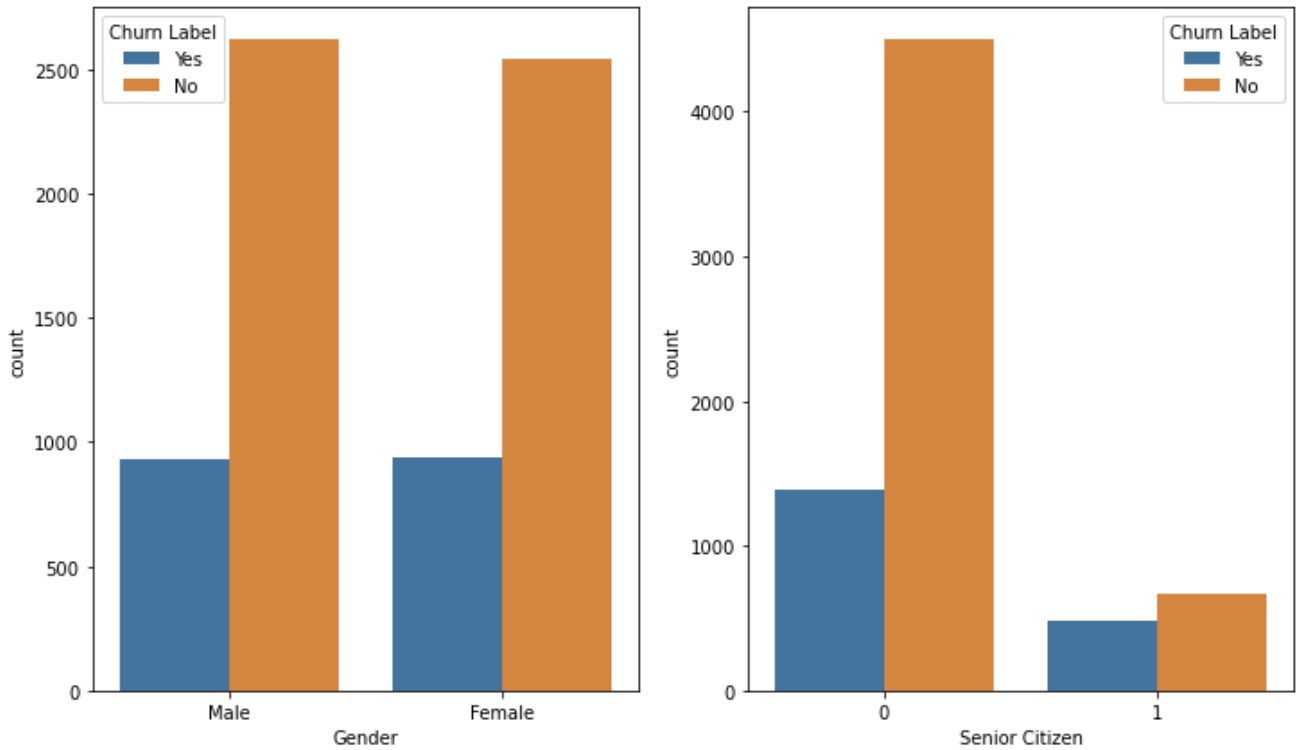
Churn = 'Yes'

	Tenure Months	Monthly Charges	Total Charges
count	1869.000000	1869.000000	1869.000000
mean	17.979133	74.441332	1531.796094
std	19.531123	24.666053	1890.822994
min	1.000000	18.850000	18.850000
25%	2.000000	56.150000	134.500000
50%	10.000000	79.650000	703.550000
75%	29.000000	94.200000	2331.300000
max	72.000000	118.350000	8684.800000

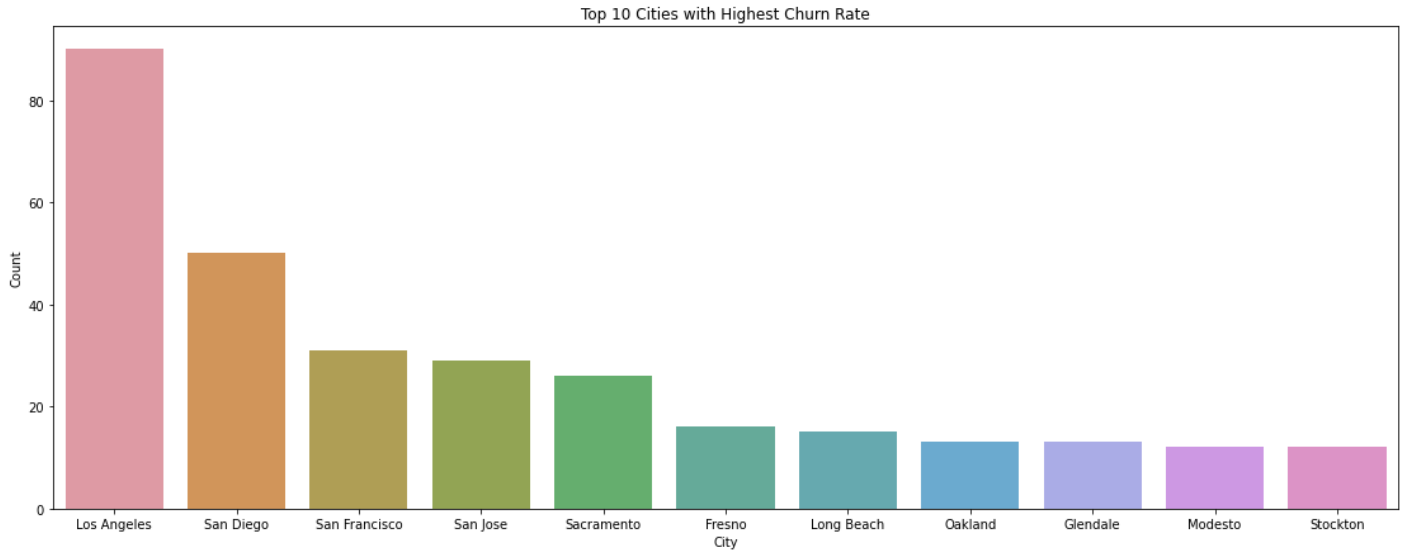
Churn = 'No'

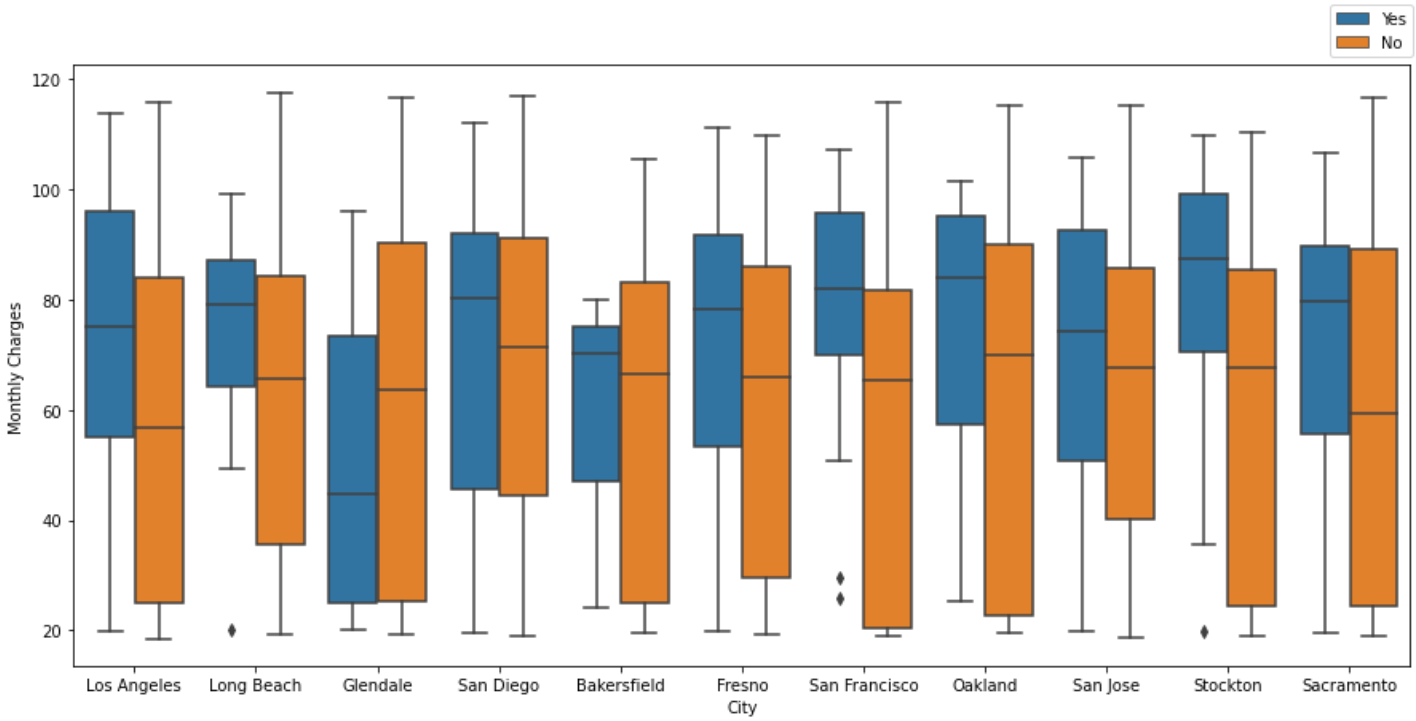
	Tenure Months	Monthly Charges	Total Charges
count	5163.00000	5163.000000	5163.000000
mean	37.65001	61.307408	2555.344141
std	24.07694	31.094557	2329.456984
min	1.00000	18.250000	18.800000
25%	15.00000	25.100000	577.825000
50%	38.00000	64.450000	1683.600000
75%	61.00000	88.475000	4264.125000
max	72.00000	118.750000	8672.450000

Distribution of the Class label



City-wise breakdown of the Class label





Data Classification Methods

Our ***end goal*** is to ***predict the "Churn Label,"*** our ***class attribute as 'Yes 'or 'No'*** based on the input dataset. Since there are only two possible values for our class attribute, we started looking at the best classification models for Binary Classification.
Popular algorithms that can be used for binary classification includes:

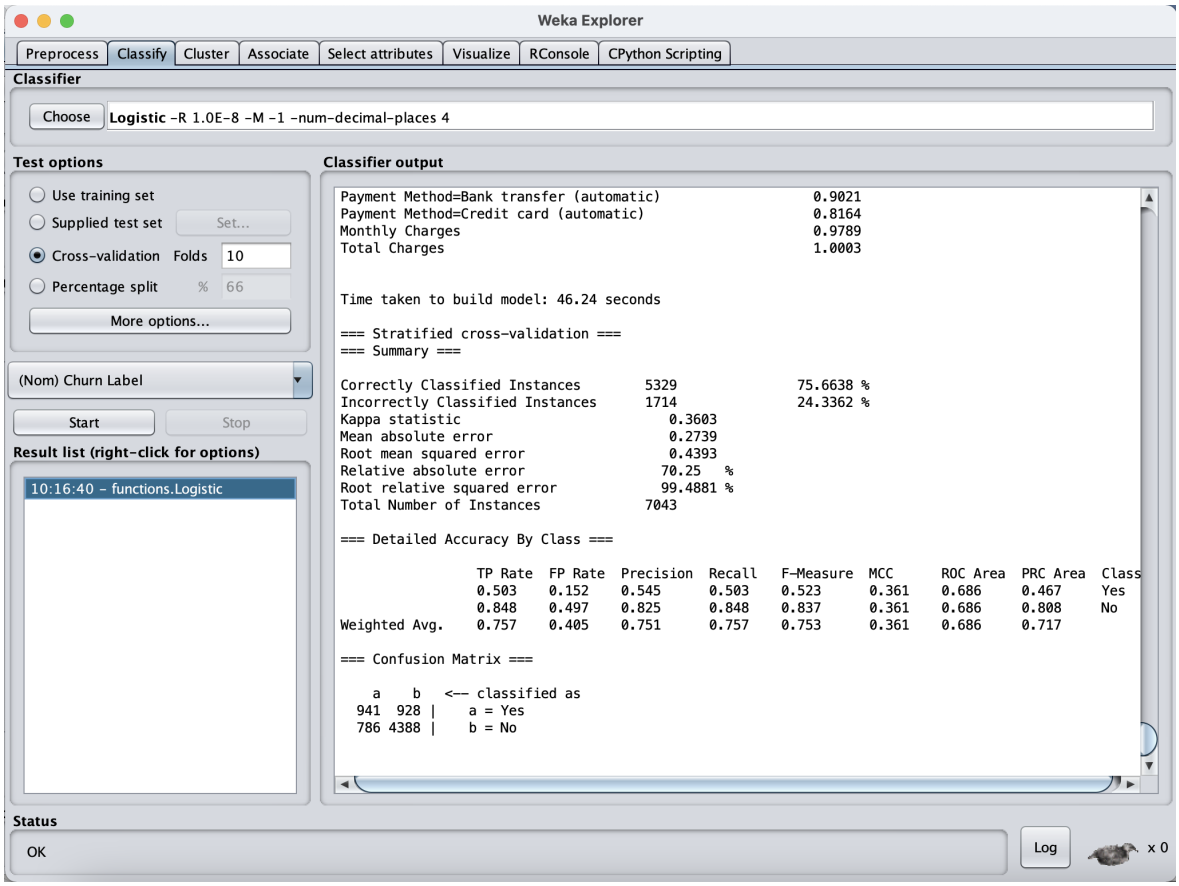
1. Logistic Regression
2. k-Nearest Neighbors
3. Decision Trees
4. Support Vector Machine
5. Naive Bayes

We tried, tested, and selected all except SVM from the above list since Weka's implementation for SVM (SMO) was computationally heavy to model the training set.
And, since Decision Tree gave us a reasonably good accuracy, we tried Random Forest, which actually was at par and even provided the best results among all for some instances.
Thus, we selected the below *five classification algorithms* for churn prediction.

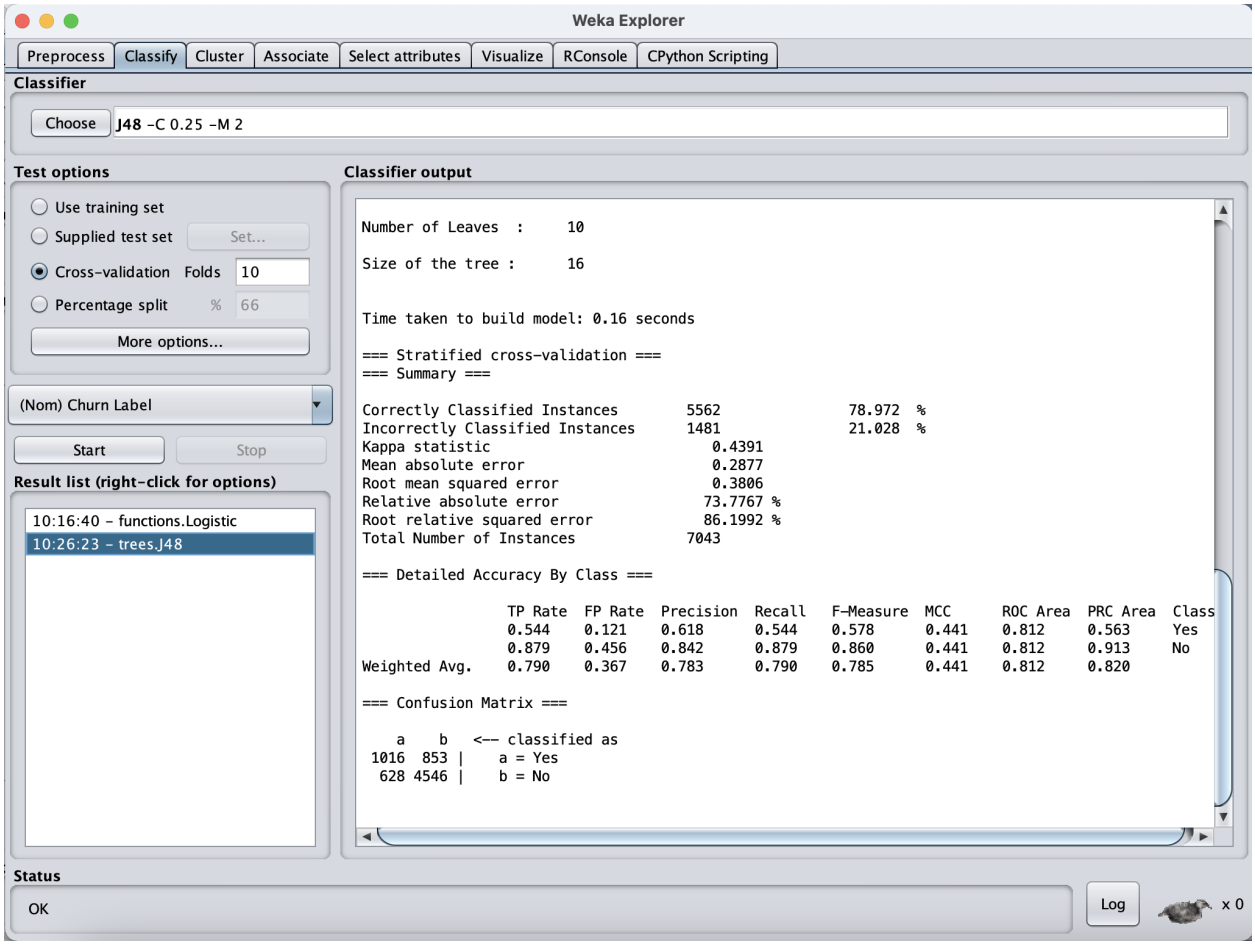
1. **Logistic**
2. **Decision Tree (J48)**
3. **Naïve Bayes**
4. **RandomForest**
5. **IBk (k=10)**

