DATA MINING

Airline Delay Prediction

Mohit Mohit
Abhishek Rai Sharma

Table of Contents

1 Introduction

Delay is painful. We get annoyed when the flights are delayed. However, when we know that the flight will be late, it easy to be at ease. This project aims to make a prediction on the basis of the data collected from Bureau of Transportation Statistics, U.S. Department of Transportation [**https://www.transtats.bts.gov/Fields.asp?Table_ID=236].** To achieve classification, we will use JMP Pro to process our data and Weka to perform the data mining classification and attribute selection algorithms.

2 Description of Dataset

2.1 Data Description

The original data set contains information for all commercial flights in the US from 1987 to 2018. Since the data set is extremely large, we will extract a reasonable subset of the data.

As the data size is very large, we have decided to analyze only the 2018 data (which itself is around 2GB). The detailed schema and proper understanding of the attributes are given below.

SYS_FIELD_NAME	FIELD_DESC
YEAR	Year
FL_DATE	Flight Date (yyyymmdd)
OP_UNIQUE_CARRIER	Unique Carrier Code.
OP_CARRIER_FL_NUM	Flight Number
ORIGIN_AIRPORT_ID	Origin Airport, Airport ID.
ORIGIN_CITY_NAME	Origin Airport, City Name
ORIGIN_STATE_ABR	Origin Airport, State Code
DEST_AIRPORT_ID	Destination Airport, Airport ID.
DEST_CITY_NAME	Destination Airport, City Name
DEST_STATE_ABR	Destination Airport, State Code
CRS_DEP_TIME	CRS Departure Time (local time: hhmm)
DEP_TIME	Actual Departure Time (local time: hhmm)
DEP_DELAY_NEW	Difference in minutes between scheduled and actual departure time. Early departures set to 0.
CRS_ARR_TIME	CRS Arrival Time (local time: hhmm)
ARR_TIME	Actual Arrival Time (local time: hhmm)
ARR_DELAY_NEW	Difference in minutes between scheduled and actual arrival time. Early arrivals set to 0.
CANCELLED	Cancelled Flight Indicator (1=Yes)

CANCELLATION_CODE	Specifies The Reason For Cancellation
DIVERTED	Diverted Flight Indicator (1=Yes)
CRS_ELAPSED_TIME	CRS Elapsed Time of Flight, in Minutes
ACTUAL_ELAPSED_TIME	Elapsed Time of Flight, in Minutes
AIR_TIME	Flight Time, in Minutes
FLIGHTS	Number of Flights
DISTANCE	Distance between airports (miles)
CARRIER_DELAY	Carrier Delay, in Minutes
WEATHER_DELAY	Weather Delay, in Minutes
NAS_DELAY	National Air System Delay, in Minutes
SECURITY_DELAY	Security Delay, in Minutes
LATE_AIRCRAFT_DELAY	Late Aircraft Delay, in Minutes
DIV_AIRPORT_LANDINGS	Number of Diverted Airport Landings

This dataset consists of 7076405 tuples and 30 attributes.

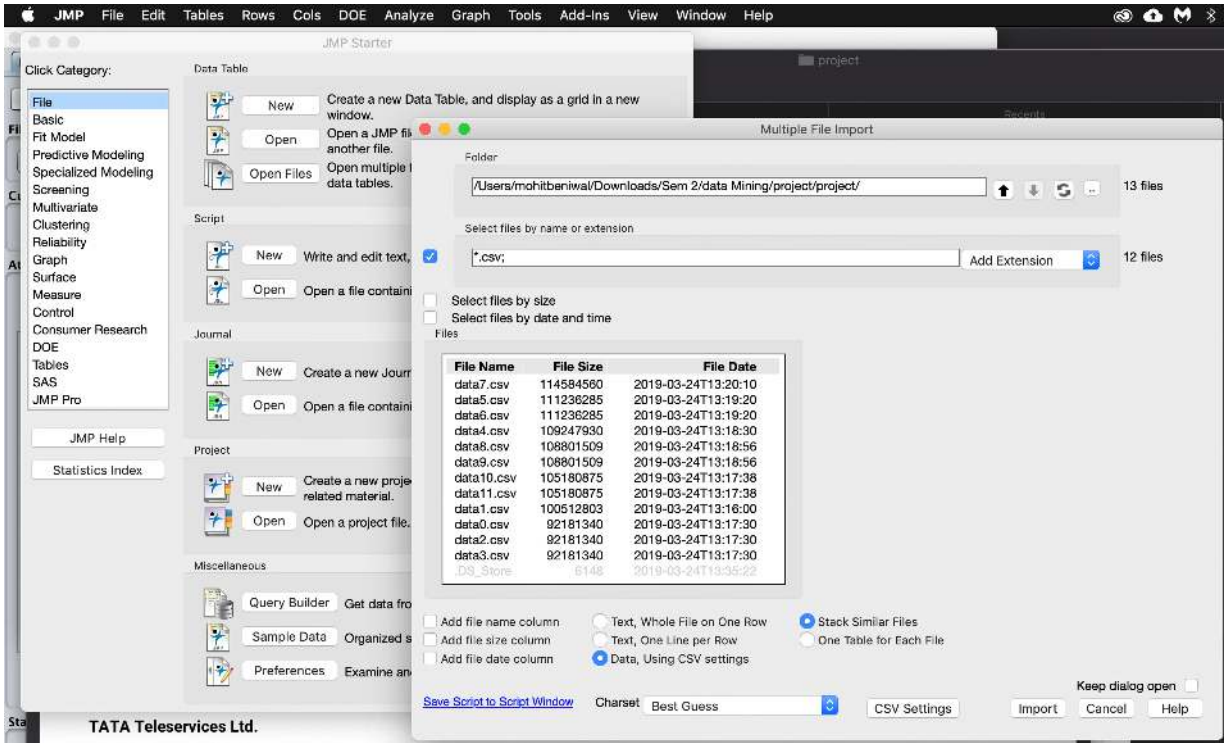
3 Goal

The goal of this project is to identify the factors which are most likely to cause flight delays. Moreover, we want to predict whether a flight will be delayed.

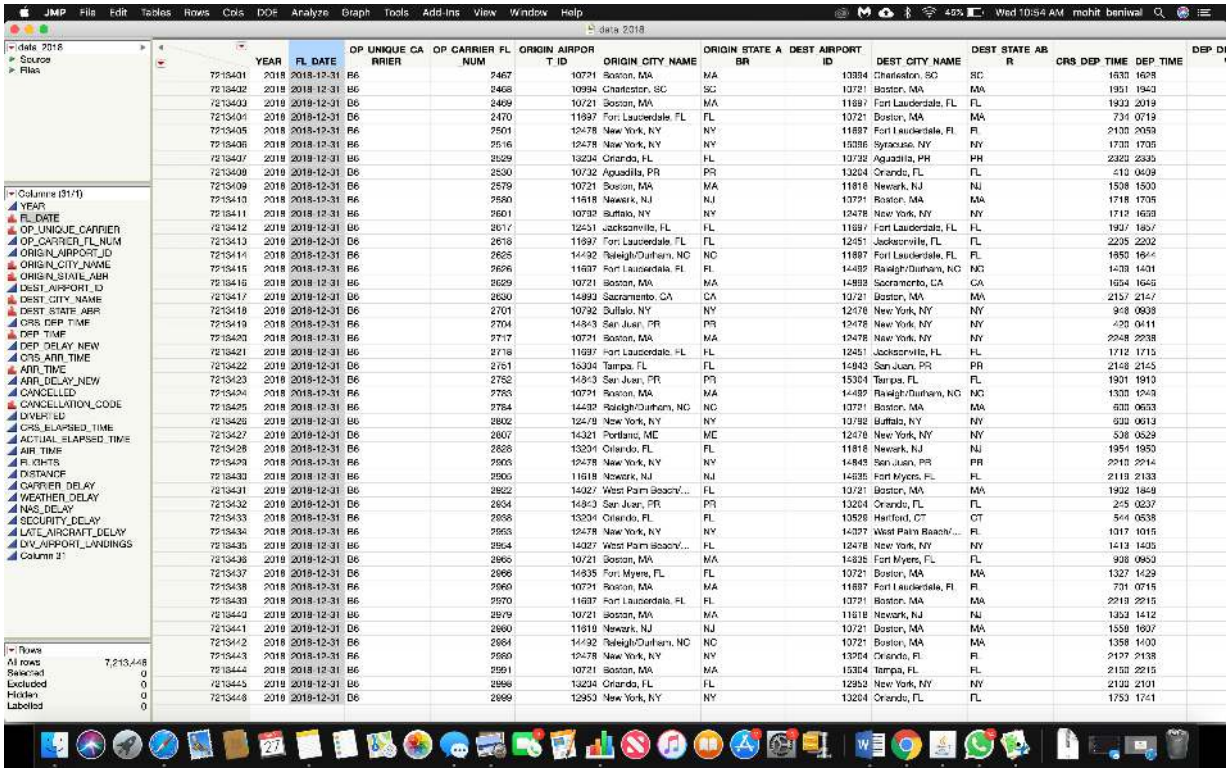
4 Data Manipulation in JMP Pro

As our dataset is huge and has multiple files to load from, we decided to use JMP PRO by looking at its efficiency to Handel large data. First, we renamed the downloaded data files (Jan-Dec) 2018 to data0.csv-data11.csv.

Process to load multiple files in JMP Pro:
Files -> open multiple ->
Set your directory and select extension as .csv



Now press Import to see the imported data from all 12 files into one single JMP table.

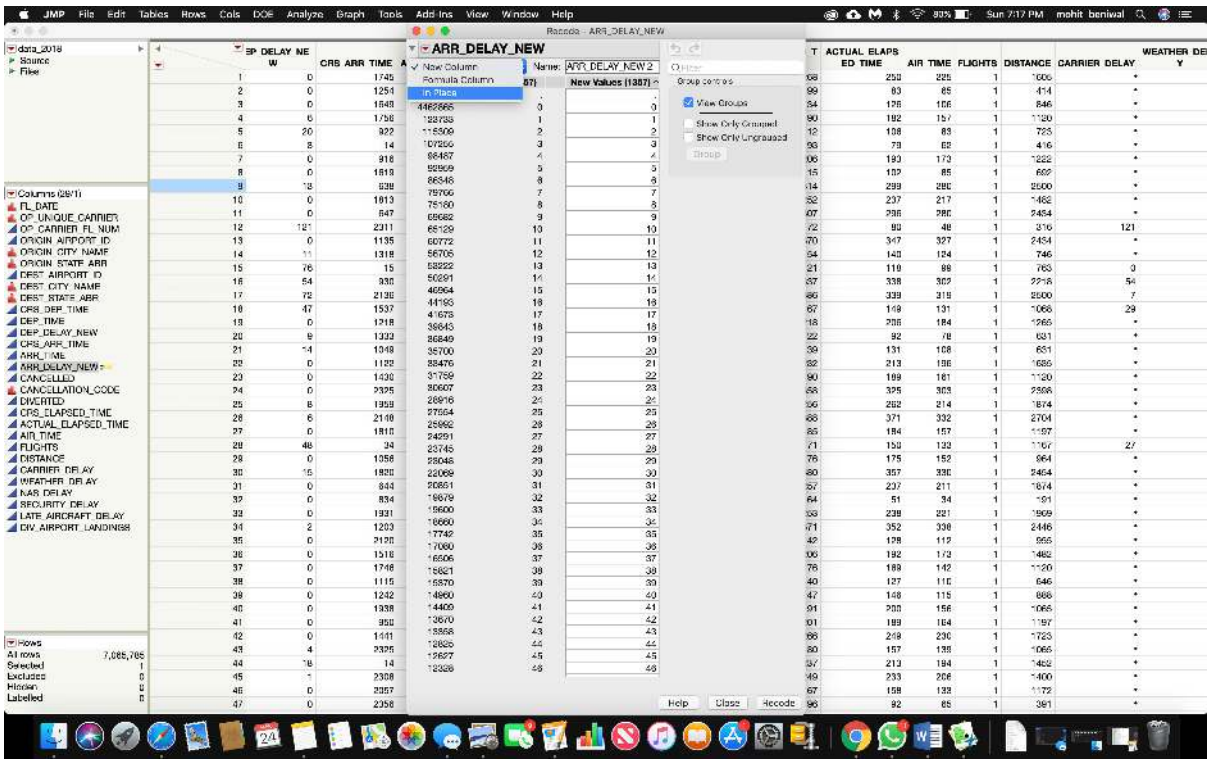


4.1 Creating the Class Attribute

We have chosen "ARR_DELAY_NEW" as our class attribute as mentioned in the project proposal. However, the attribute holds numeric continuous values, representing difference in minutes between scheduled and actual arrival time. Early arrivals set to 0.

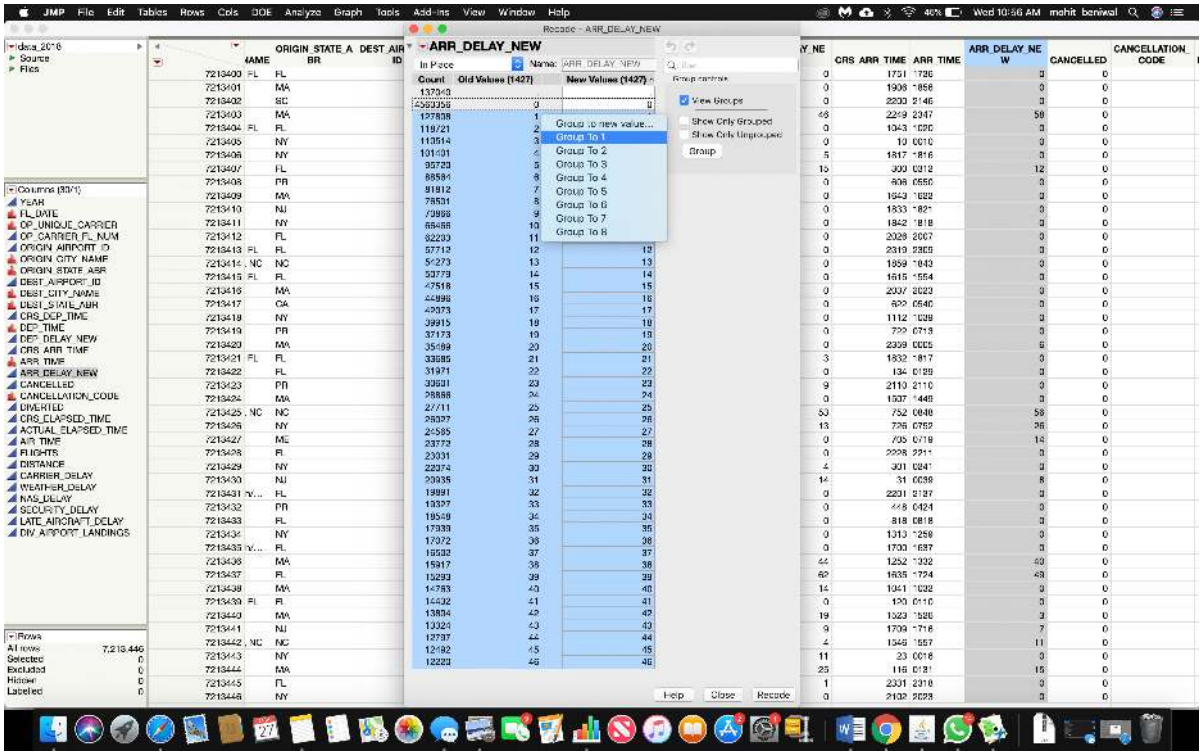
Hence, to represent weather a flight was delayed or not we will put all delay values in minutes to 1. So that 0 can represent that there was no delay and 1 can represent there was a delay. To do so,

Right click on "ARR_DELAY_NEW" -> Recode



Select "In place" from the dropdown so that changes will be done in the existing column, instead of making a new column.

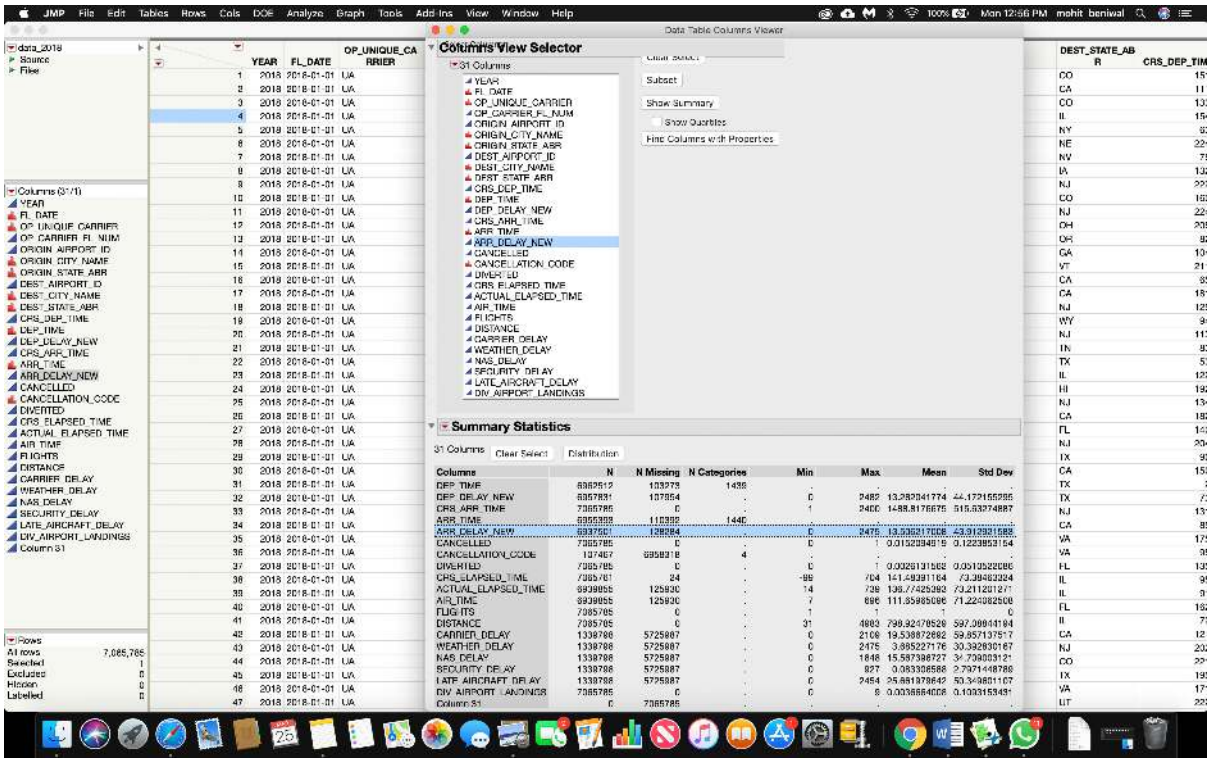
Now select all the values except '0' and '.' to group them as '1'.
We can see some tuples have missing values '.' which we will handle in the next step.



4.2 Data Analysis

In Column Viewer, we can check for missing values, mean, median, mode and quartile range etc. under show summary.

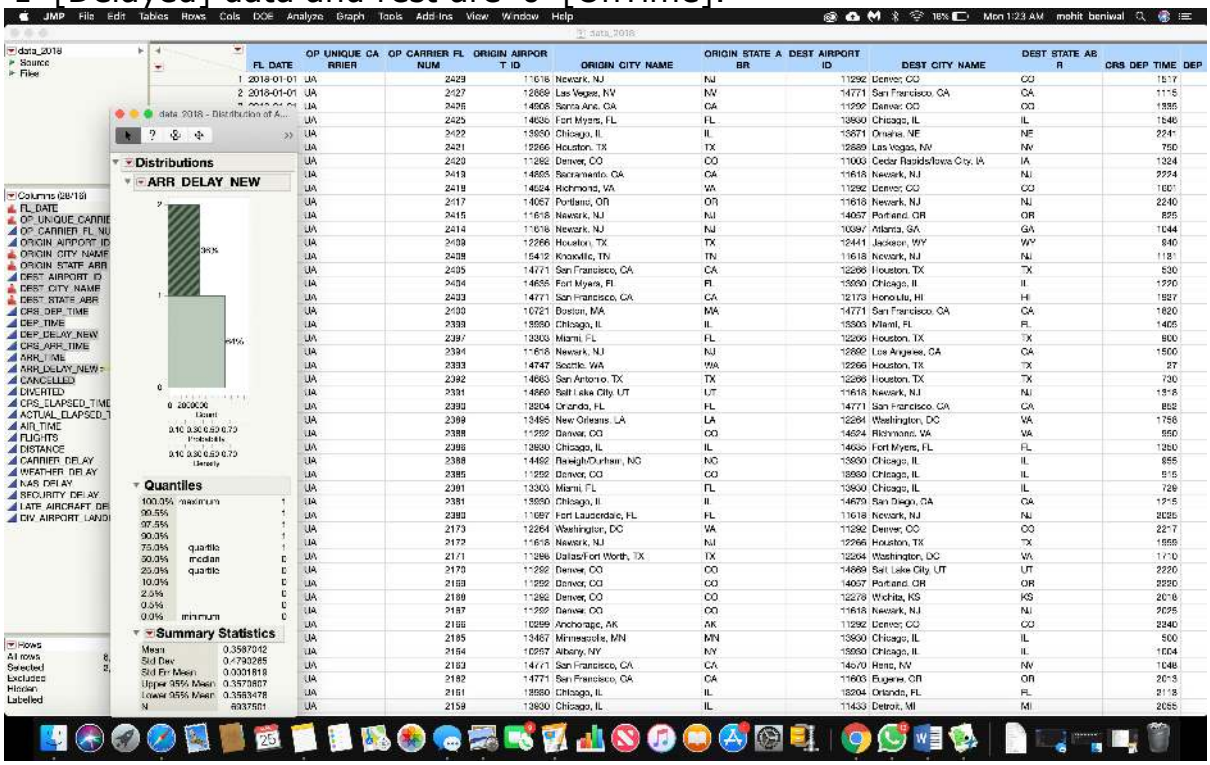
Cols -> Columns Viewer -> Select All Columns -> Click Show Summary



We can see that lots of attributes have missing values most importantly the “ARR_DELAY_NEW” (class attribute).
Therefore, we cannot decide whether they are delayed or not so we will delete those tuples in our data cleaning process.

4.3 Is our class imbalanced?

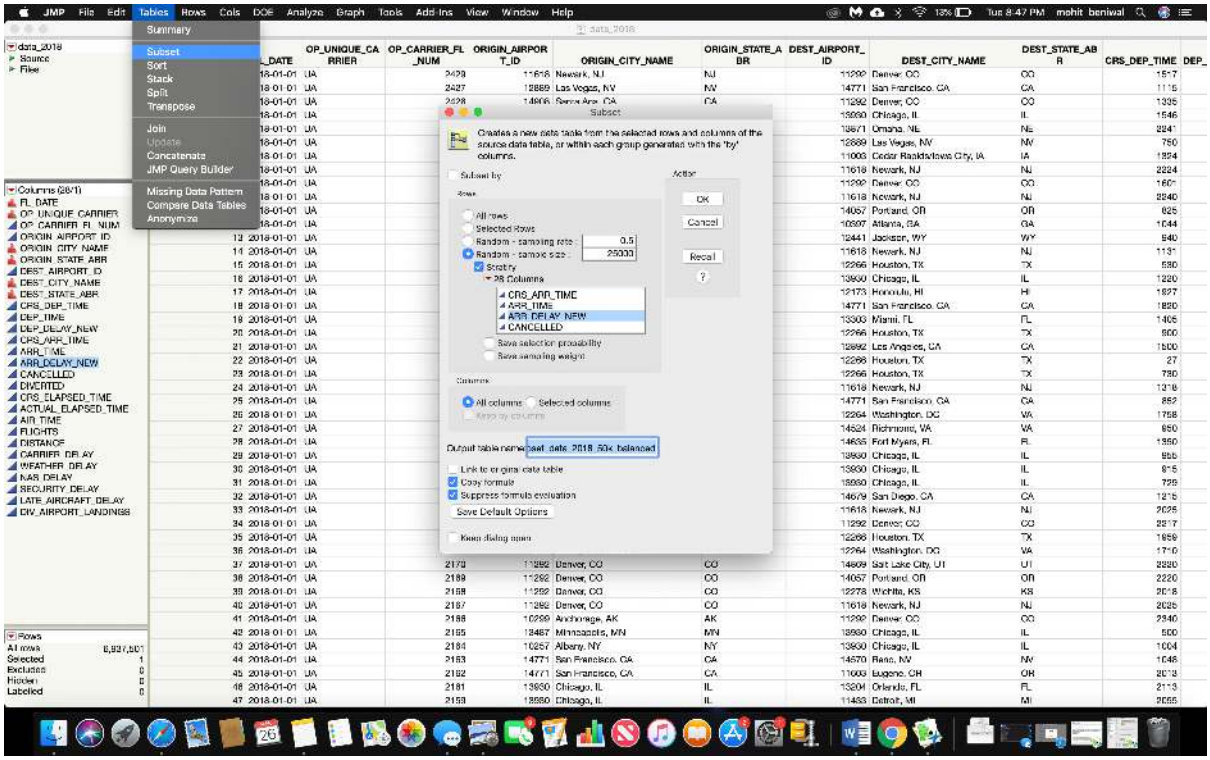
Yes, our class in imbalanced as we can see that we have only 36% of the ‘1’ [Delayed] data and rest are ‘0’ [OnTime].