Classification Model Testing using 10 fold cross-validation

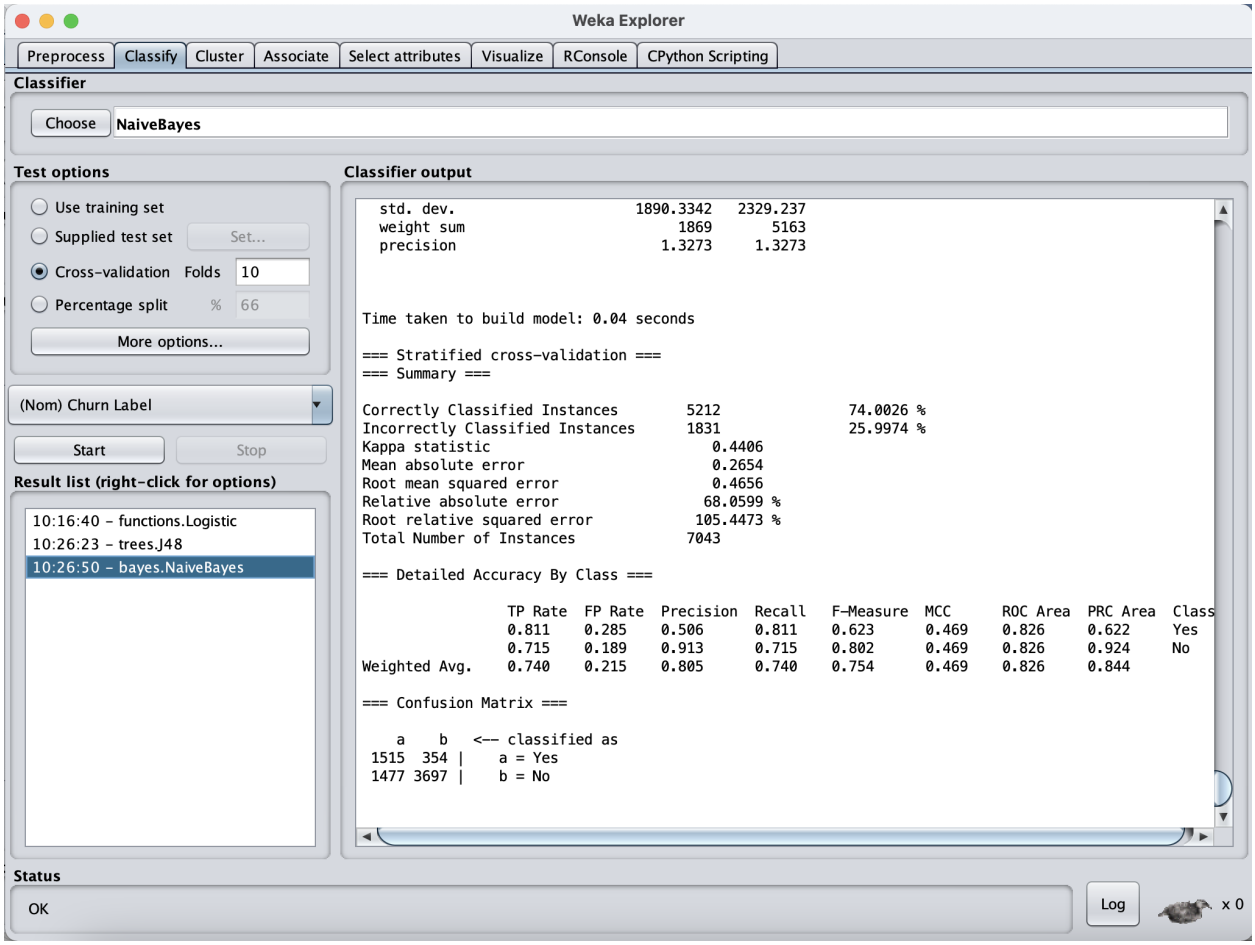
1. Logistic



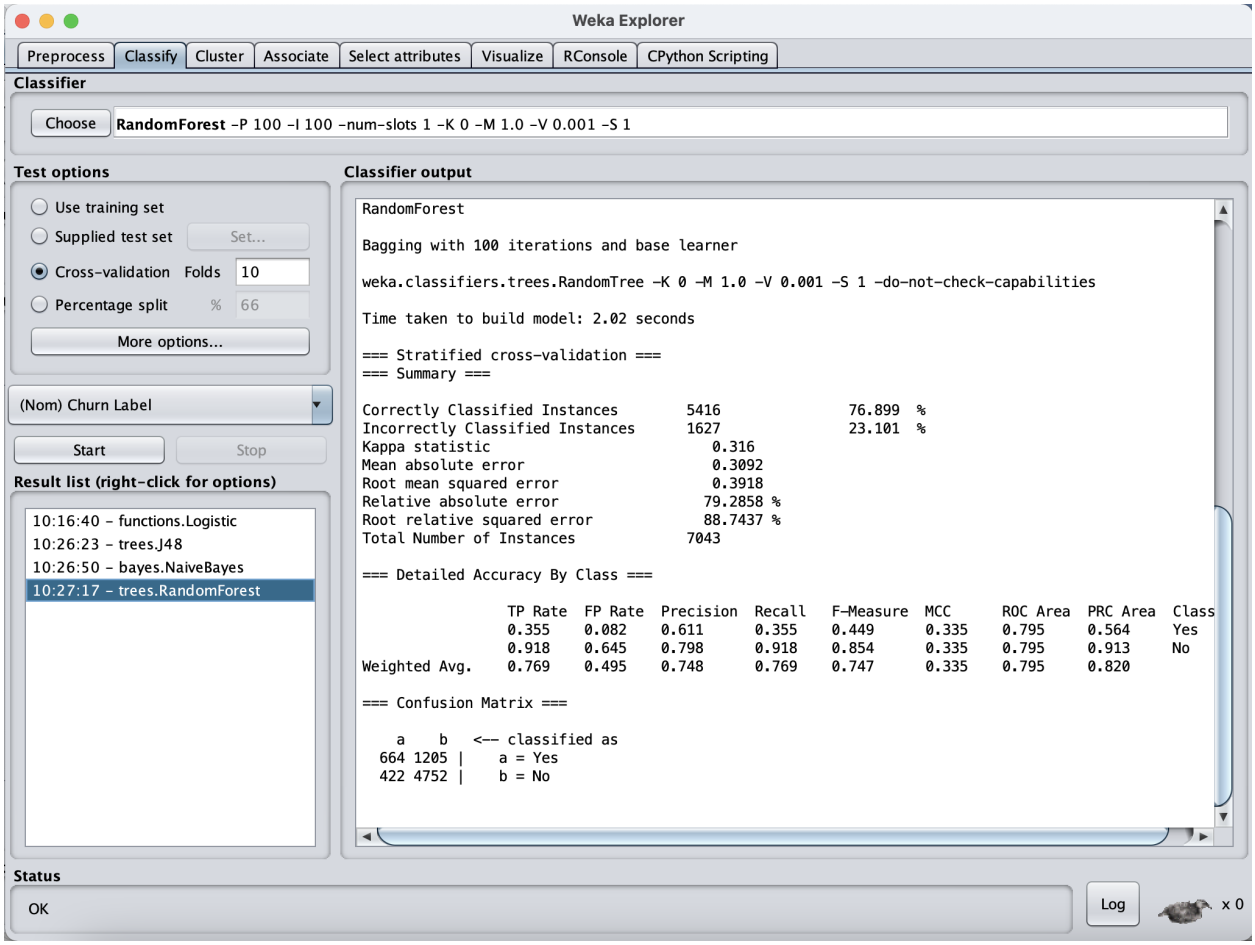
2. Decision Tree (J48)



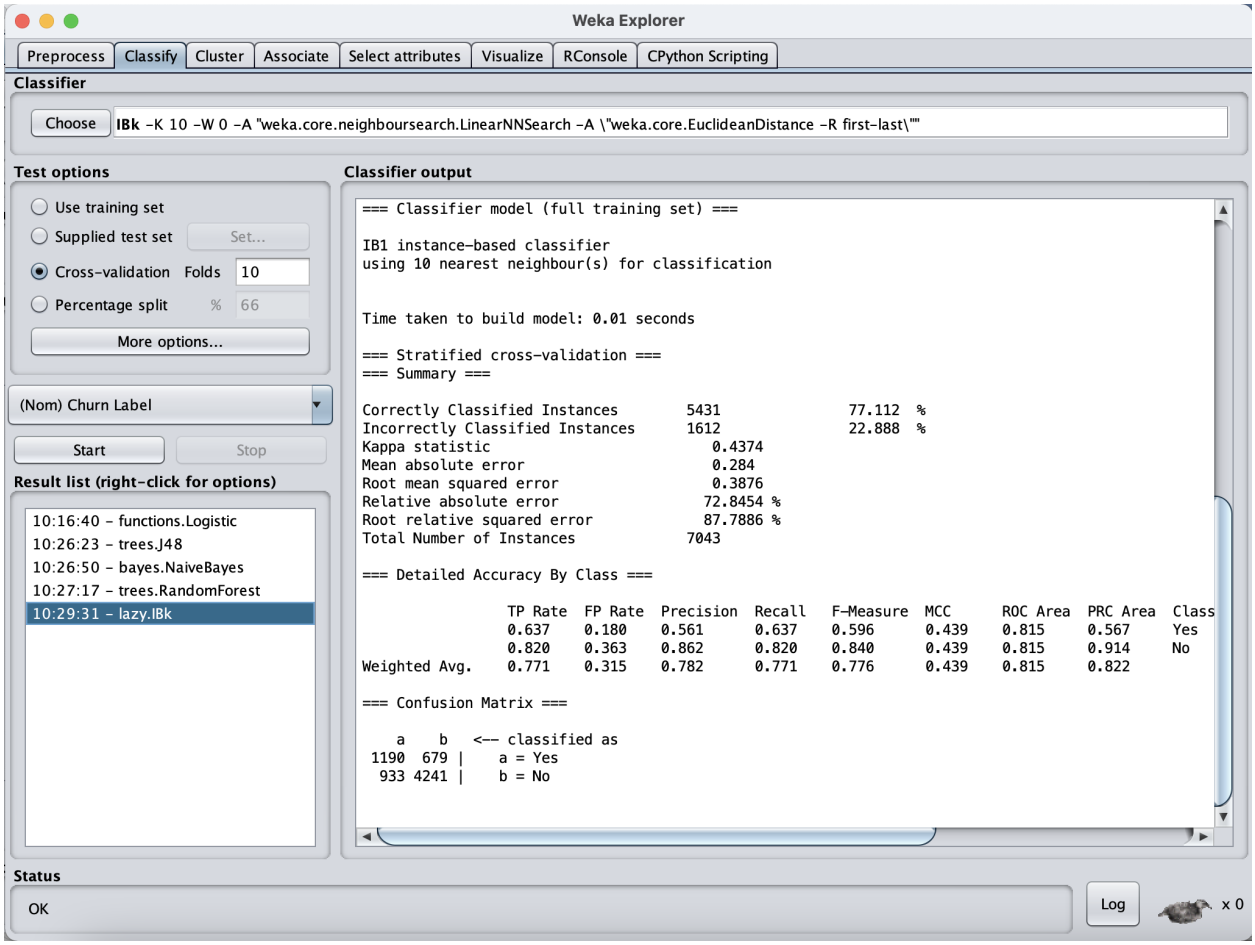
3. Naïve Bayes



4. RandomForest



5. IBk (k=10)



Preparing the data for classification

The data is split into training and testing instances using **the sklearn package**, StratifiedShuffleSplit to avoid class imbalance problems in the test and train sets. The split percentage is **66% training and 34% testing.**

The test and training datasets created using StratifiedShuffleSplit are:

Telco_customer_churn_training.arff with ***4648 instances***
Telco_customer_churn_testing.arff with ***2395 instances***

Attribute Selection Methods

Attribute Selection is performed on weka to find the most relevant attributes for predicting the Churn Label.
Attribute selection in weka is divided into two parts:

1. **Attribute Evaluator** is the technique by which each attribute in your dataset (also called a column or feature) is evaluated in the context of the output variable (e.g., the class).
2. **Search Method** is the technique to try or navigate different combinations of attributes in the dataset to arrive on a shortlist of chosen features.

Following are the attribute selection methods, along with the search method that we chose for our dataset.

- **CfsSubsetEval with BestFirst**

  CfsSubsetEval: Creates subsets that correlate highly with the class value and low correlation with each other.
  BestFirst: Uses a best-first search strategy to navigate attribute subsets.
  The subset of the attribute is used for the classification.

- **CorrelationAttributeEval with Ranker**

  A popular technique for selecting the most relevant attributes from the dataset is to use correlation. Correlation is more formally referred to as *Pearson's correlation coefficient* in statistics.
  In this evaluator, correlation is calculated between each attribute and the output variable. We selected only those attributes with a moderate-to-high positive or negative correlation (close to -1 or 1) and dropped those attributes with a low correlation (value close to zero).
  Also, the best attributes are selected based on rank.

- **GainRatioAttributeEval with Ranker**

  Here the *gain ratio* for the class is calculated. Best attributes are selected based on rank.

- **InfoGainAttributeEval with Ranker**

  Here the *information gain* (also called *entropy*) for each attribute for the output variable is calculated. Entry values vary from 0 (no information) to 1 (maximum information). Those attributes that contribute more information will have a higher information gain value and are selected. In contrast, those that do not add much information will have a lower score and are removed.
  Best attributes are selected based on rank.

- **Manual Selection based on EDA**

  We selected a list of attributes based on our exploratory data analysis and learning from running the above attribute selectors.

Reduced training and testing dataset using Attribute Selection methods

Using the following attribute selection methods, reduced training and testing datasets are created from Weka.

- **CfsSubsetEval with BestFirst**
- **CorrelationAttributeEval with Ranker**
- **GainRatioAttributeEval with Ranker**
- **InfoGainAttributeEval with Ranker**
- **Manual Selection based on EDA**

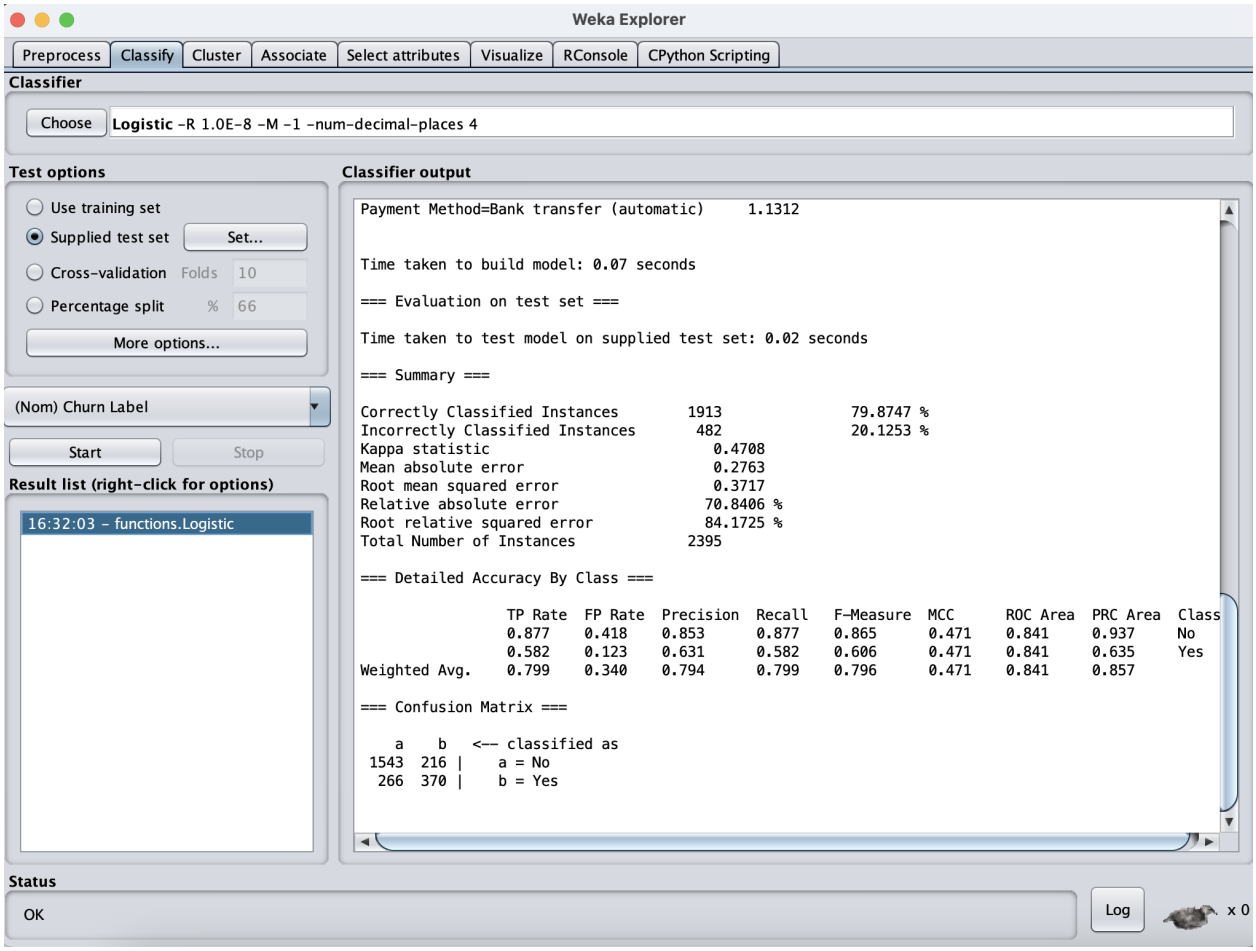
Attribute Selection (Selected Attributes)

BestFirst + CfsSubsetEval	Ranker + CorrelationAttributeEval		Ranker + GainRatioAttributeEval		Ranker + InfoGainAttributeEval		Manual Selection using EDA
Dependents	0.35598	Tenure Months	0.09755	Contract	0.21937	City	Zip Code
Tenure Months	0.3319	Contract	0.07669	Dependents	0.14061	Contract	Partner
Online Security	0.27125	Online Security	0.06285	Online Security	0.11355	Tenure Months	Dependents
Tech Support	0.26611	Tech Support	0.06108	Tech Support	0.09411	Online Security	Tenure Months
Contract	0.2582	Dependents	0.05263	Internet Service	0.09164	Tech Support	Multiple Lines
Payment Method	0.23021	Internet Service	0.04805	Tenure Months	0.08052	Internet Service	Internet Service
	0.20135	Total Charges	0.04179	Device Protection	0.06318	Monthly Charges	Tech Support
							Contract
							Monthly Charges