4.4 Data Reduction and class balancing

To balance and reduce the number of tuples we decided to take a subset of the dataset in JMP Pro, as Weka was unable to perform this task. Therefore, we decided to take a sample of 50,000 tuples from the data. To do so, we followed the steps given below

Tables -> Subset
Select “Random sample size” as 25,000 -> check “Stratify” -> select the class attribute “ARR_DELAY_NEW” -> change “output table name” to “subset_data_2018_50k_balanced” -> Click “OK”

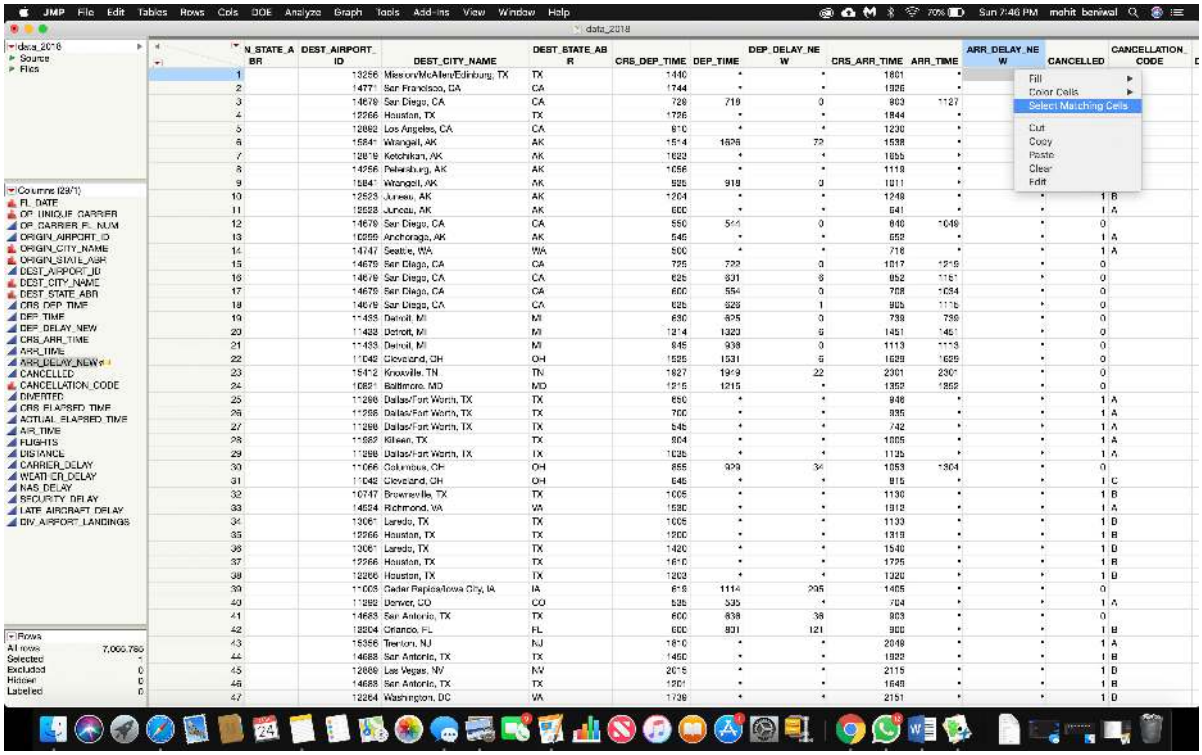


By doing this JMP will create a subset of data by removing the class imbalance while keeping the distribution of the data same. We have chosen 25,000 tuples because while removing the class imbalance JMP takes double the number of the rows as given. So, it will end up creating a subset of 50,000.
After generating this subset, we checked for the distribution of original and subset dataset and found them to be approximately to be the same.

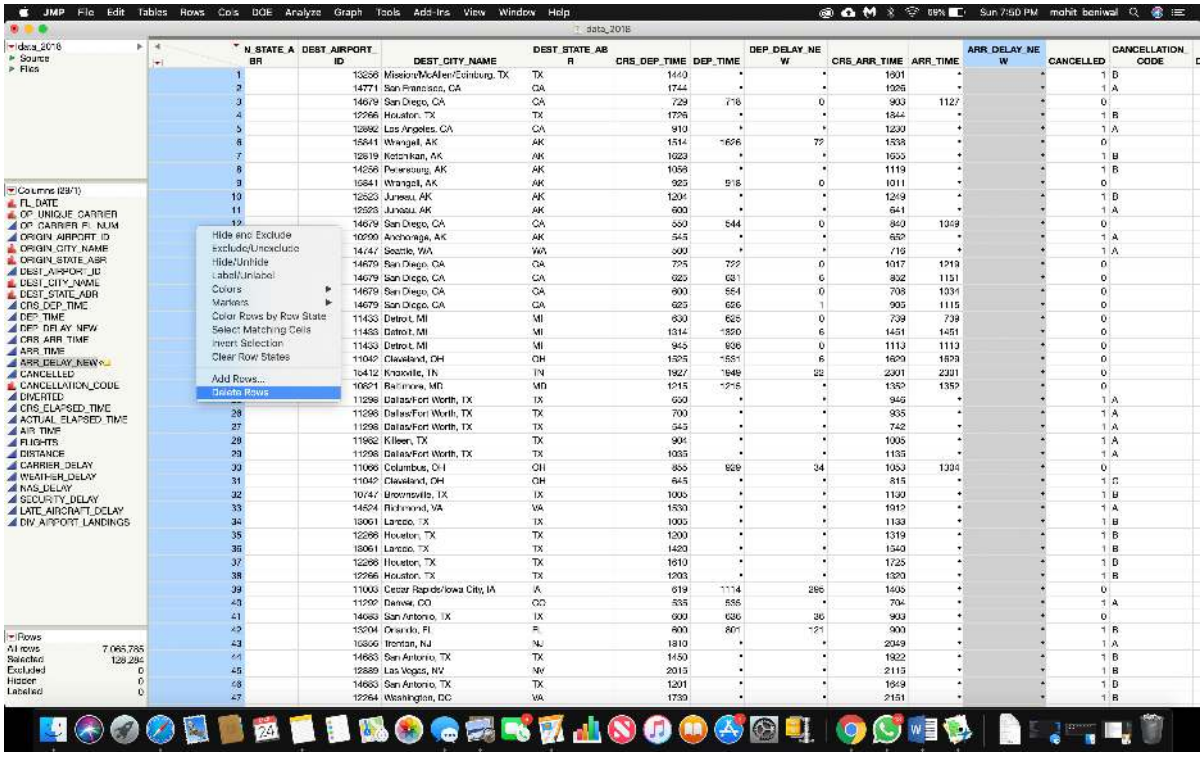
4.5 Data Cleaning

4.5.1 Cleaning data based on Class attribute

The first thing in cleaning the data will be removing tuples with unknown class attribute. So, we will sort our class attribute to find the missing values which is represented by '.'. Now scroll down in the ARR_DELAY_NEW column and right click on any '.' and choose "select matching cells" to select all the tuples with missing class attribute.

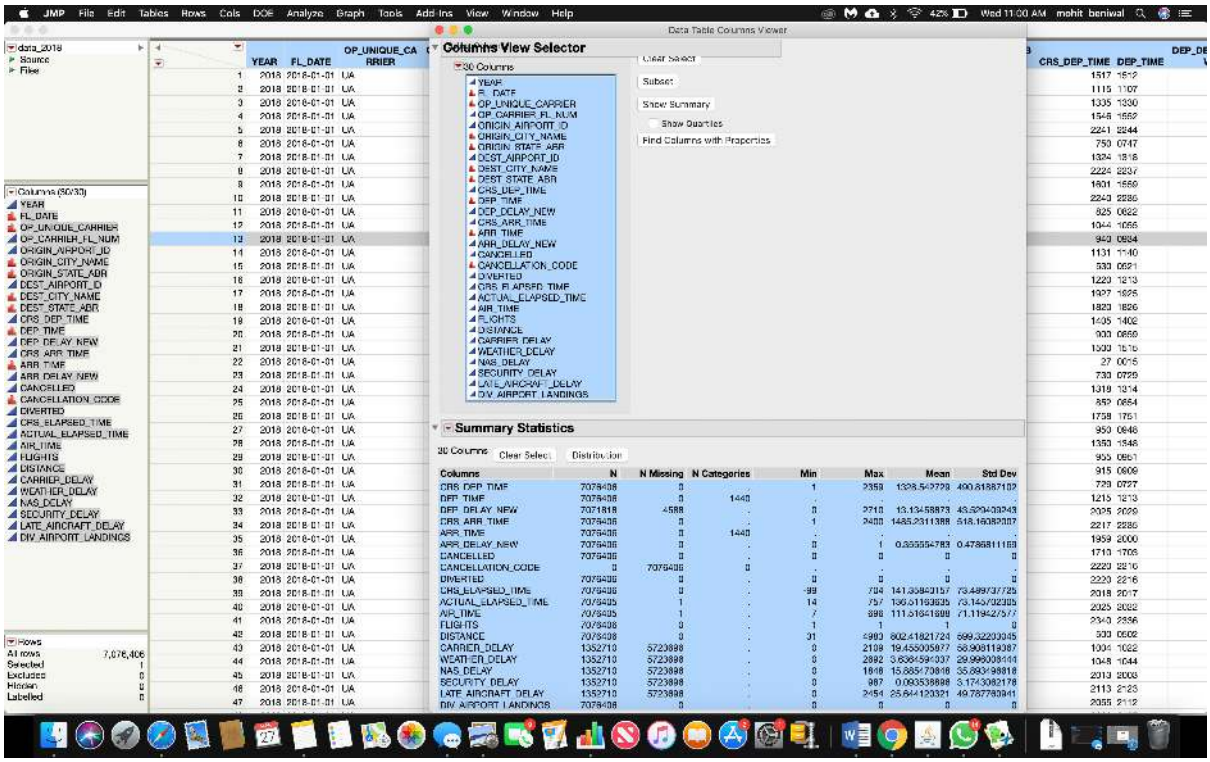


Now we can easily delete all unwanted tuples as shown below.



Again, open "Column Viewer" and now you can see that ACTUAL_ELAPSED_TIME and AIR_TIME has 1 missing value, so we will delete this tuple using the same process as we did for ARR_DELAY_NEW.