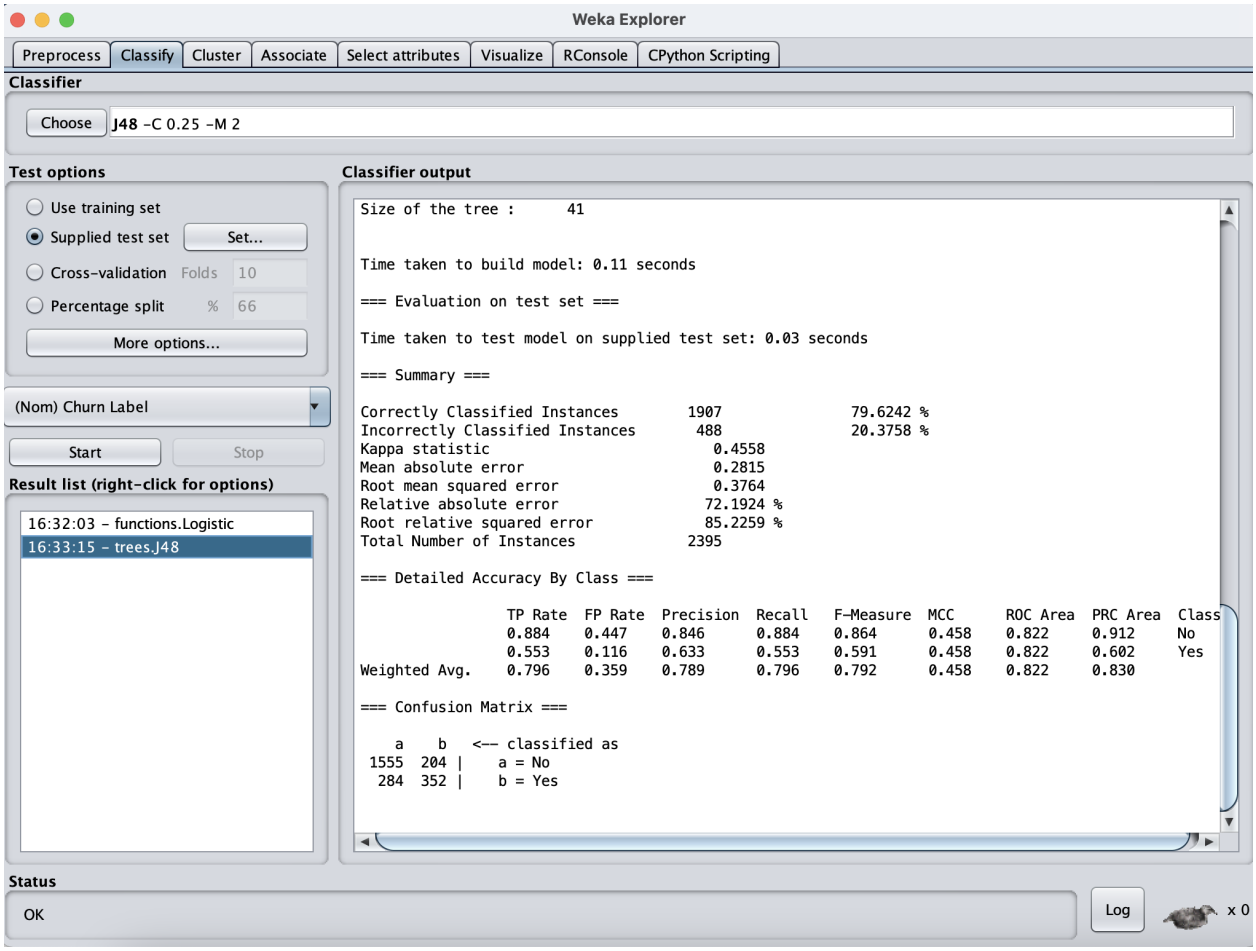
Classifier Models

We ran the identified classification algorithms on the reduced datasets created using different attribute selection methods from above.
Below are those 25 models:

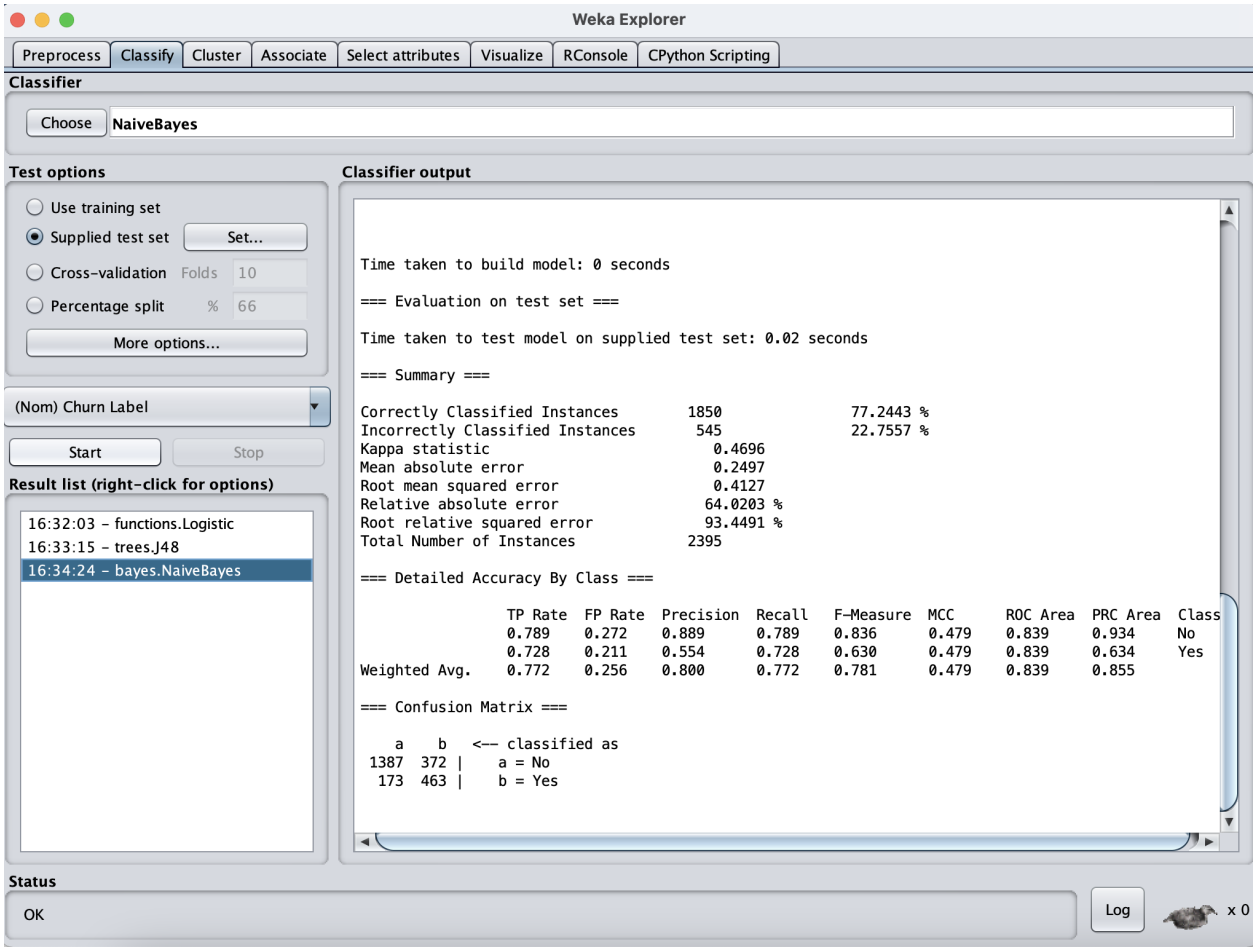
1. Classification Algorithm: *Logistic* Attribute Selector: *CfsSubsetEval*



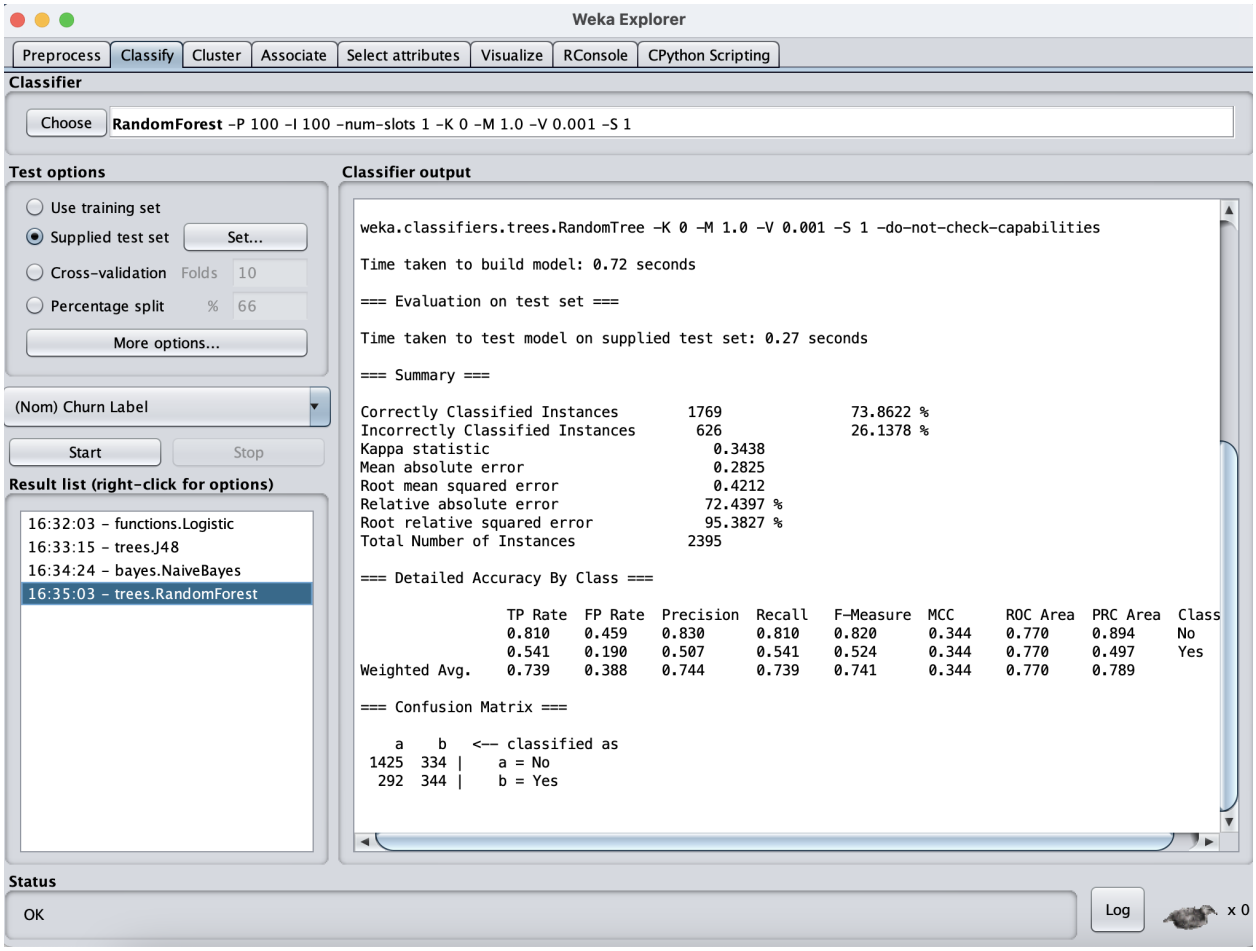
2. Classification Algorithm: *Decision Tree(J48)* Attribute Selector: *CfsSubsetEval*



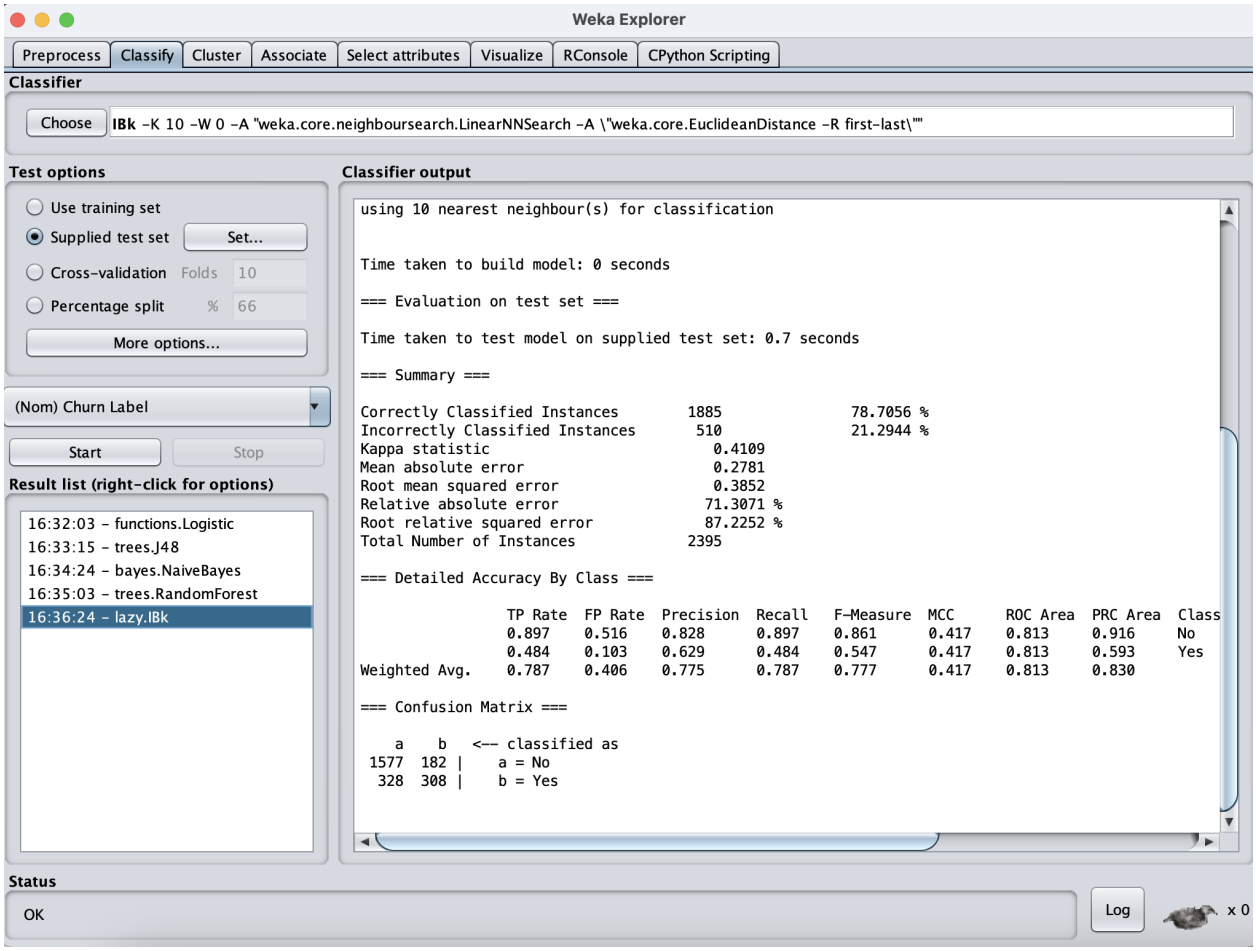
3. Classification Algorithm: *Naïve Bayes* Attribute Selector: *CfsSubsetEval*



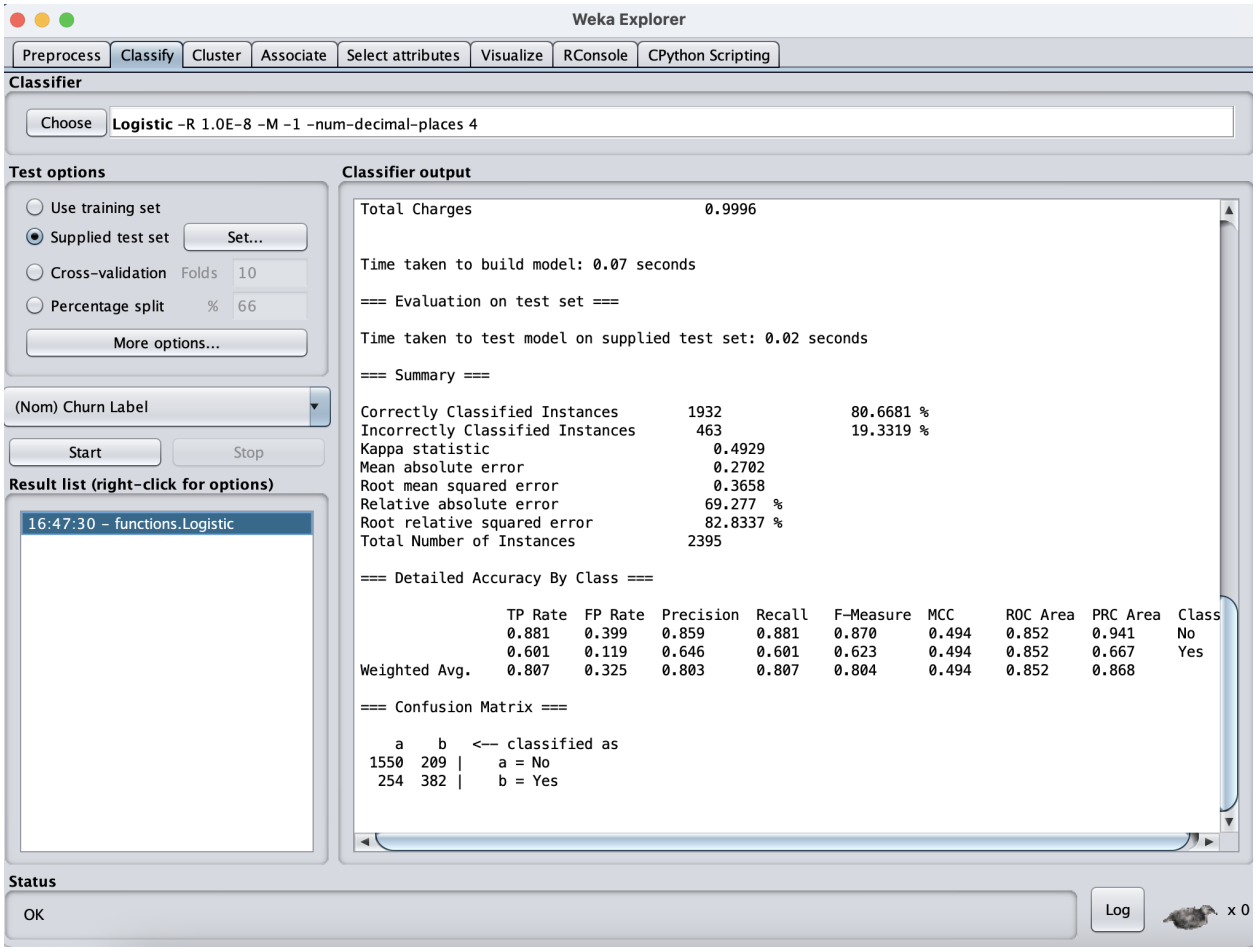
4. Classification Algorithm: *Random Forest* Attribute Selector: *CfsSubsetEval*