Also, attribute CANCELLATION_CODE has 0 category and all missing values. Hence, we will delete this column by
Right Click -> Delete Column

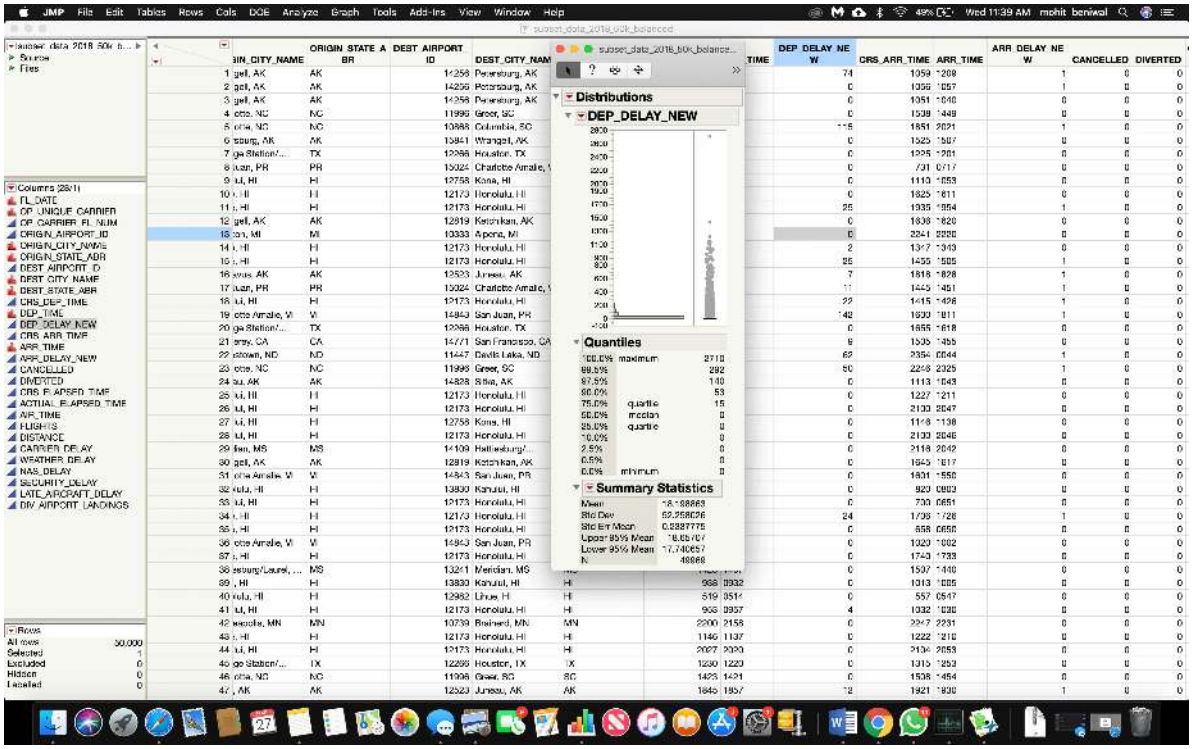


By looking at the data we realized that the YEAR column is not giving any added information as we already have FL_DATE column. So, we have decided to delete the YEAR column.
Right Click -> Delete Column

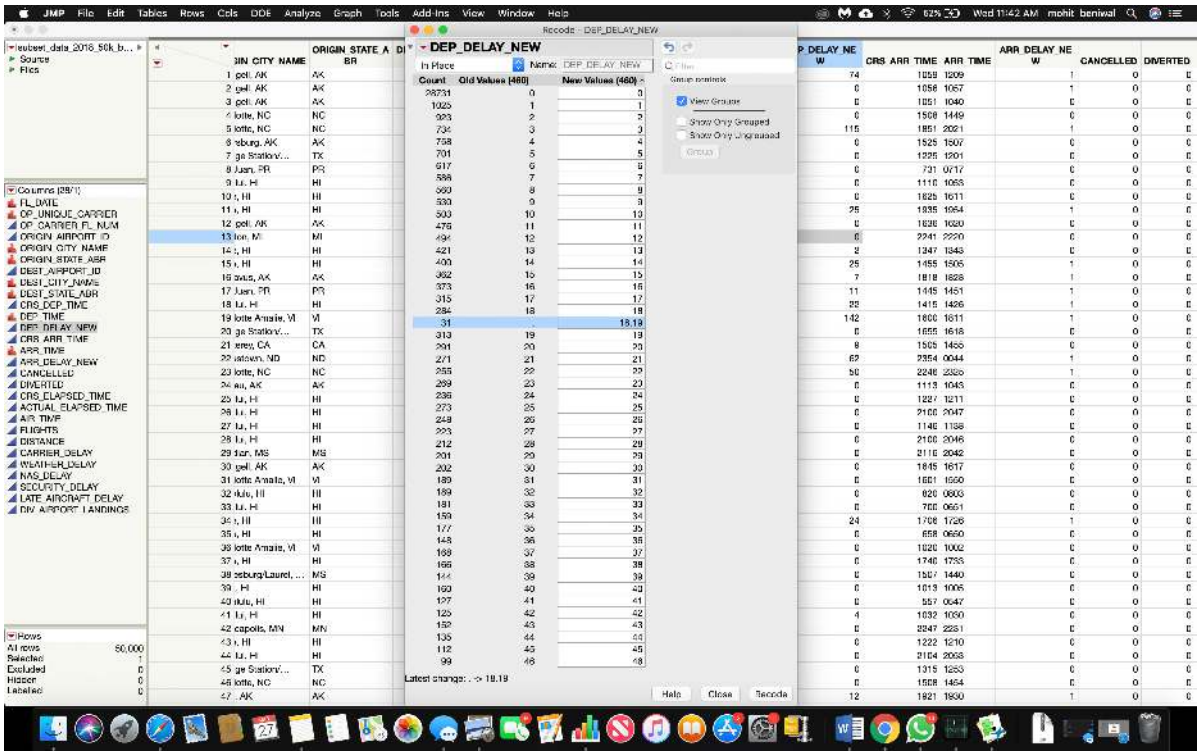
4.6 Handling Missing Data

After analyzing the dataset, we found that columns DEP_DELAY_NEW, CARRIER_*DELAY,* WEATHER_DELAY, NAS_DELAY, SECURITY_DELAY and LATE_AIRCRAFT_DELAY have a lot of missing values. Thus, for the missing values of DEP_DELAY_NEW, we will replace them by their mean because we can't set them to be 0 or something else, as they represent the magnitude of the delay. However, for the remaining columns, the tuples with missing values had no delay. Hence, we will replace them with 0.

For DEP_DELAY_NEW, we followed the process given below.



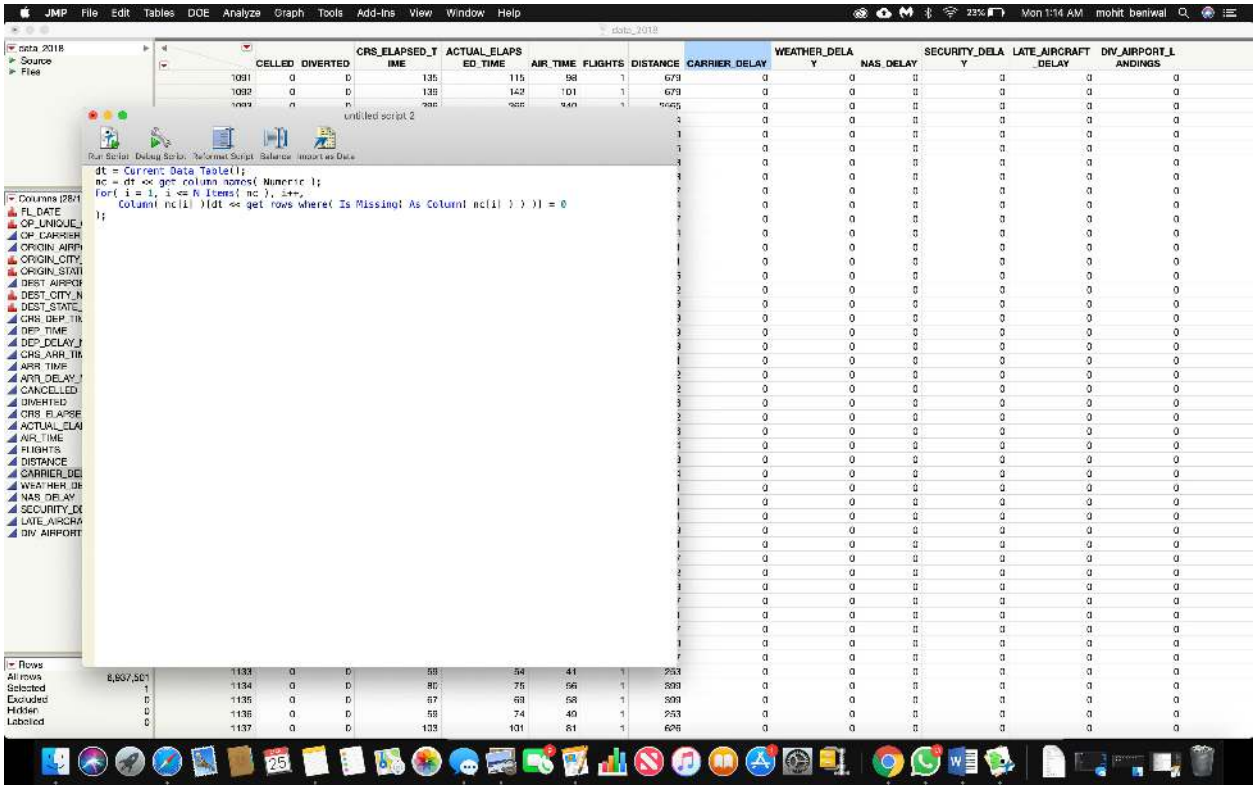
As we can see that the mean is 18.19, we will set the missing value with the mean.



For the rest of the columns with missing values in the dataset, we replaced them with '0' by using a script:

File -> New-> New Script -> Paste the given script

```
dt = Current Data Table();
nc = dt << get column names( Numeric );
For( i = 1, i <= N Items( nc ), i++,
        Column( nc[i] )[dt << get rows where( Is Missing( As Column( nc[i]
) ) )] = 0
);
```



Click Run.

4.7 Exporting data into csv

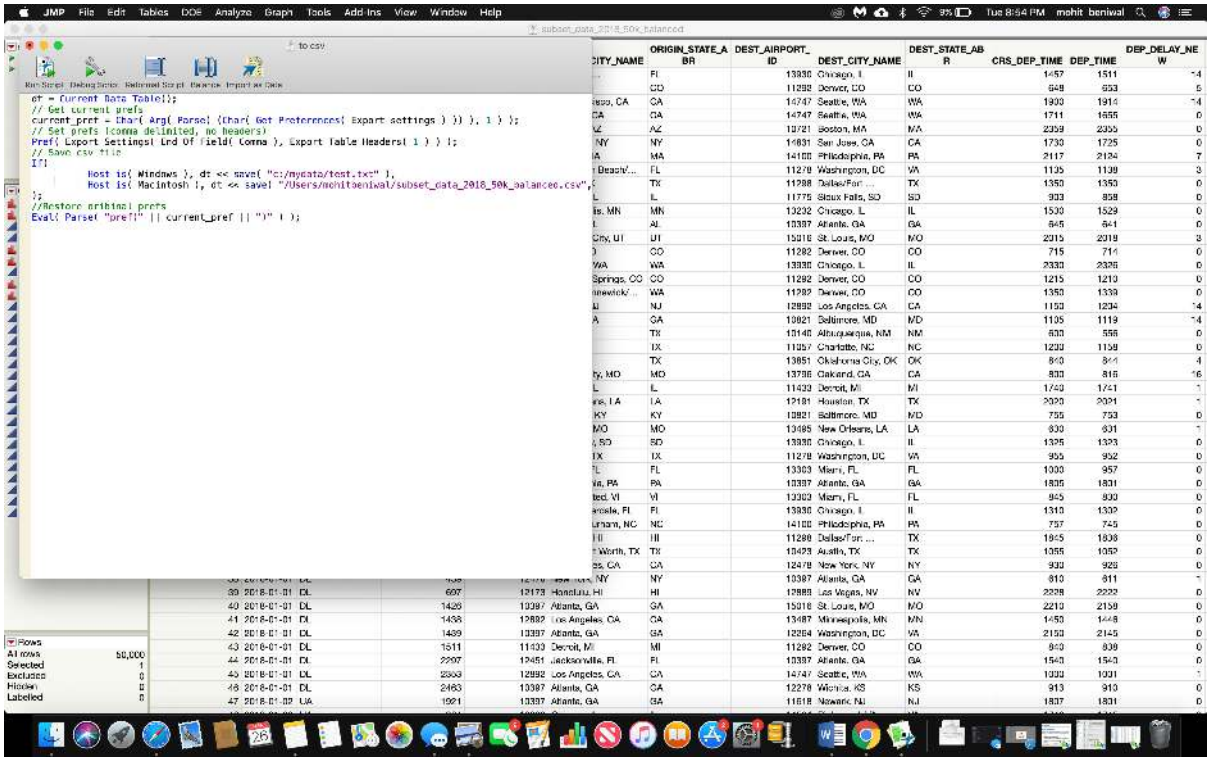
After all the cleaning, we exported the subset dataset by running a script.