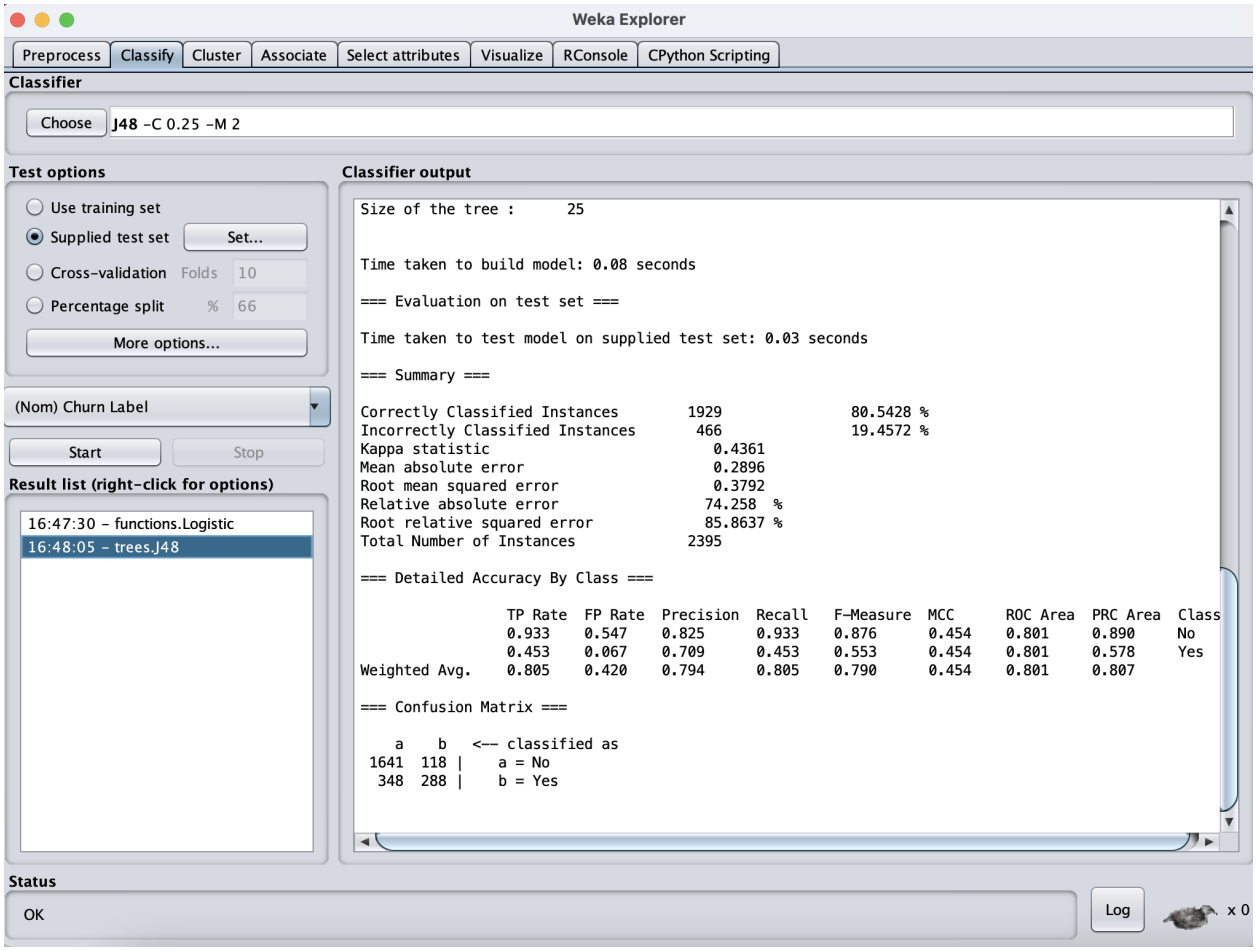
5. Classification Algorithm: *IBk(k=10)* Attribute Selector: *CfsSubsetEval*



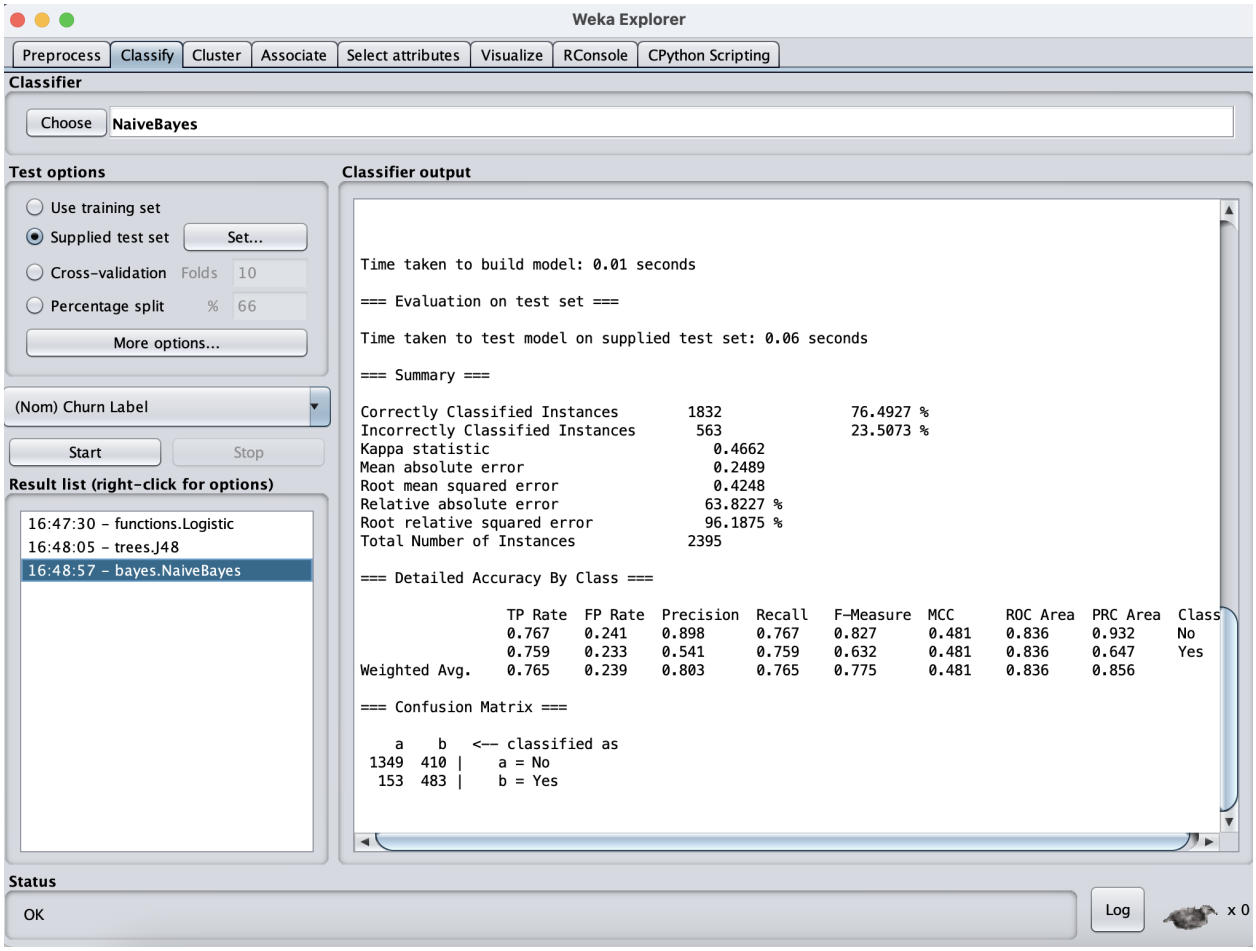
6. Classification Algorithm: *Logistic* Attribute Selector: *CorrelationAttributeEval*



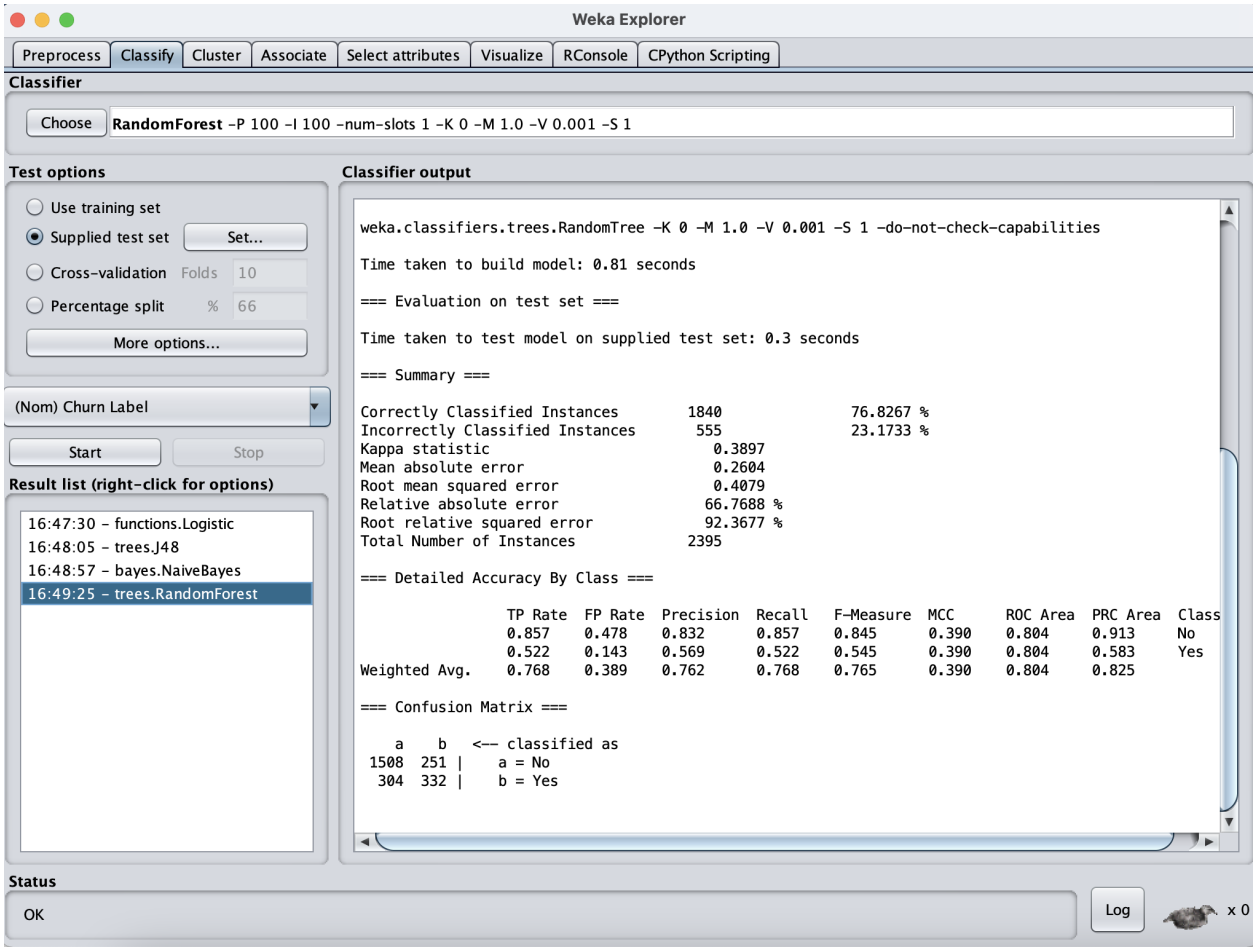
7. Classification Algorithm: *Decision Tree(J48)* Attribute Selector: *CorrelationAttributeEval*



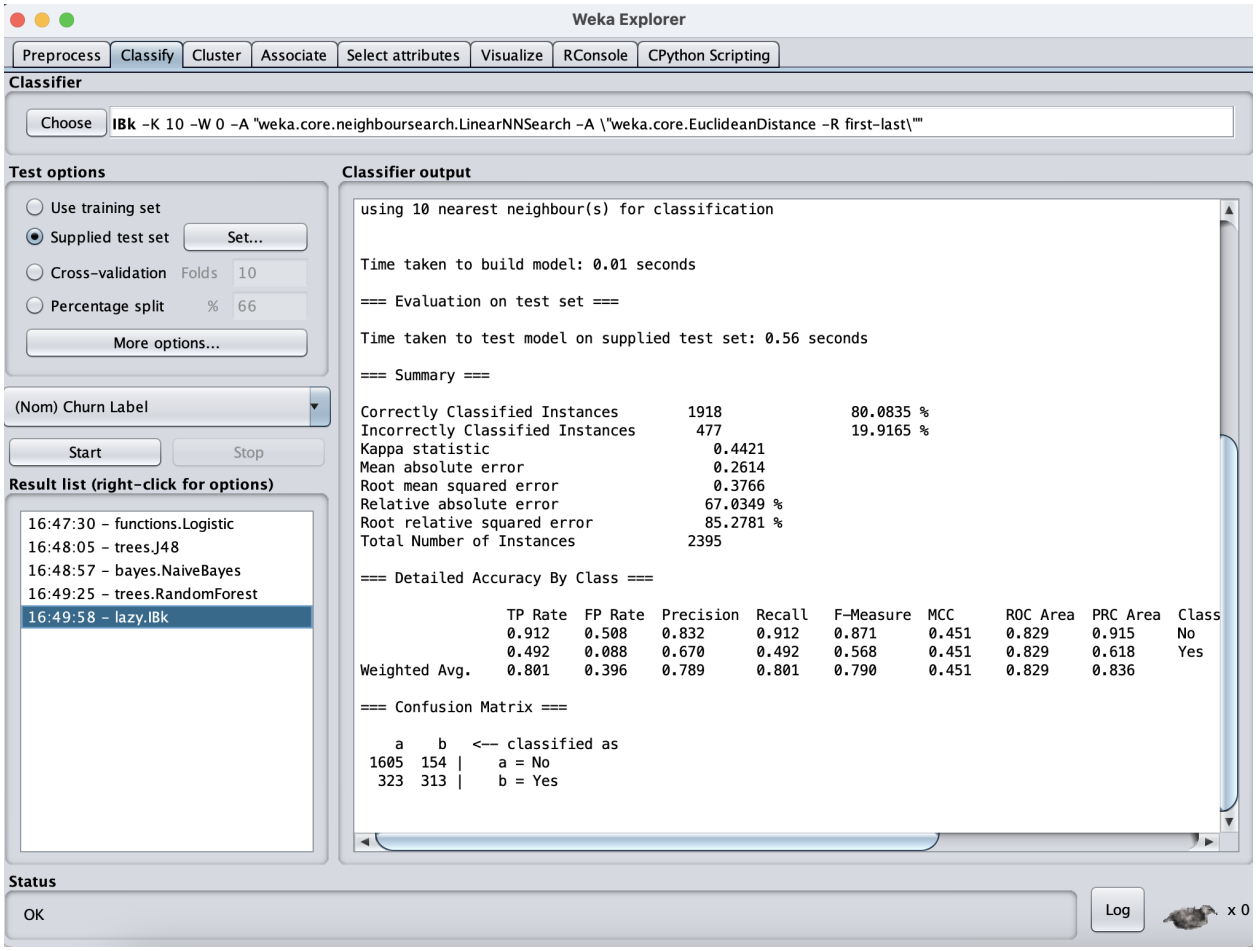
8. Classification Algorithm: *Naïve Bayes* Attribute Selector: *CorrelationAttributeEval*



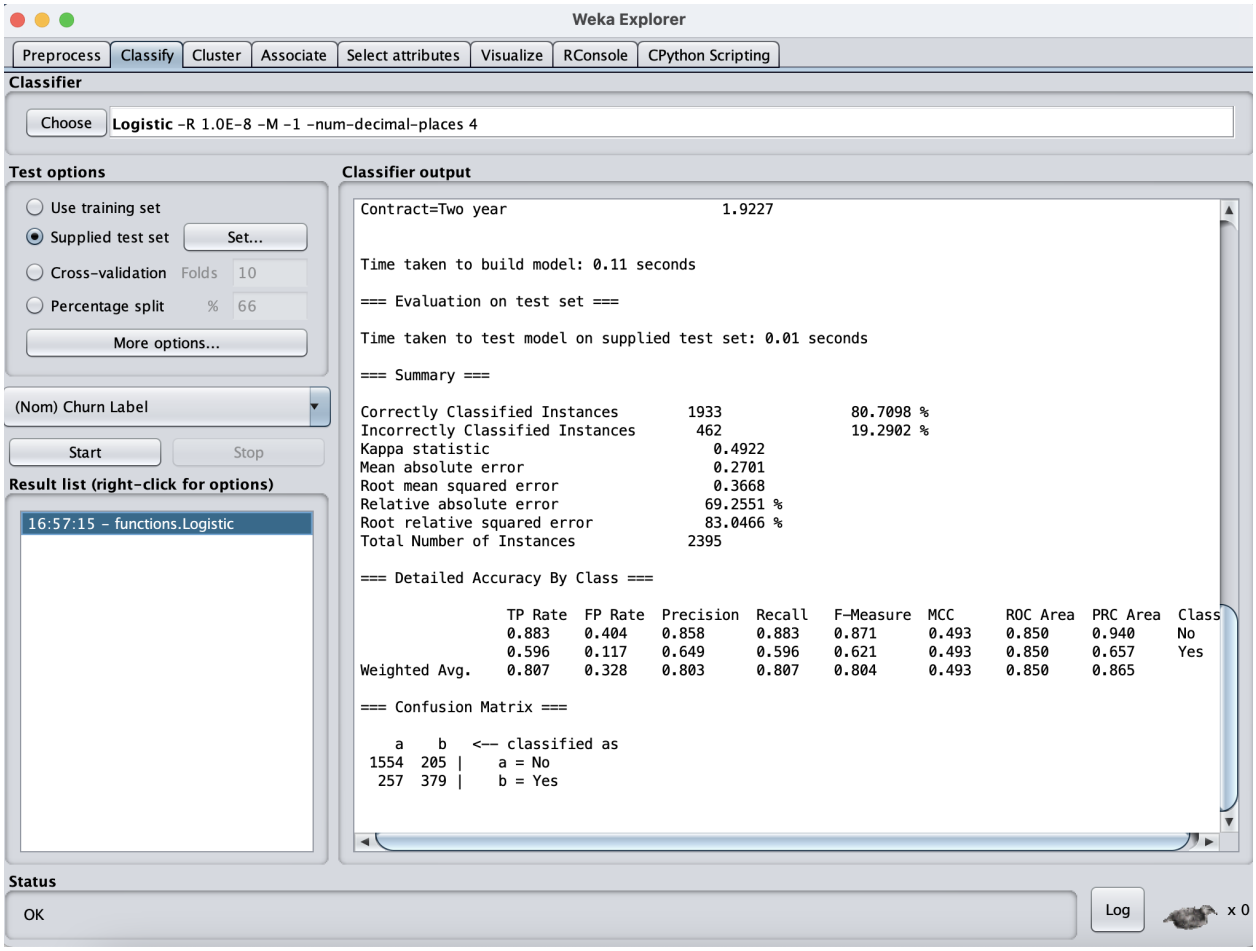
9. Classification Algorithm: *Random Forest* Attribute Selector: *CorrelationAttributeEval*



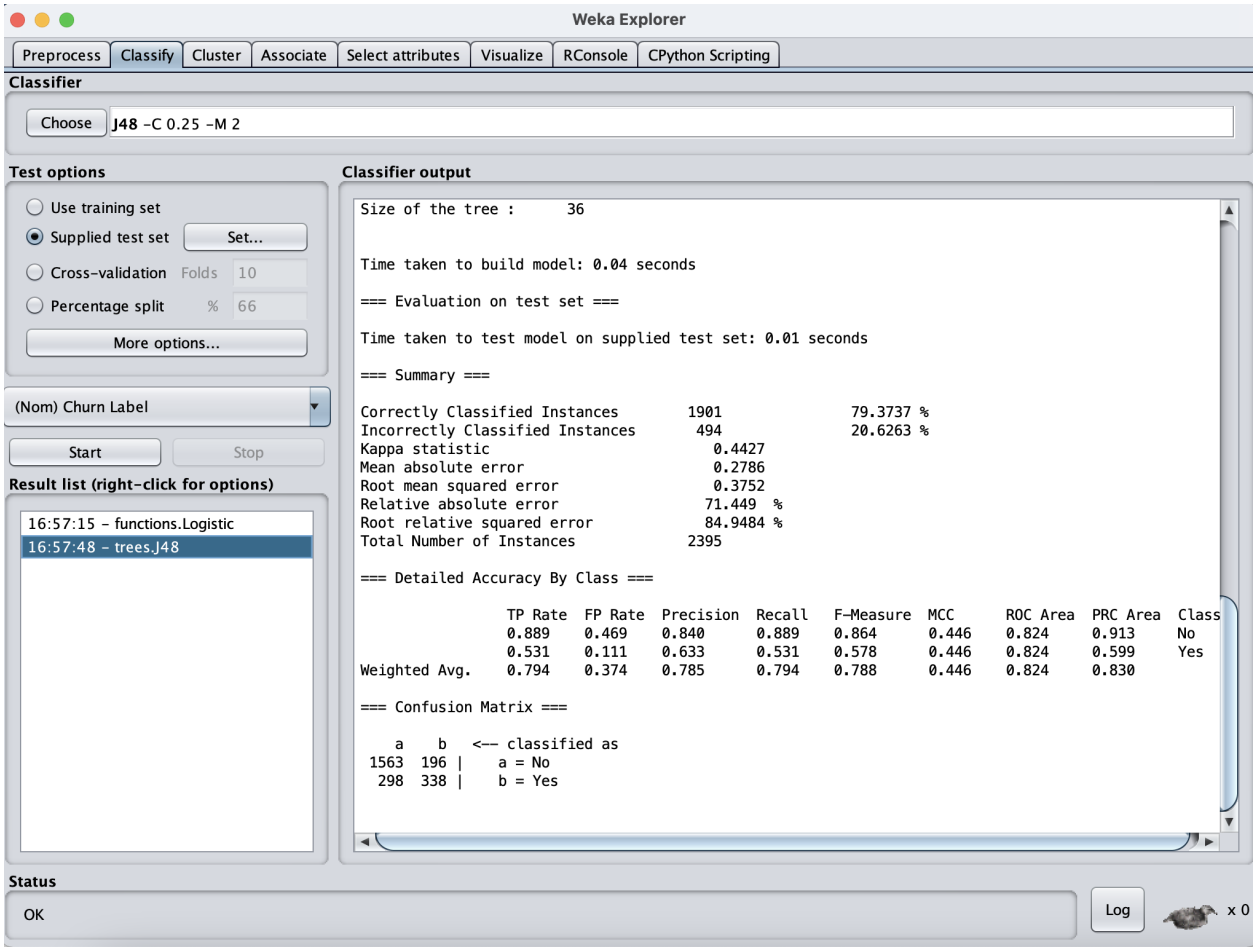
10. Classification Algorithm: *IBk(k=10)* Attribute Selector: *CorrelationAttributeEval*



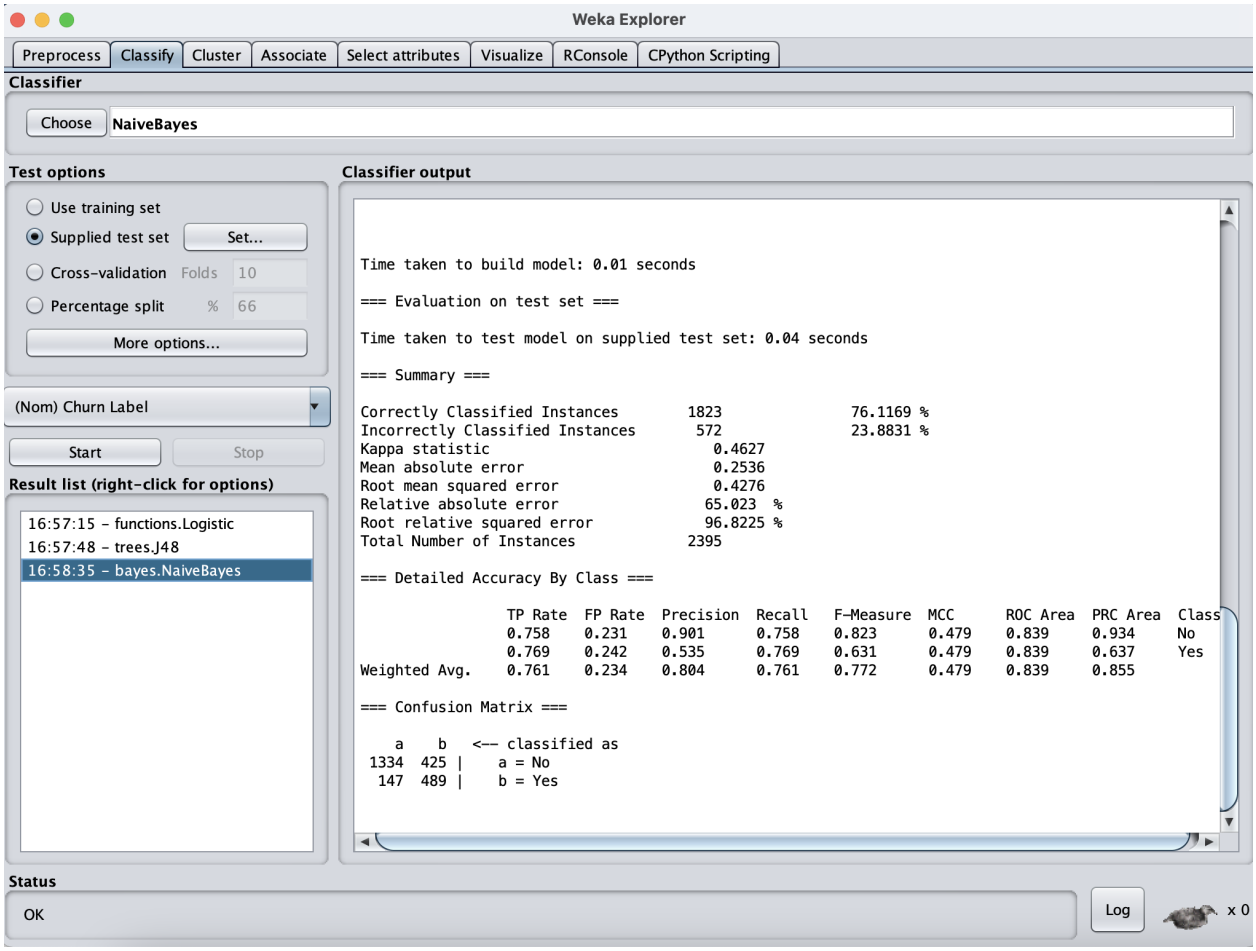
11. Classification Algorithm: *Logistic* Attribute Selector: *GainRatioAttributeEval*



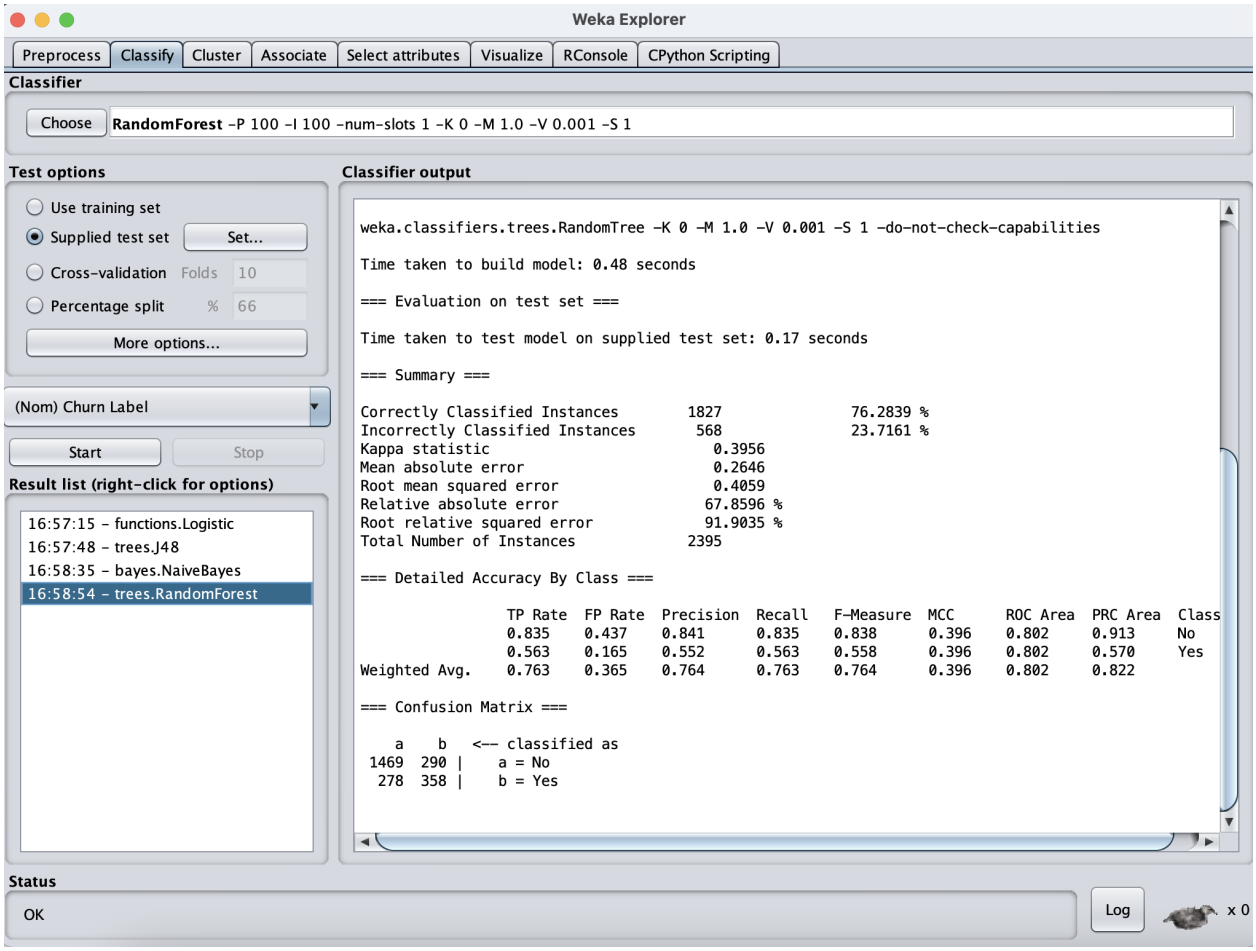
12. Classification Algorithm: *Decision Tree(J48)* Attribute Selector: *GainRatioAttributeEval*



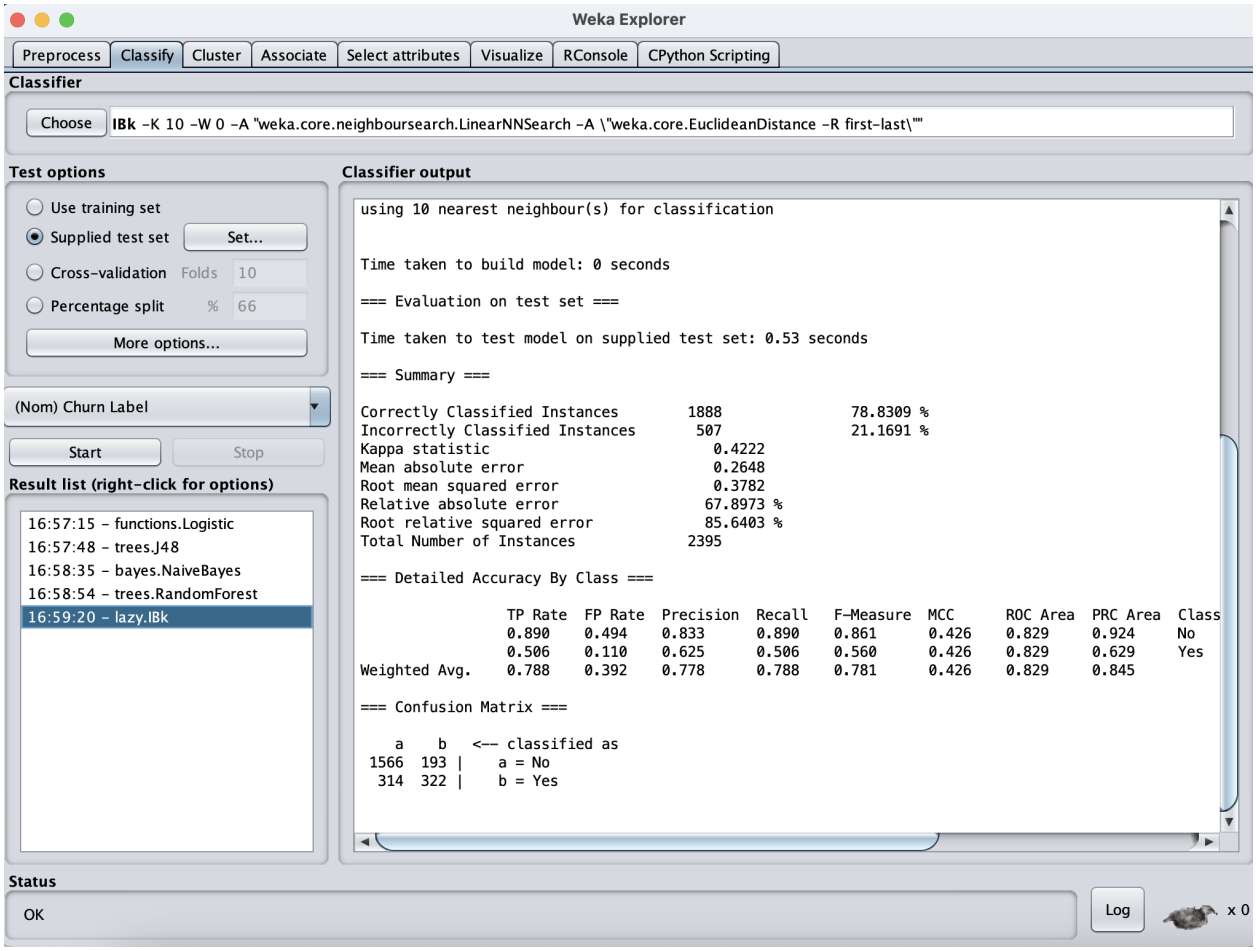
13. Classification Algorithm: *Naïve Bayes* Attribute Selector: *GainRatioAttributeEval*



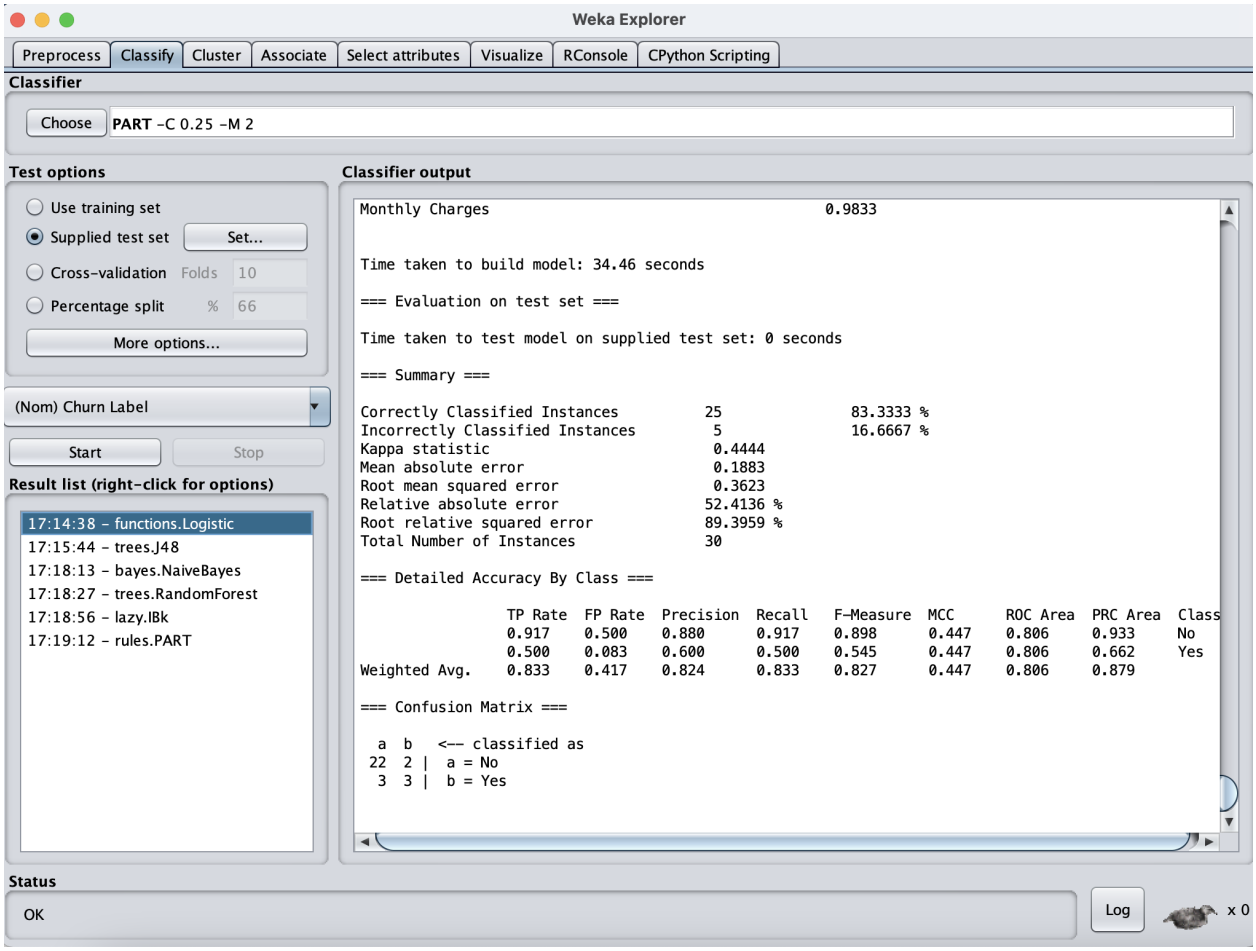
14. Classification Algorithm: *Random Forest* Attribute Selector: *GainRatioAttributeEval*