File -> New-> New Script -> Paste the given script

Also, please specify the path you want to save the file and make sure that the specified directory is present else the script won't work.

```
dt = Current Data Table();
// Get current prefs
current_pref = Char( Arg( Parse( (Char( Get Preferences( Export settings
) )) ), 1 ) );
// Set prefs (comma delimited, no headers)
```

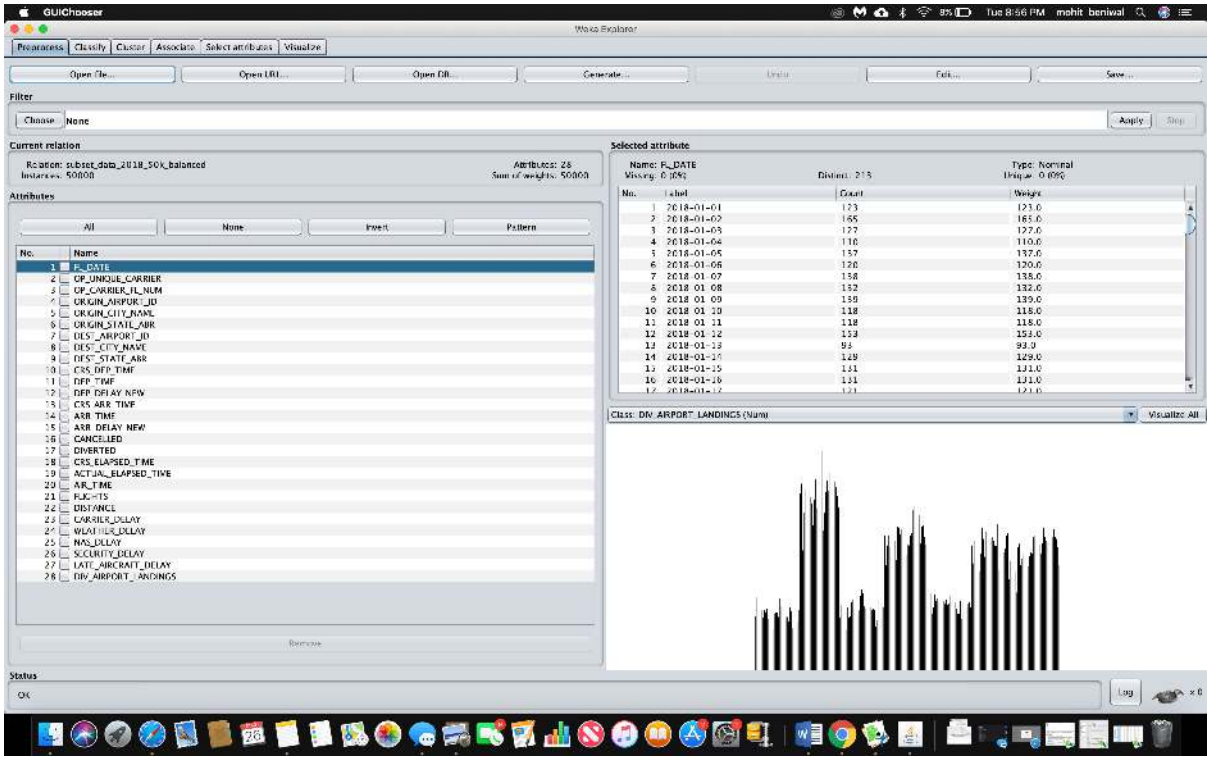
```
Pref( Export Settings( End Of Field( Comma ), Export Table Headers( 1 ) ) )
);
// Save csv file
If(
        Host is( Windows ), dt << save( "c:/mydata/test.txt" ),
        Host is( Macintosh ), dt << save(
"/Users/mohitbeniwal/subset_data_2018_50k_balanced.csv", text )
);
//Restore oribinal prefs
Eval( Parse( "pref(" || current_pref || ")" ) );
```



Click Run .

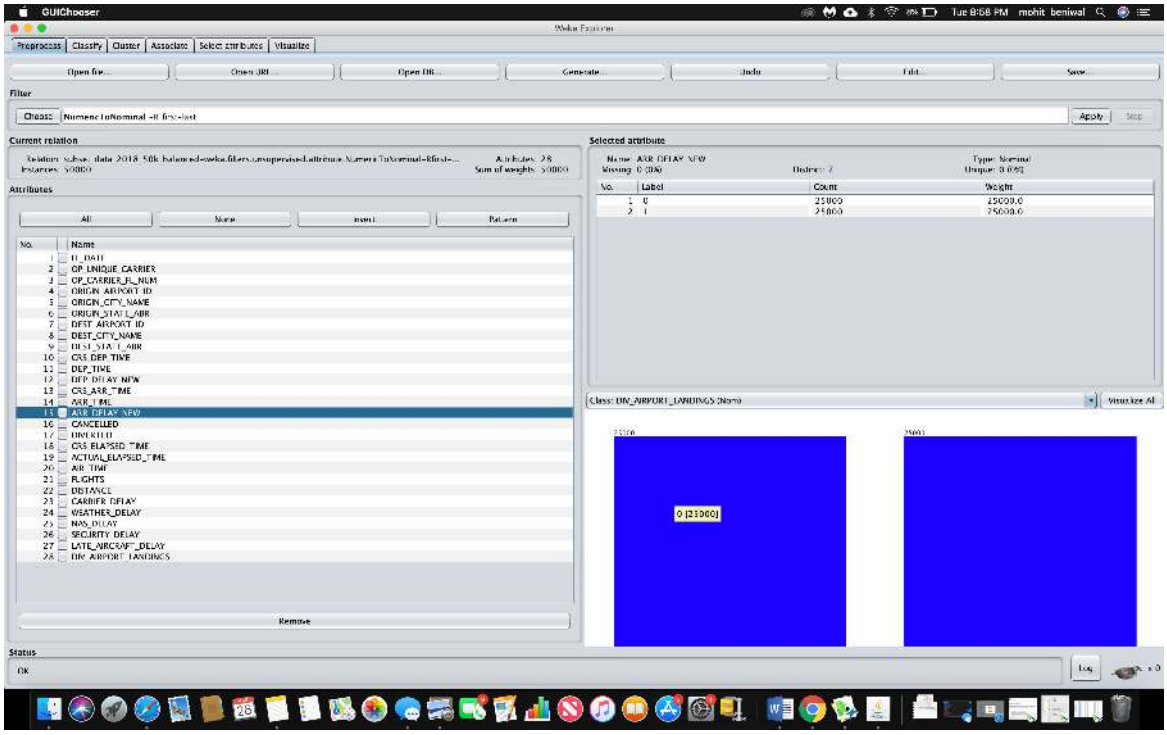
5 Data Loading into Weka

We imported the subset data with 50,000 tuples into Weka.

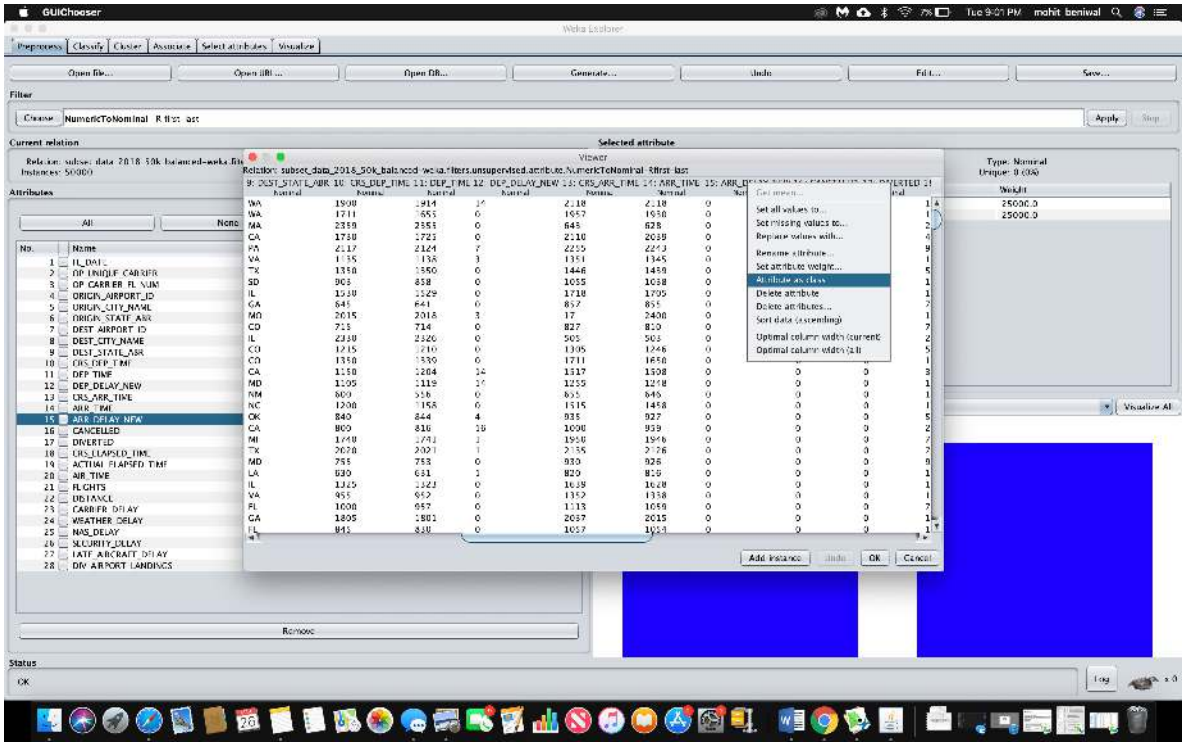


5.1 Specifying class attribute

As our class attribute has values ‘0’ and ‘1’, it is categorized as numeric. However, to run most of the classification algorithms, we have to make it nominal by selecting
Filter -> unsupervised -> attributes -> NumericToNominal as shown below.

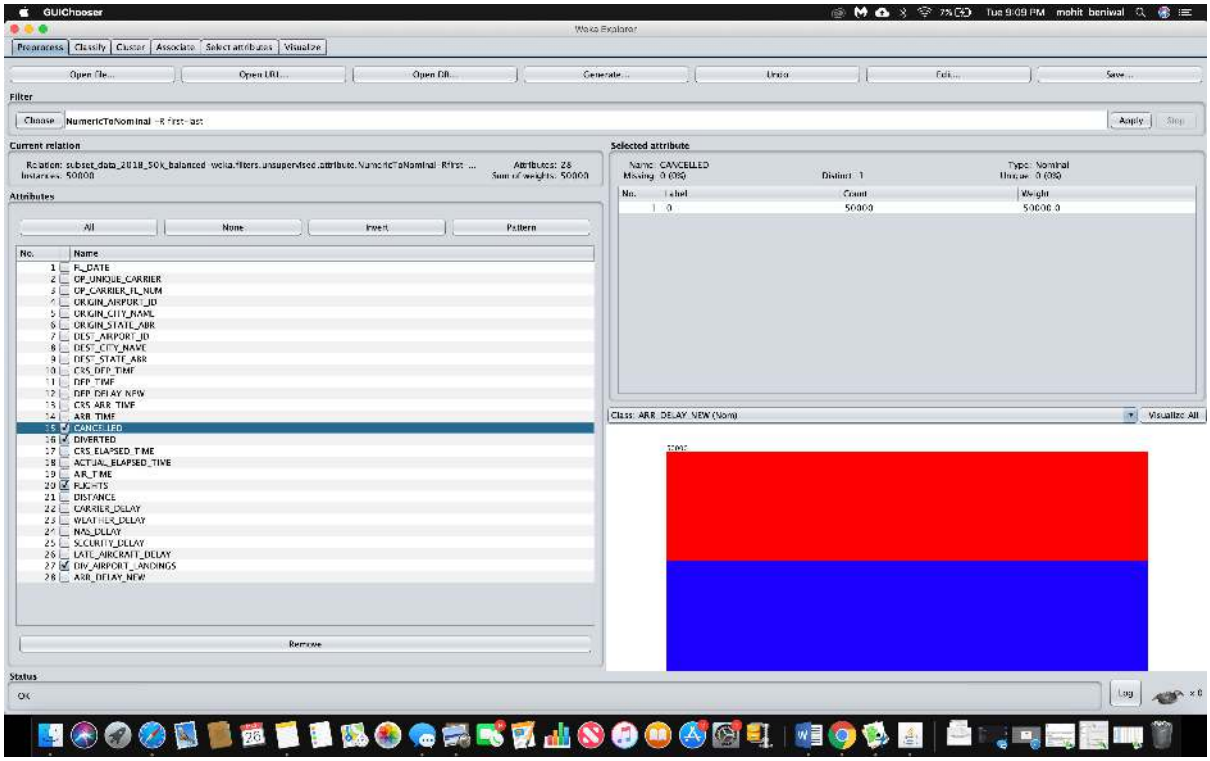


Now click on edit and make the “ARR_DELAY_NEW” as class attribute.