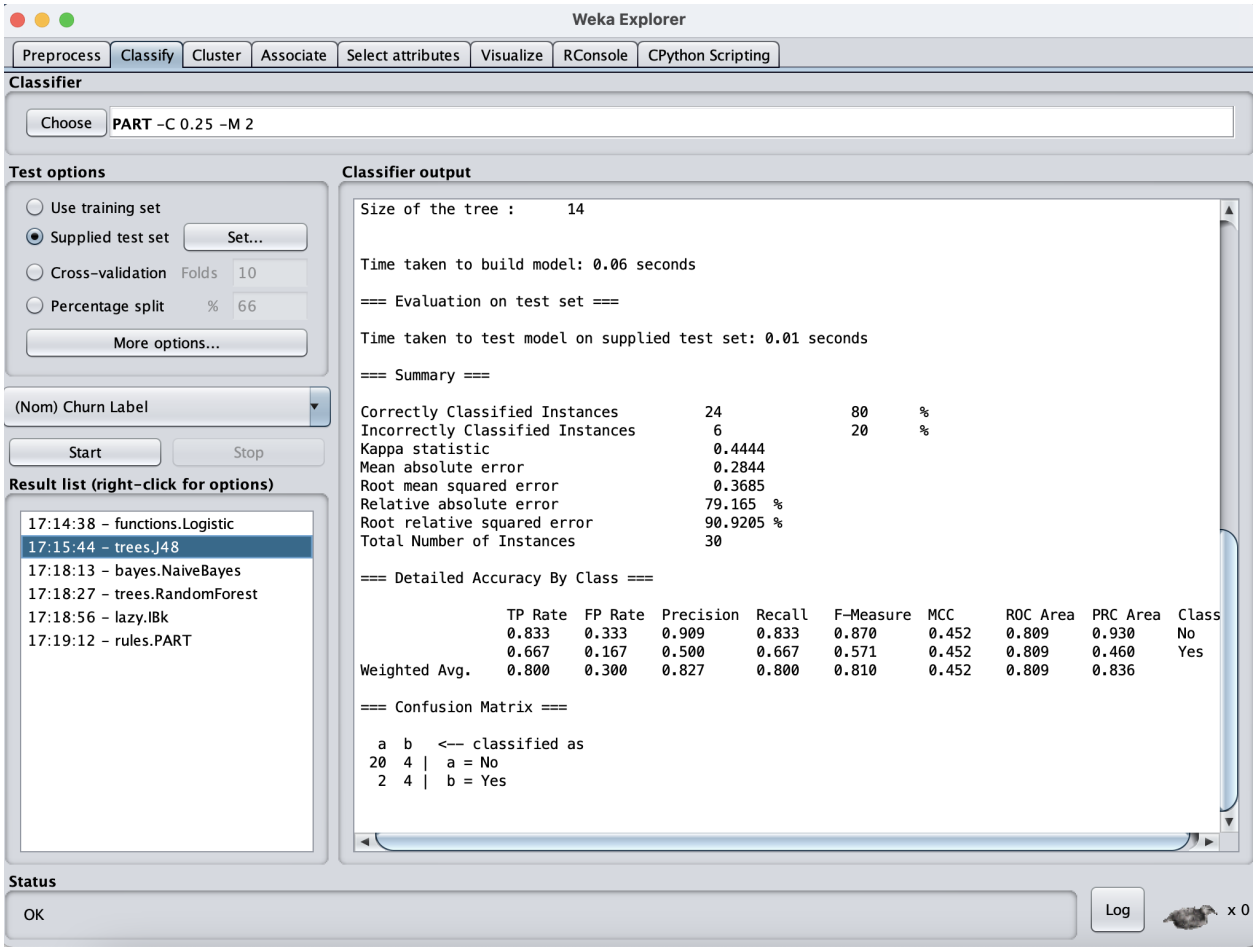
15. Classification Algorithm: *IBk(k=10)* Attribute Selector: *GainRatioAttributeEval*



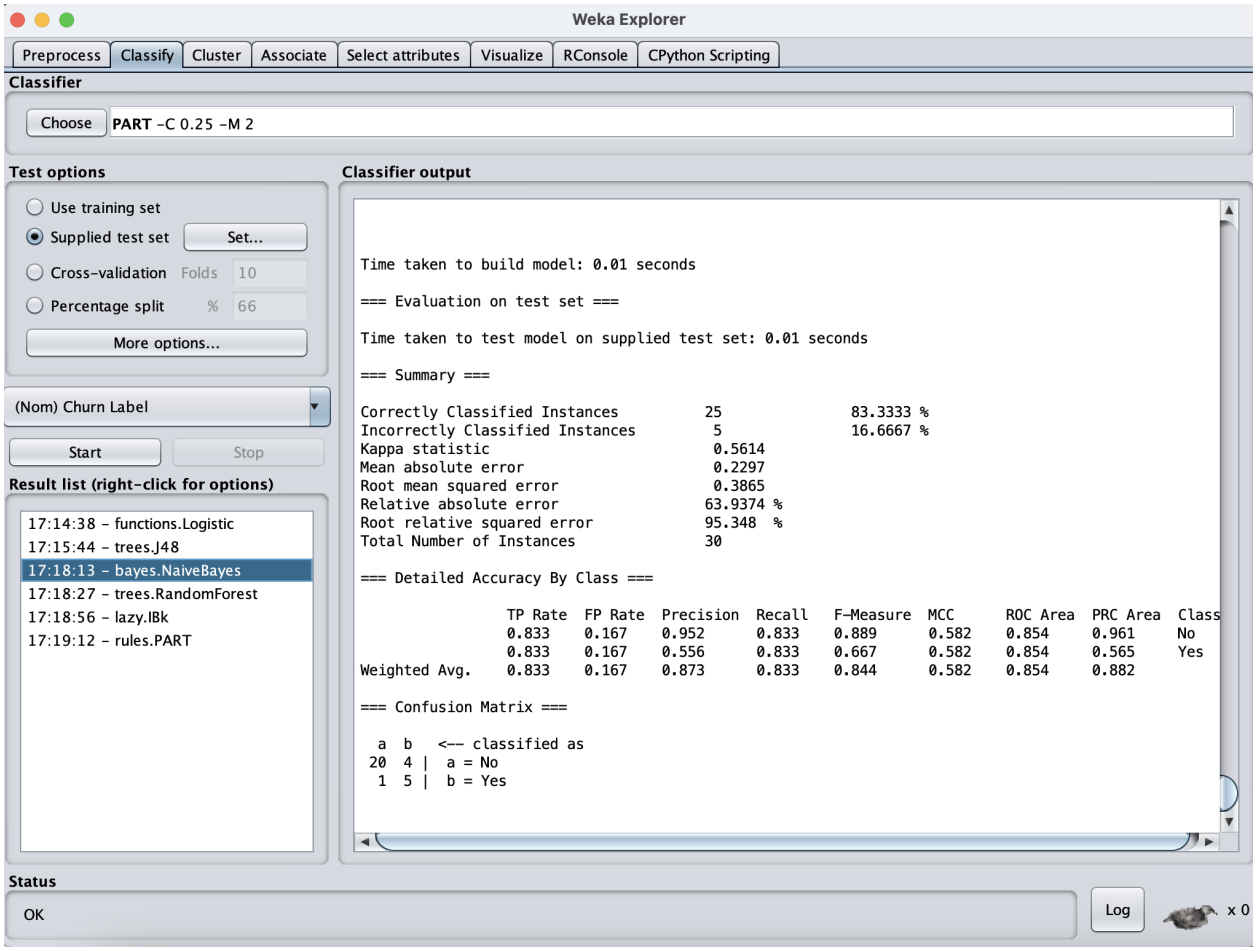
16. Classification Algorithm: *Logistic* Attribute Selector: *InfoGainAttributeEval*



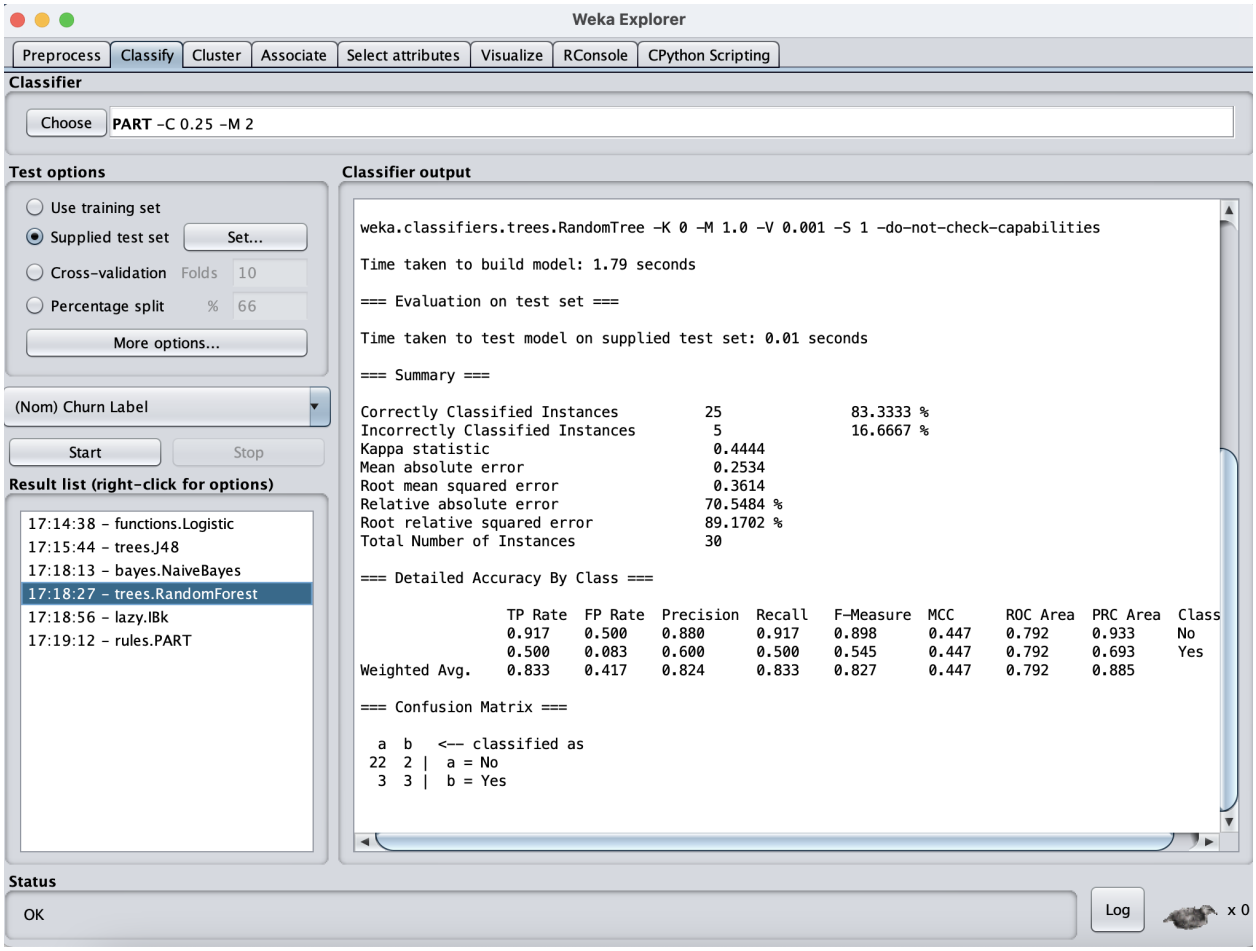
17. Classification Algorithm: *Decision Tree(J48)* Attribute Selector: *InfoGainAttributeEval*



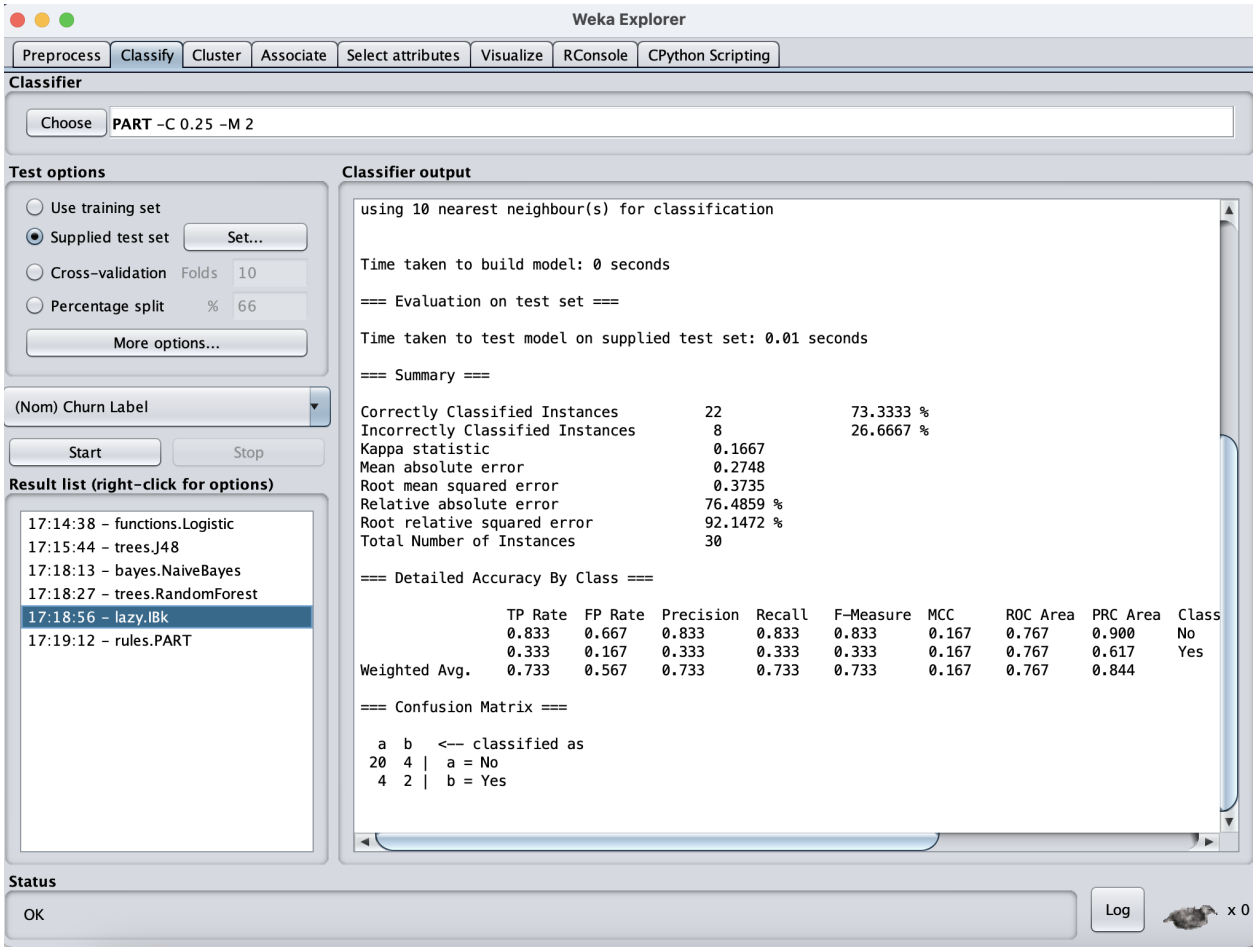
18. Classification Algorithm: *Naïve Bayes* Attribute Selector: *InfoGainAttributeEval*



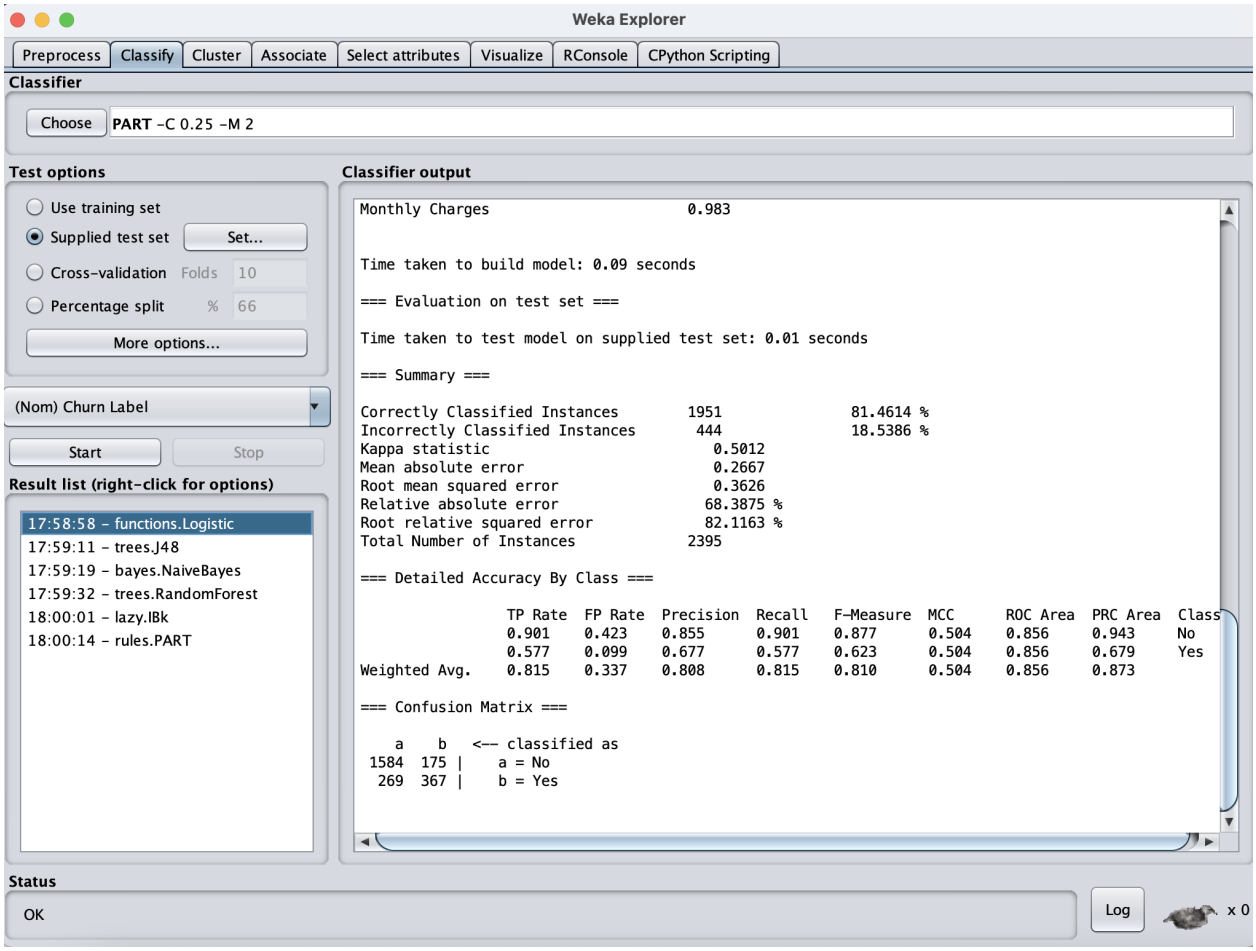
19. Classification Algorithm: *Random Forest* Attribute Selector: *InfoGainAttributeEval*