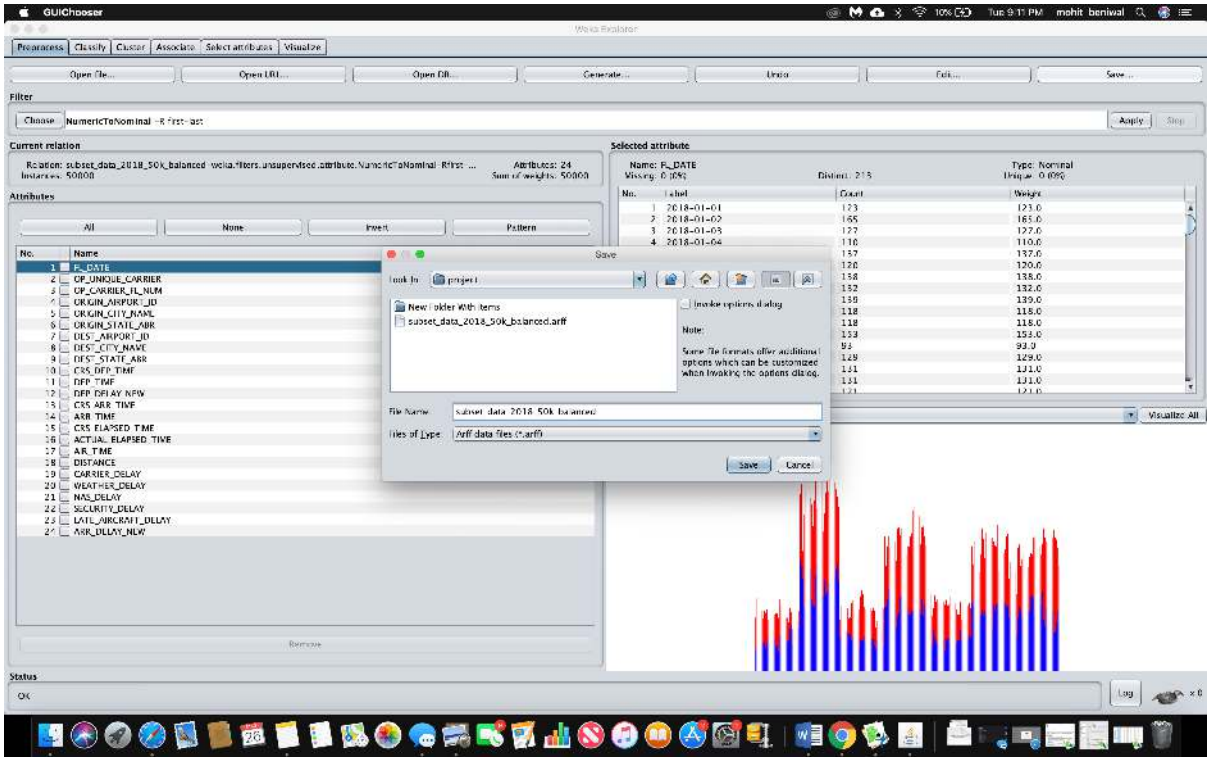
5.2 Data Wrangling

Now after looking through all the attributes we found that the CANCELLED, DIVERTED, FLIGHTS and DIV_AIRPORT_LANDINGS have only single values either ‘0’ or ’1’. Hence, we will delete them.



5.3 Saving the WEKA compatible file [.arff].

The csv file doesn't save the attributes header like nominal or numeric. It selects the default type on the basis of the data. Thus, we will save our dataset as .arff to maintain the column types.

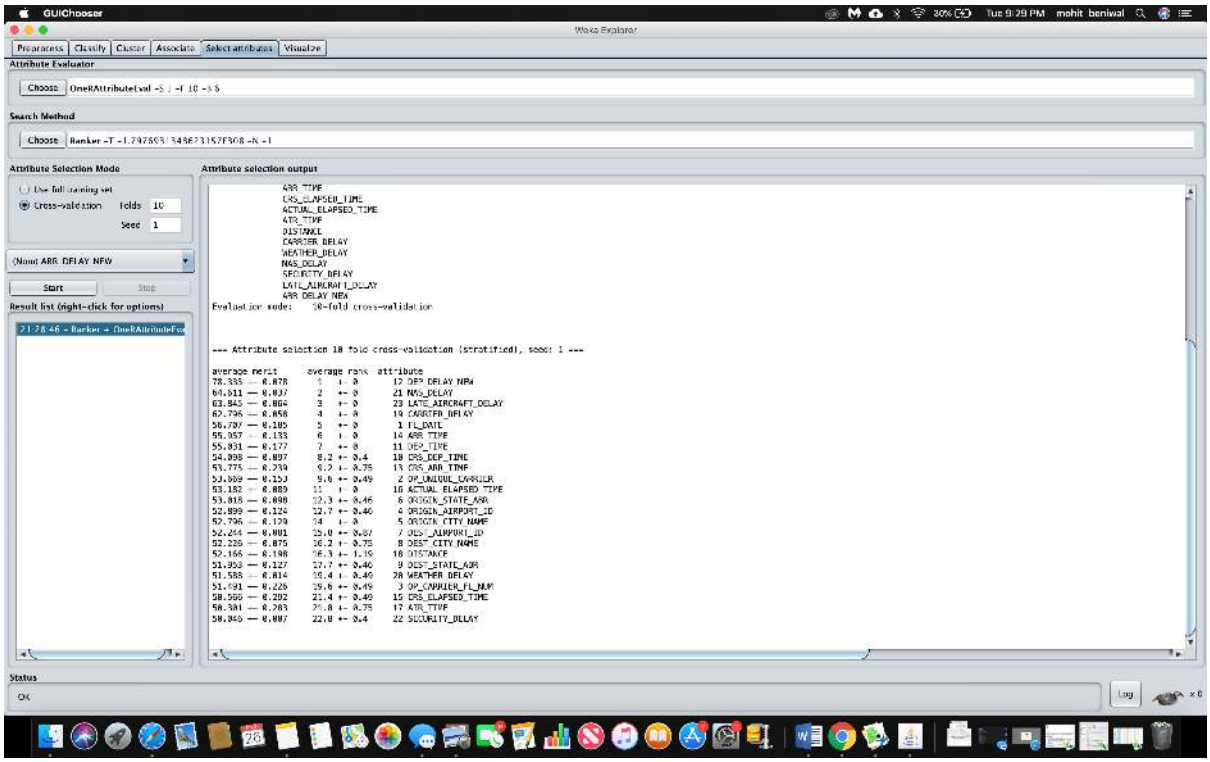


6 Attribute Selection

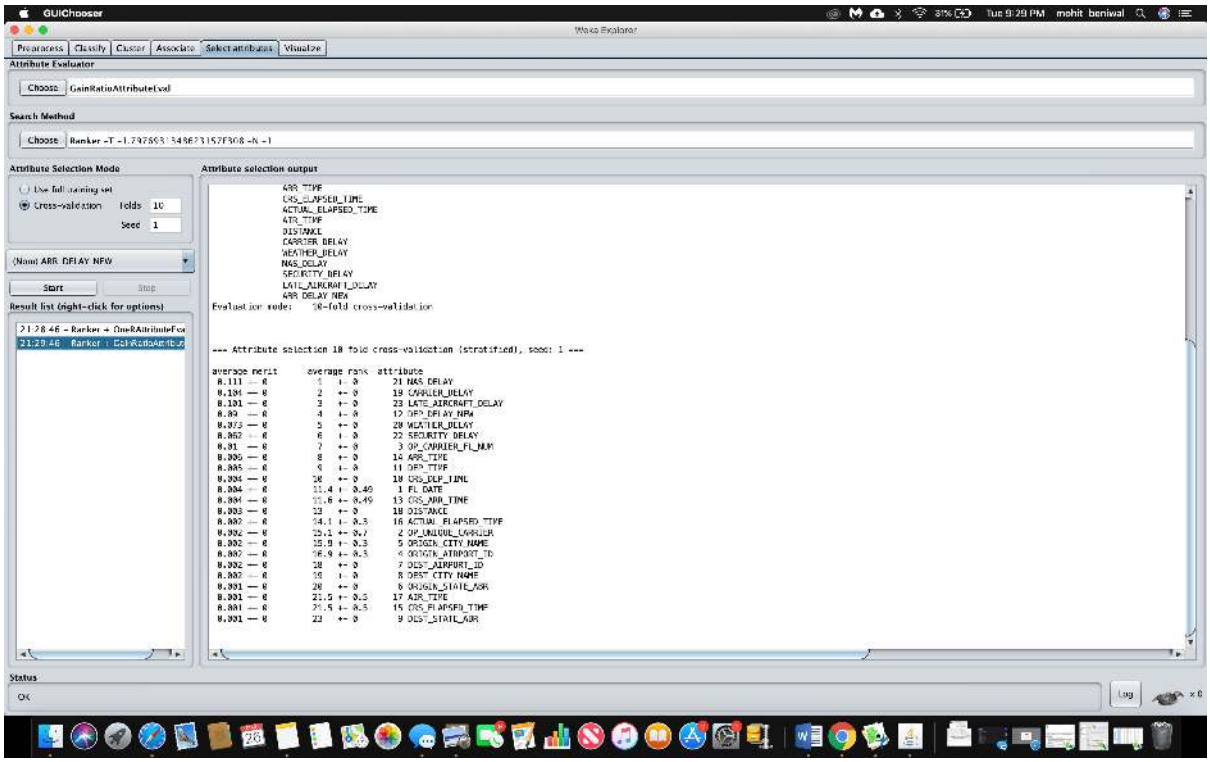
After running through all the available attribute selection algorithms, we chose these 5 evaluations to predict best set of pairs which would be considered for our model building. Here we chose Ranker Search method for the selection.

1. OneR Attribute Evaluation
2. Gain Ratio Attribute Evaluation
3. Classifier Attribute Evaluation
4. InfoGain Attribute Evaluation
5. SymmetricalUncert Attribute Evaluation

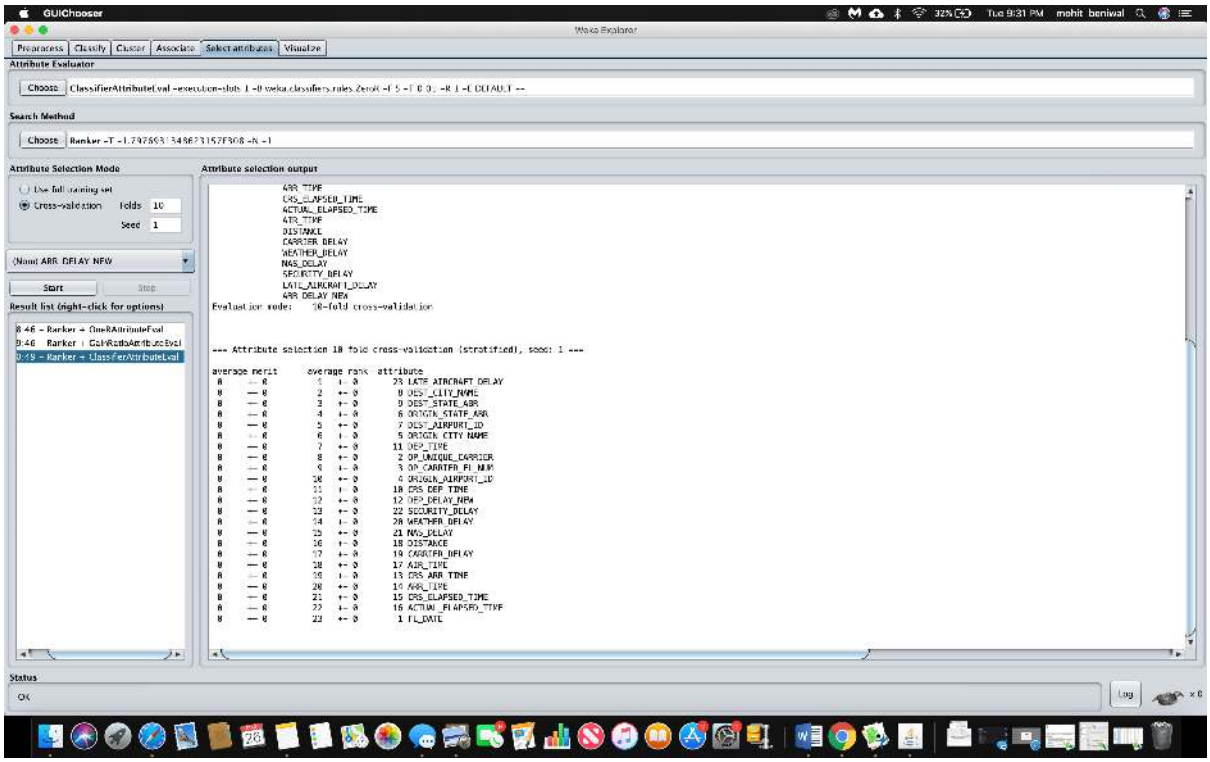
6.1 OneR Attribute Evaluation



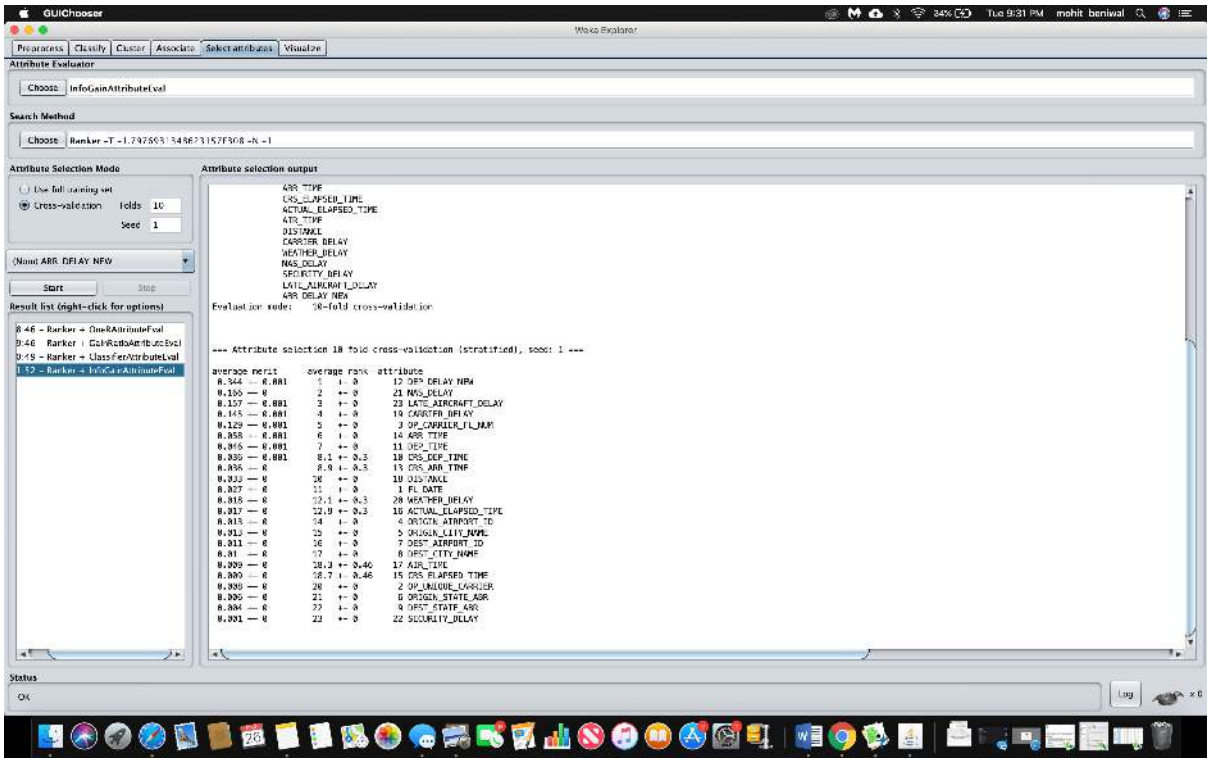
6.2 Gain Ratio Attribute Evaluation



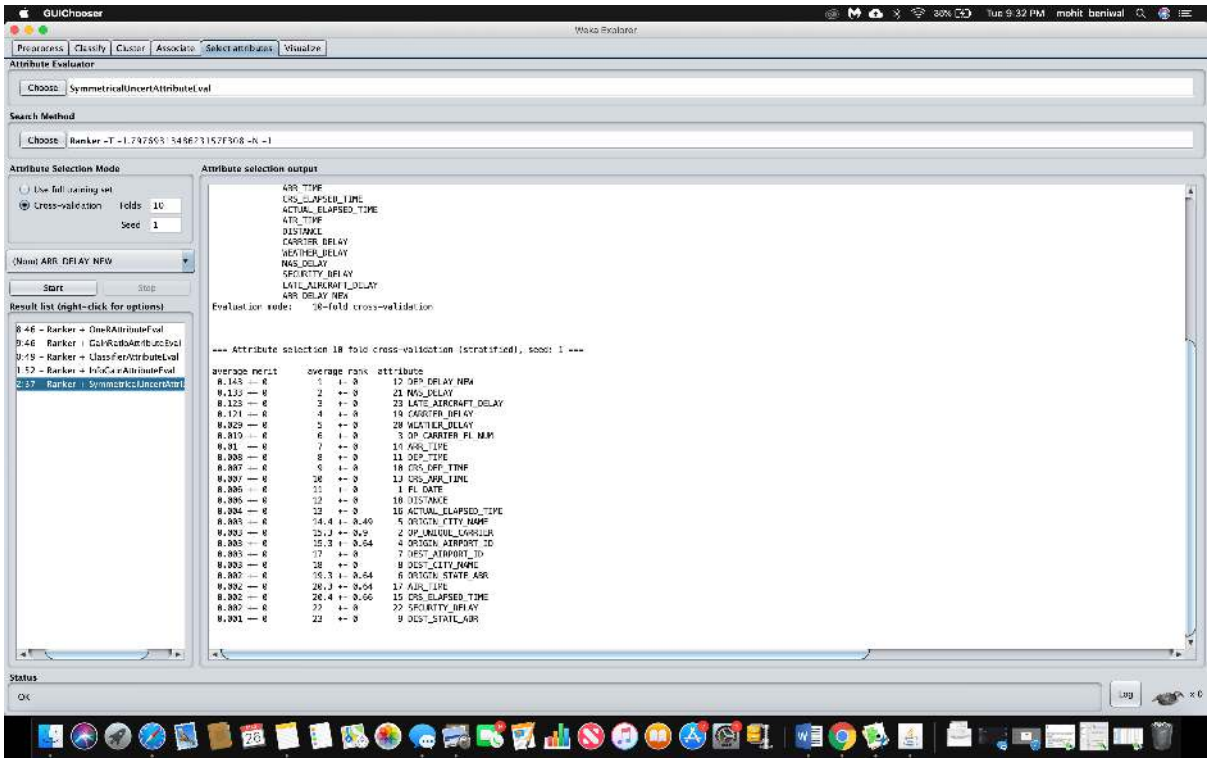
6.3 Classifier Attribute Evaluation



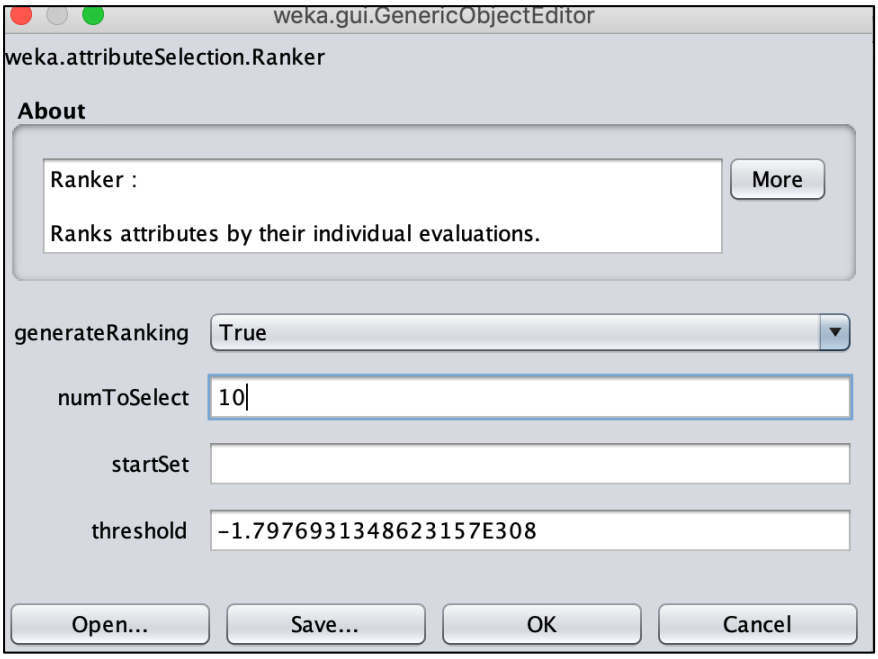
6.4 Info Gain Attribute Evaluation



6.5 SymmetricalUncert Attribute Evaluation



We will use only the Top 10 for dimension reduction. Using ranker method.



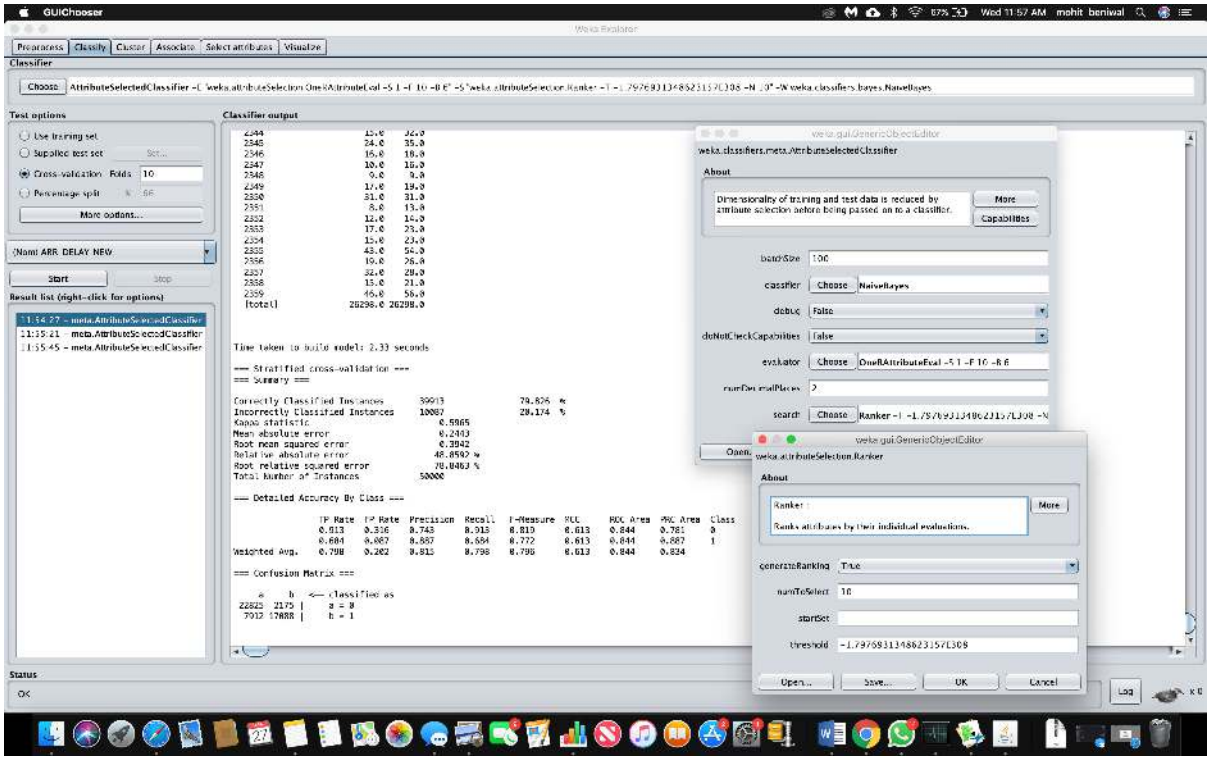
7 Classification

Classification is a data mining function that assigns items in a collection to target categories or classes. The goal of classification is to accurately predict the target class for each case in the data. For example, a classification model could be used to identify loan applicants as low, medium, or high credit risks.