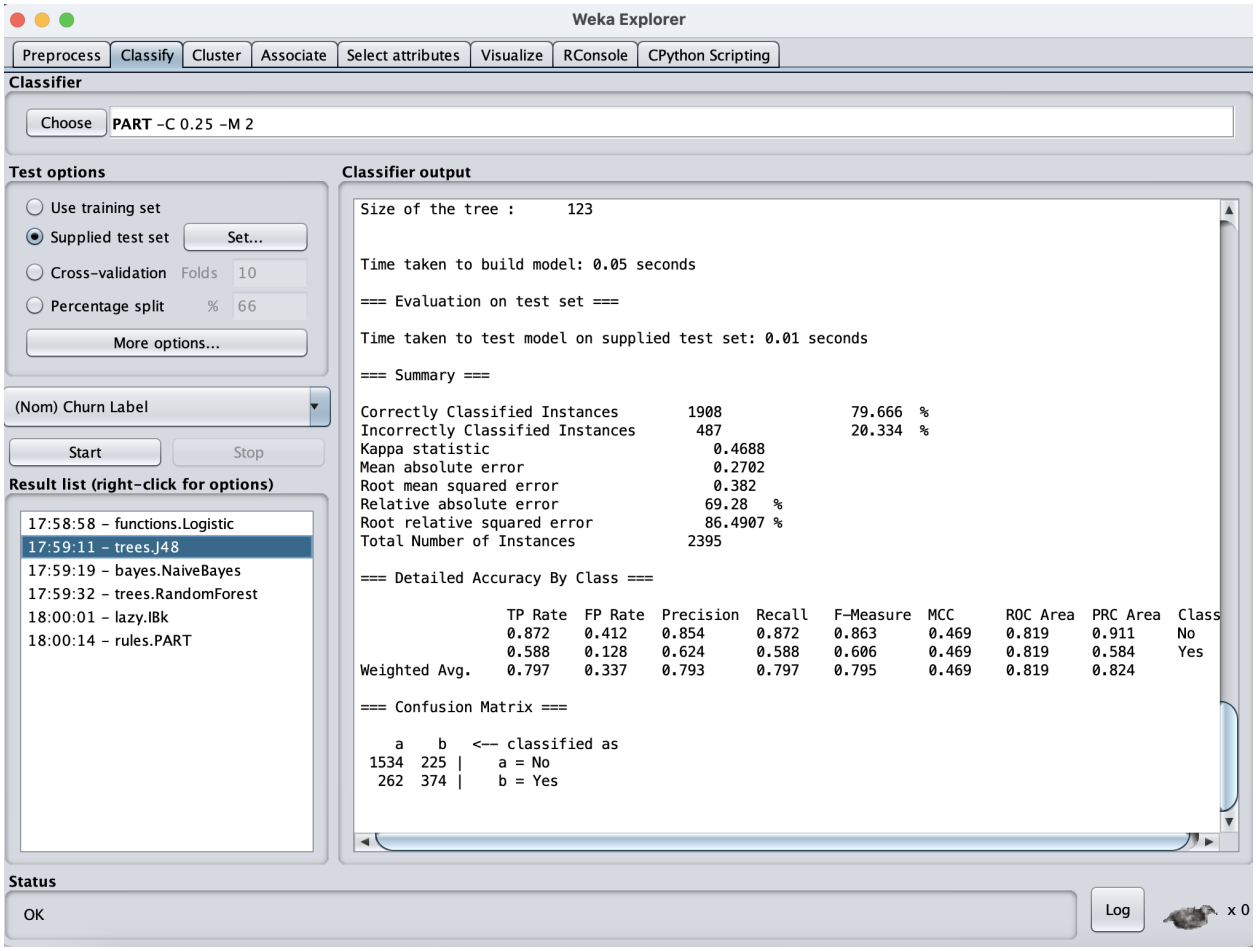
20. Classification Algorithm: *IBk(k=10)* Attribute Selector: *InfoGainAttributeEval*



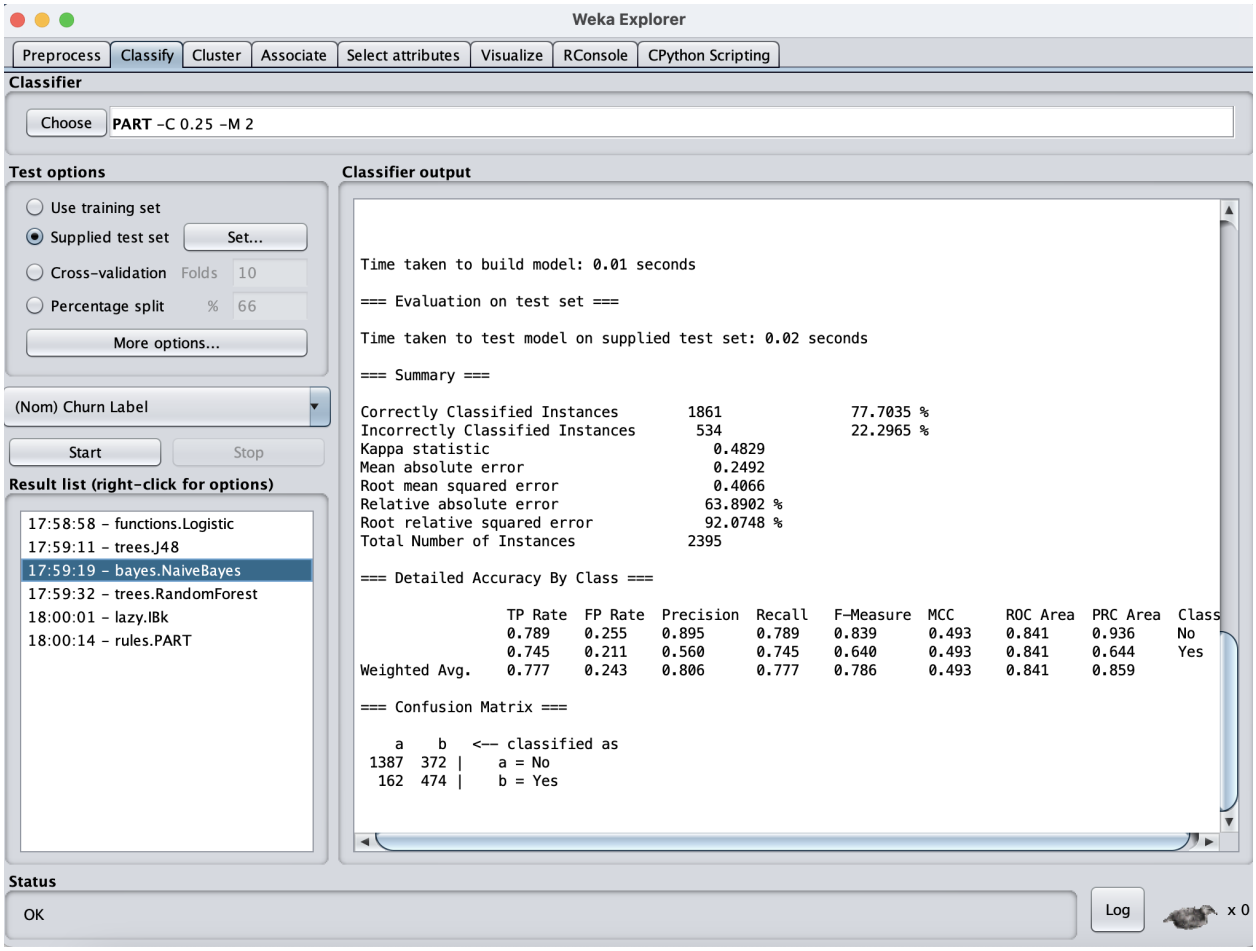
21. Classification Algorithm: *Logistic* Attribute Selector: *Manual Selection*



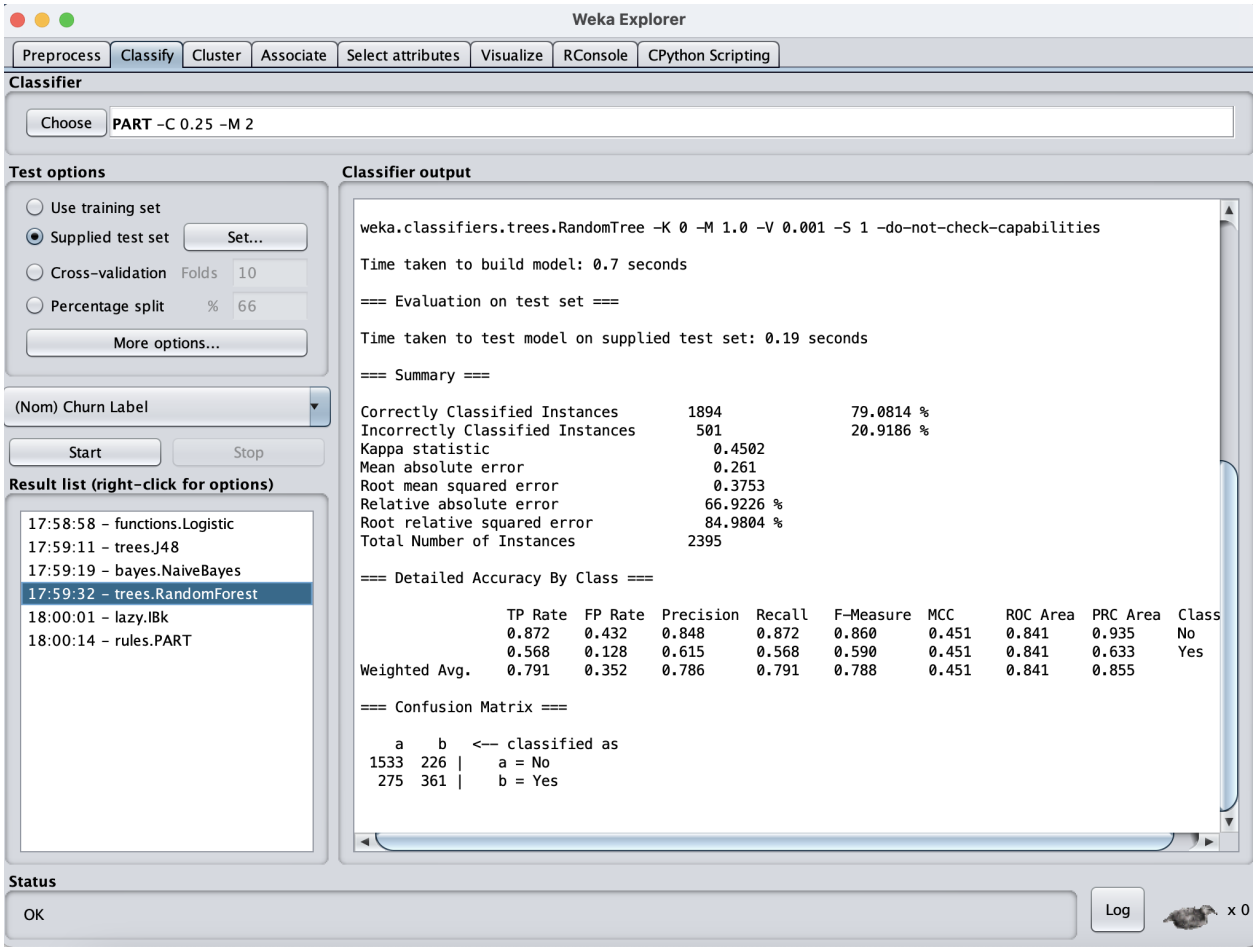
22. Classification Algorithm: *Decision Tree(J48)* Attribute Selector: *Manual Selection*



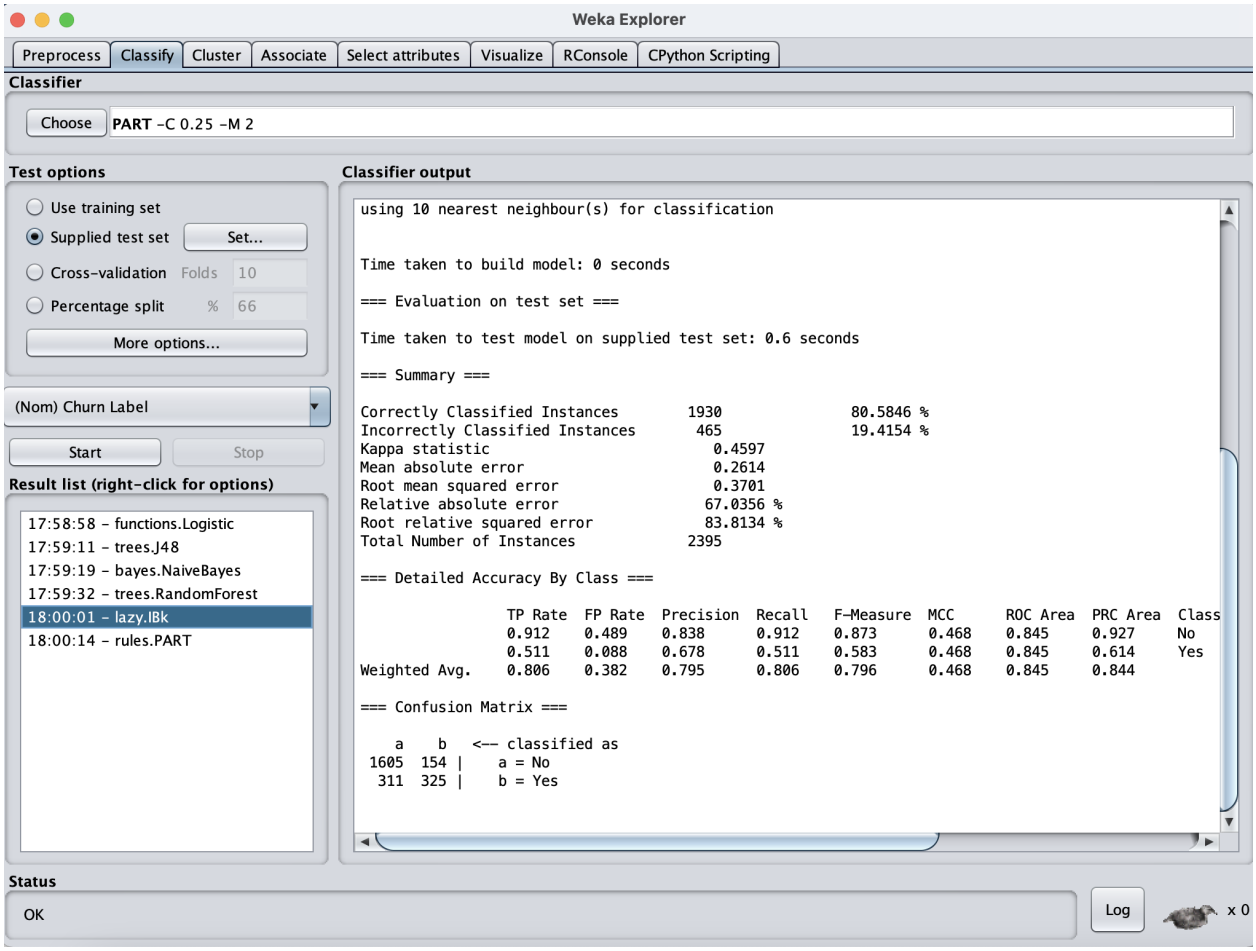
23. Classification Algorithm: *Naïve Bayes* Attribute Selector: *Manual Selection*



24. Classification Algorithm: *Random Forest* Attribute Selector: *Manual Selection*



25. Classification Algorithm: *IBk(k=10)* Attribute Selector: *Manual Selection*



Model Evaluation

Following are the metrics considered for the evaluation of models.

- Accuracy
- RMSE
- TPR for Class = “Yes”

We considered RMSE and TPR since the data set is imbalanced. About **70%** of the data belongs to **class “No”** and only 30 % for class “Yes.” So just looking at accuracy may give us a false sense of higher accuracy. To account for the cost of misclassification of the minor sample (class = “Yes”), we considered TPR for class = ”Yes.” Also, from a business perspective, it’s more *important to predict a customer who is more likely to churn (class = “Yes”)* versus not churn (class = “No”).

Model Accuracy (%)

	Cross-Validation	CFS_BF	Correlation	GainRatio	InfoGain	Manual	Average
Logistic	75.66	79.87	80.67	80.71	83.33	81.46	**81.21**
J48	78.97	79.62	80.54	79.37	80.00	79.67	**79.84**
RF	76.90	73.86	76.83	76.28	83.33	79.08	**77.88**
NB	74.00	77.24	76.49	76.12	83.33	77.70	**78.18**
IBk	77.11	78.71	80.08	78.83	73.33	80.58	**78.31**
Average	**76.53**	**77.86**	**78.92**	**78.26**	**80.66**	**79.70**	

RMSE

	Cross-Validation	CFS_BF	Correlation	GainRatio	InfoGain	Manual	Average
Logistic	0.44	0.37	0.37	0.37	0.36	0.36	**0.37**
J48	0.38	0.38	0.38	0.38	0.37	0.38	**0.38**
RF	0.39	0.42	0.41	0.41	0.36	0.38	**0.40**
NB	0.47	0.41	0.43	0.43	0.39	0.41	**0.41**
IBk	0.39	0.39	0.38	0.38	0.37	0.37	**0.38**
Average	**0.41**	**0.39**	**0.39**	**0.39**	**0.37**	**0.38**	

TPR (class='Yes')

	Cross-Validation	CFS_BF	Correlation	GainRatio	InfoGain	Manual	Average
Logistic	0.50	0.58	0.60	0.60	0.50	0.58	**0.57**
J48	0.54	0.55	0.45	0.53	0.67	0.59	**0.56**
RF	0.36	0.54	0.52	0.56	0.83	0.57	**0.61**
NB	0.81	0.73	0.76	0.77	0.83	0.75	**0.77**
IBk	0.64	0.48	0.49	0.51	0.33	0.51	**0.47**
Average	**0.57**	**0.58**	**0.57**	**0.59**	**0.63**	**0.60**	

Best Model

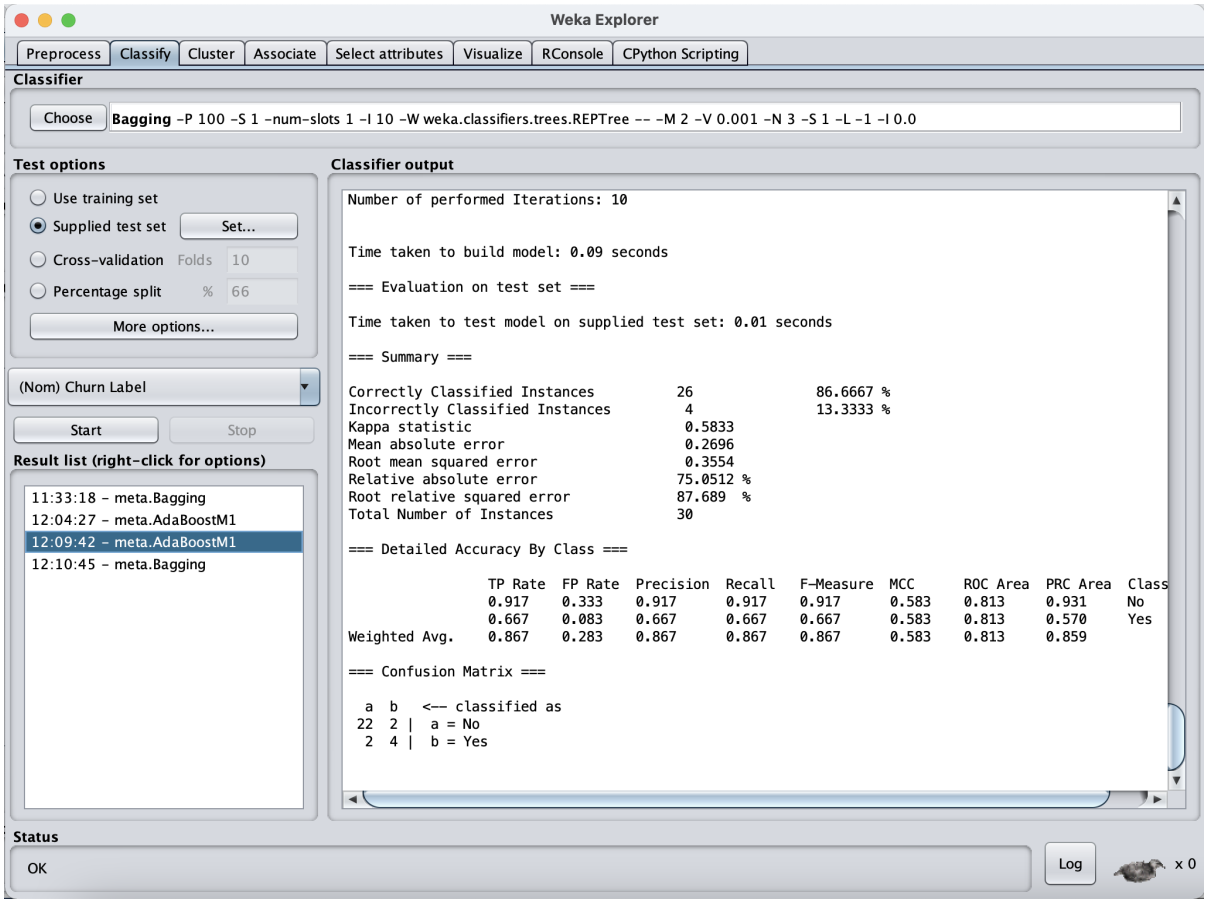
Based on the *Accuracy, RMSE, and TPR rate*, the best model is **Logistic** Attribute using the **InfoGainAttributeEval** attribute selection method.

Model Improvement

We tried to use **Bagging and Boosting with Logistic** (the best classifier model) and **InfoGainAttributeEval** (the best attribute selection method) as part of model improvement.

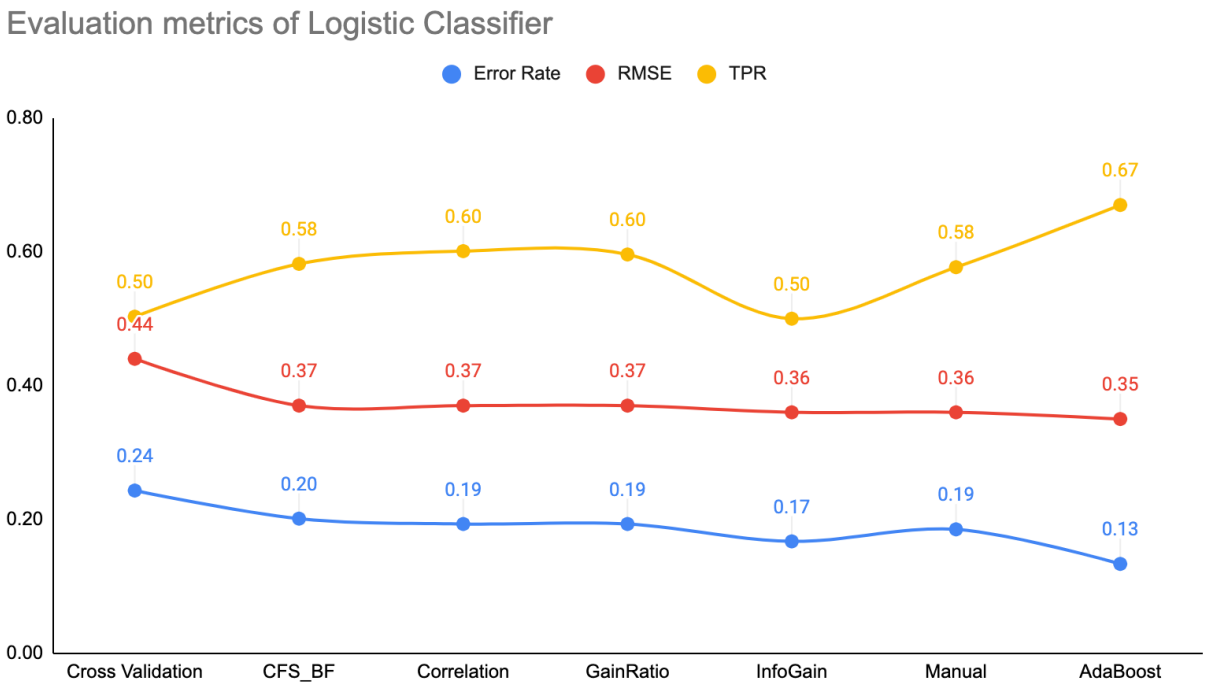
- Classification Algorithm: Bagging with Logistic Attribute Selector: InfoGainAttributeEval
- Classification Algorithm: AdaBoost with Logistic Attribute Selector: InfoGainAttributeEval

We didn't notice any significant improvement in our model.
Hence tried Adaboost with the default model - Decision Stump, and this provided better accuracy and TPR.



Following are the performance measures with Model Improvement
Accuracy = 86.67%
RMSE = 0.35
TPR(class = 'Yes') = 0.67

All the above measures are better than the best model.



The Best Model is significantly better than the 10 fold cross-validation using the same classification algorithm with all the attributes for accuracy and RMSE. But no improvement is seen for TPR.

Summary

To summarize, the ***Logistic Classifier*** proved to be a much better model in predicting the churn rate, which predicted customer churn with an **average accuracy of 81.21%**. However, using **AdaBoost**, we were able to **improve** the accuracy by another **5.4pp.**

Future Work

Unpredictability and risk are the major issues of any predictive model. Therefore, it is always good to build a probability score in the real world besides an absolute predicted outcome (churn= 'Yes' or churn = 'No'). This way, we classify the existing customers as high-risk (>80%), medium-risk (60-80%), or low-risk(<60%) based on the probability score. This way, the companies can focus on the customers upfront.

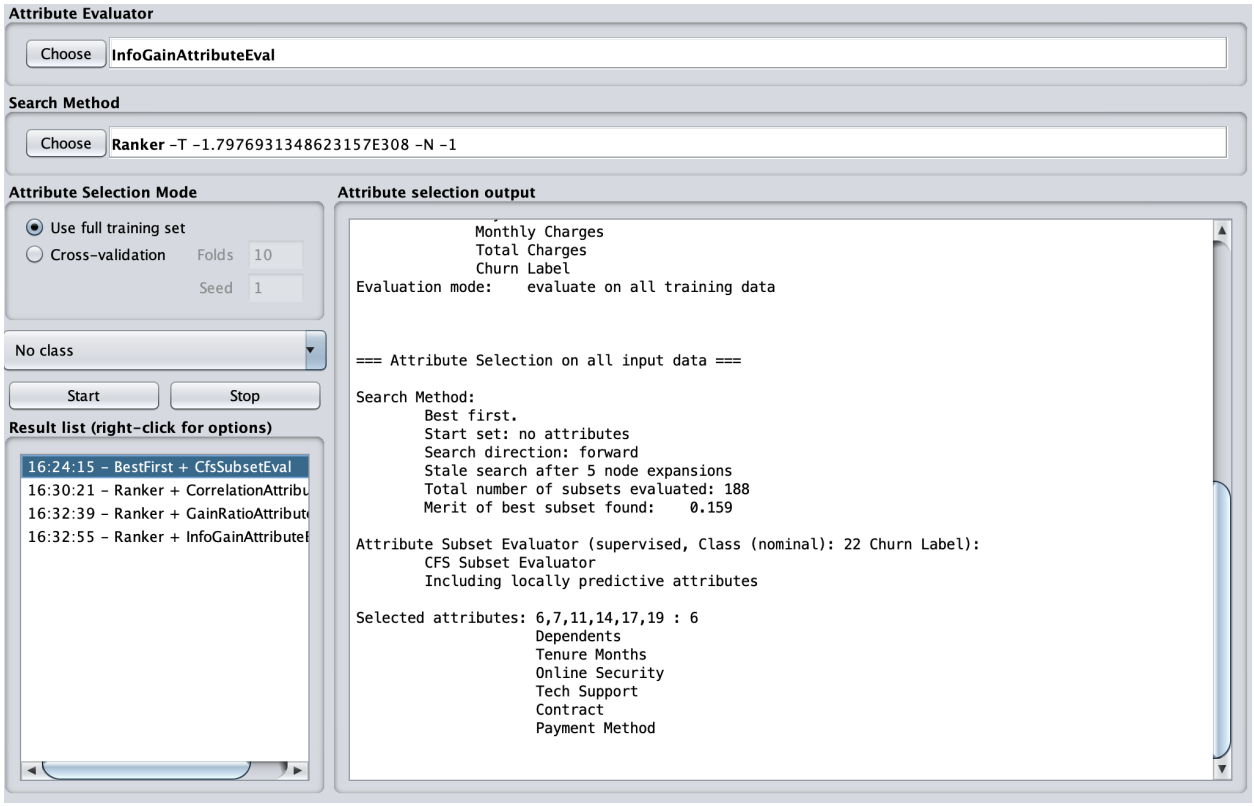
Individual contribution

Parvathy Sukumaran- Worked on building Classification Models, Model Evaluation, and Model Improvement.
Sahil Khanna - Worked on Data Pre-processing, EDA, and Attribute Selection.
Together - Worked on end-to-end development of the project and report.

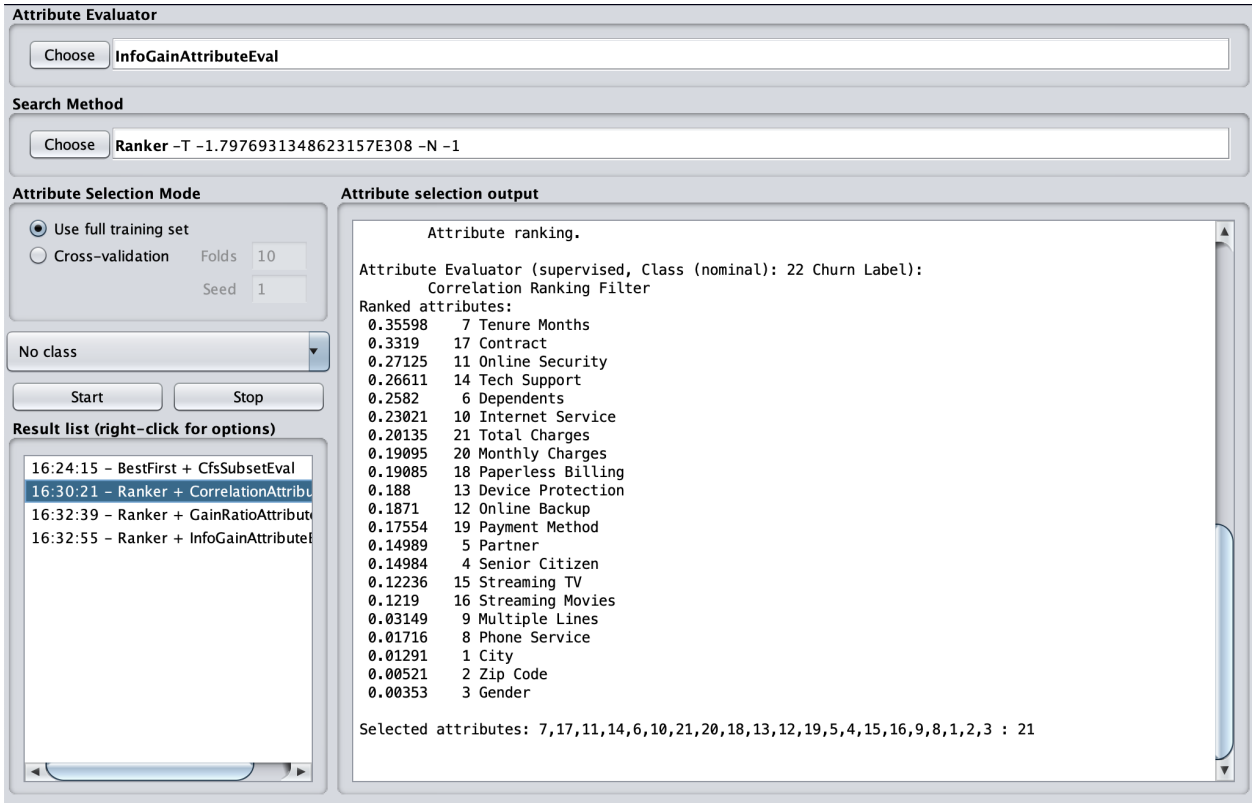
Appendix

Attribute Selection Methods (Screenshots)

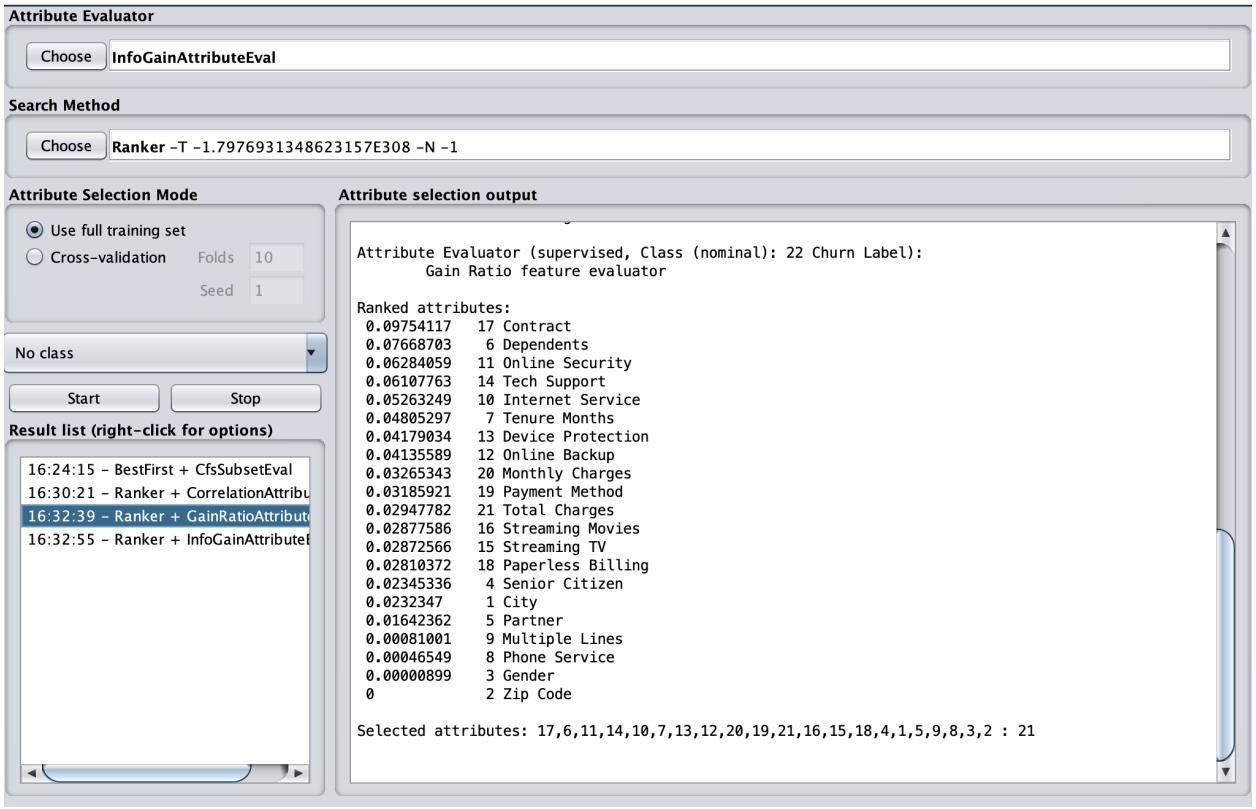
CfsSubsetEval with BestFirst



CorrelationAttributeEval with Ranker



GainRatioAttributeEval with Ranker



InfoGainAttributeEval with Ranker