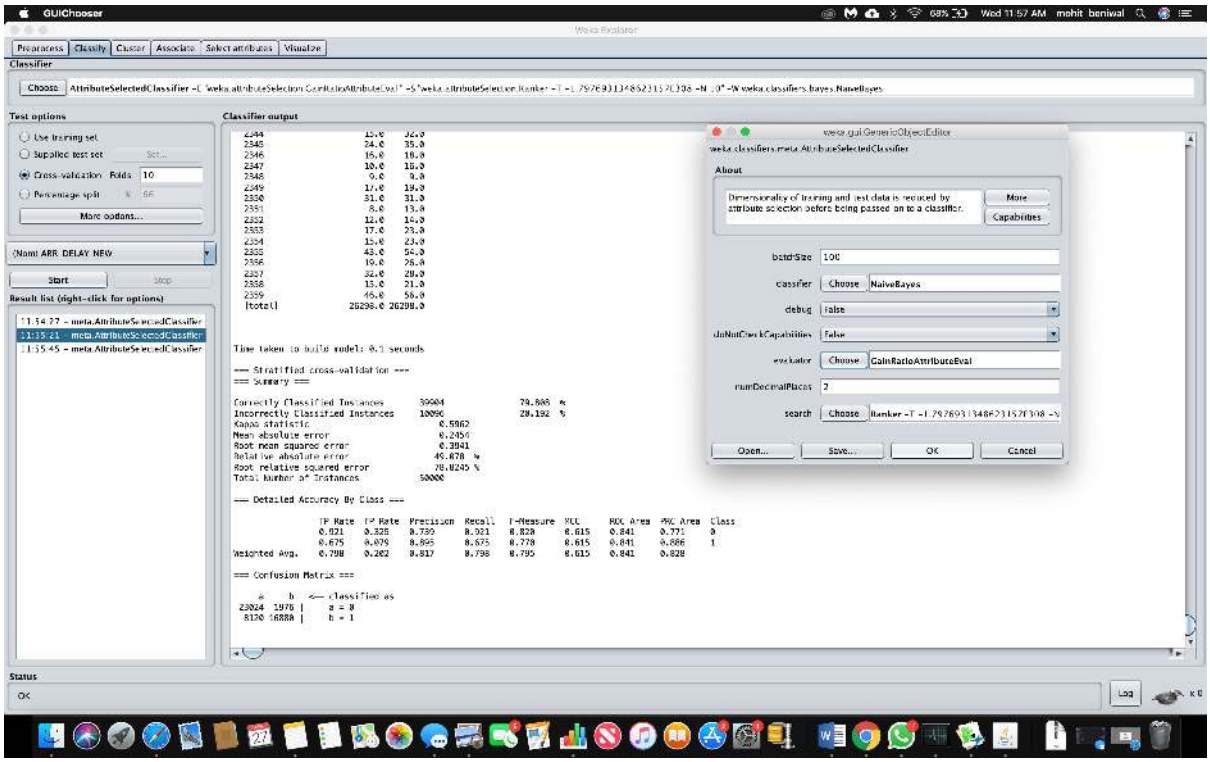
7.1 Naïve Bayes Algorithm

Naive Bayes is a simple, yet effective and commonly-used, machine learning classifier. It is a probabilistic classifier that makes classifications using the Maximum A Posteriori decision rule in a Bayesian setting. It can also be represented using a very simple Bayesian network. Naive Bayes classifiers have been especially popular for text classification and are a traditional solution for classification problems.

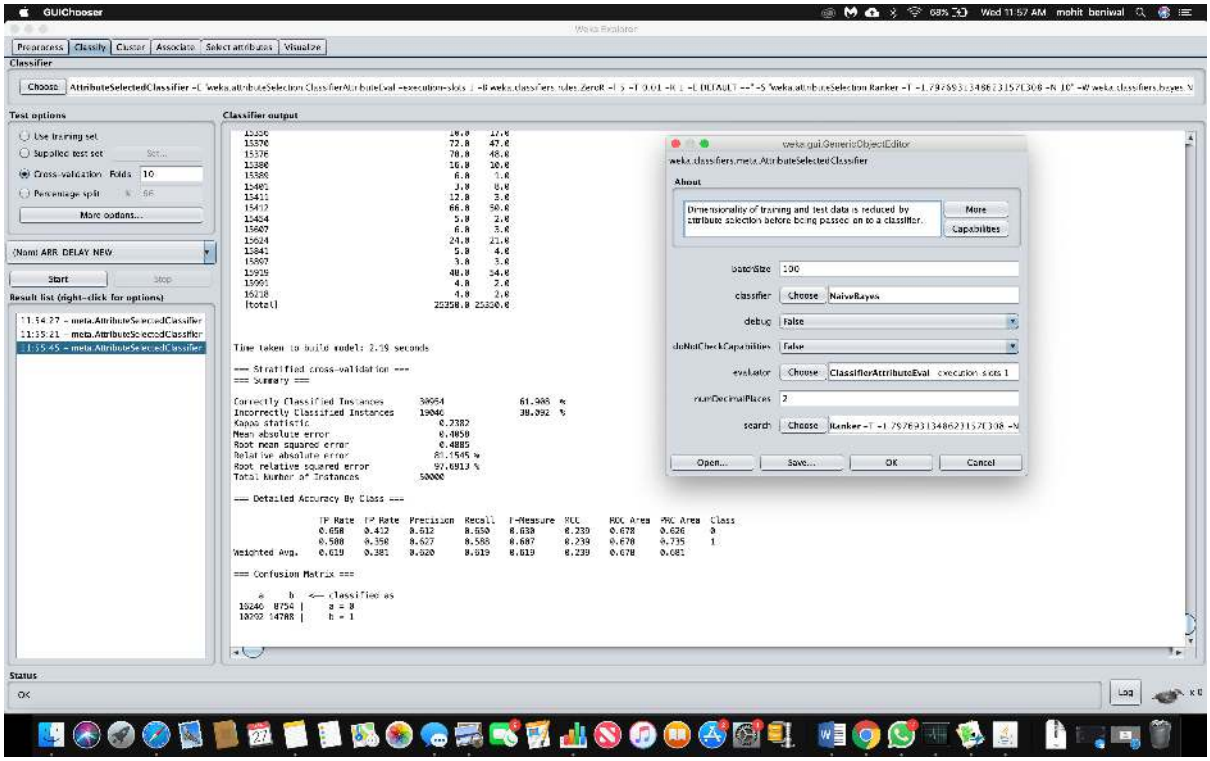
7.1.1 Using OneR Attribute Evaluation



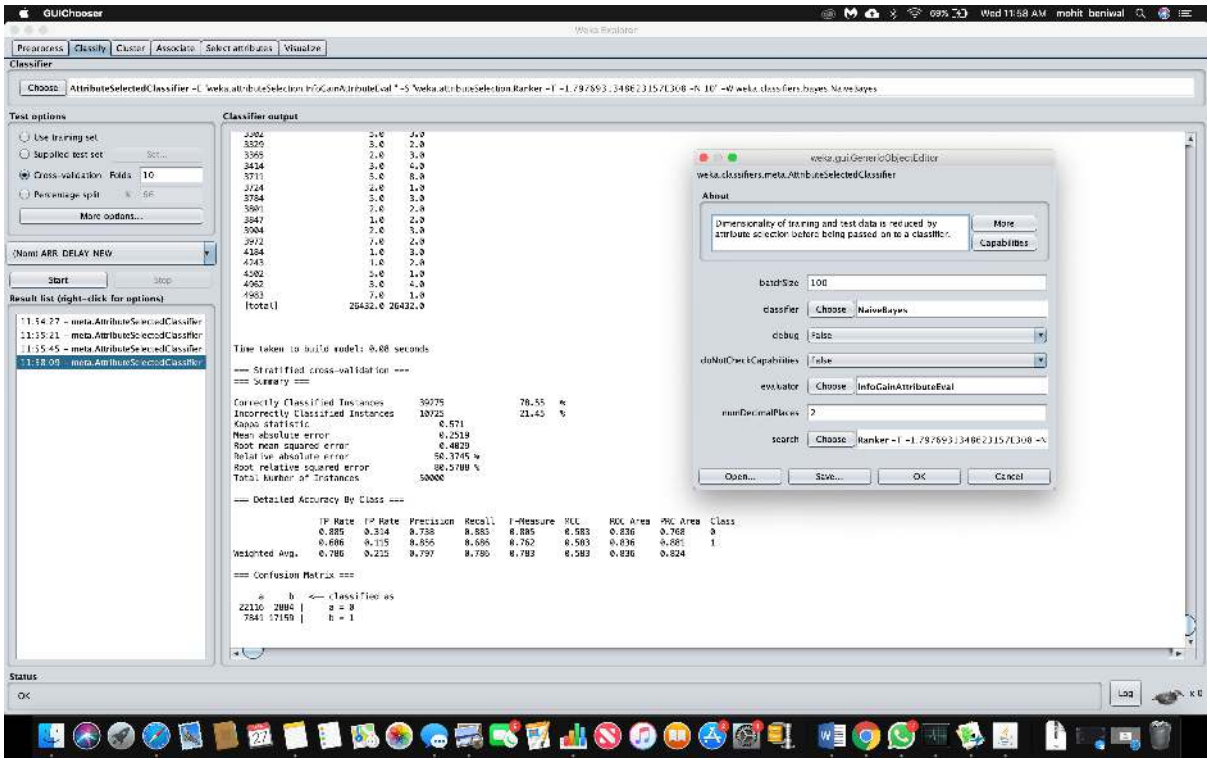
7.1.2 Using Gain Ratio Attribute Evaluation



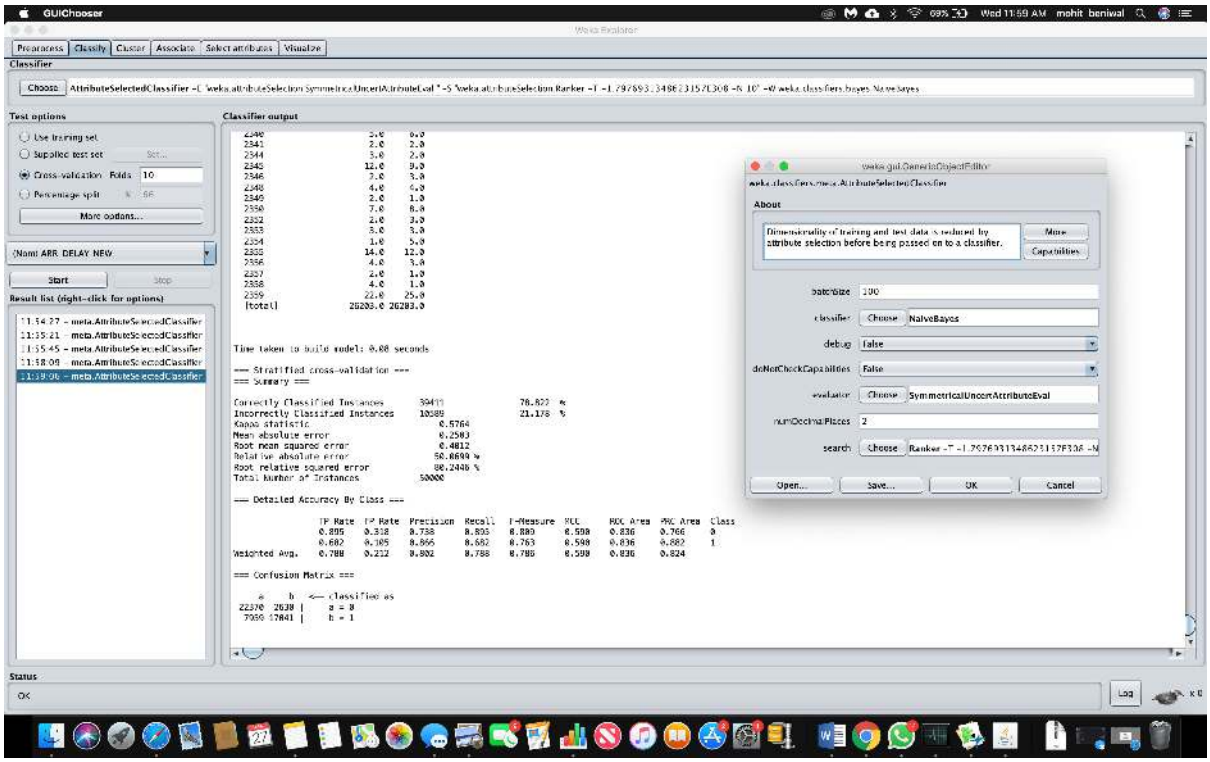
7.1.3 Using Classifier Attribute Evaluation



7.1.4 Using InfoGain Attribute Evaluation



7.1.5 Using SymmetricalUncert Attribute Evaluation



Attribute selection Algorithm	Correctly Classified Instances %	TP Rate	FP Rate	ROC Area	F-Measure	Precision	RMS Error
OneR	79.82	0.798	0.202	0.844	0.796	0.815	0.394
Gain Ratio	79.80	0.798	0.202	0.841	0.795	0.817	0.394
Classifier Attribute	61.98	0.619	0.381	0.678	0.619	0.620	0.488
Info Gain	78.55	0.786	0.215	0.836	0.783	0.797	0.402
Symmetrical Uncert	78.82	0.788	0.212	0.836	0.786	0.802	0.401

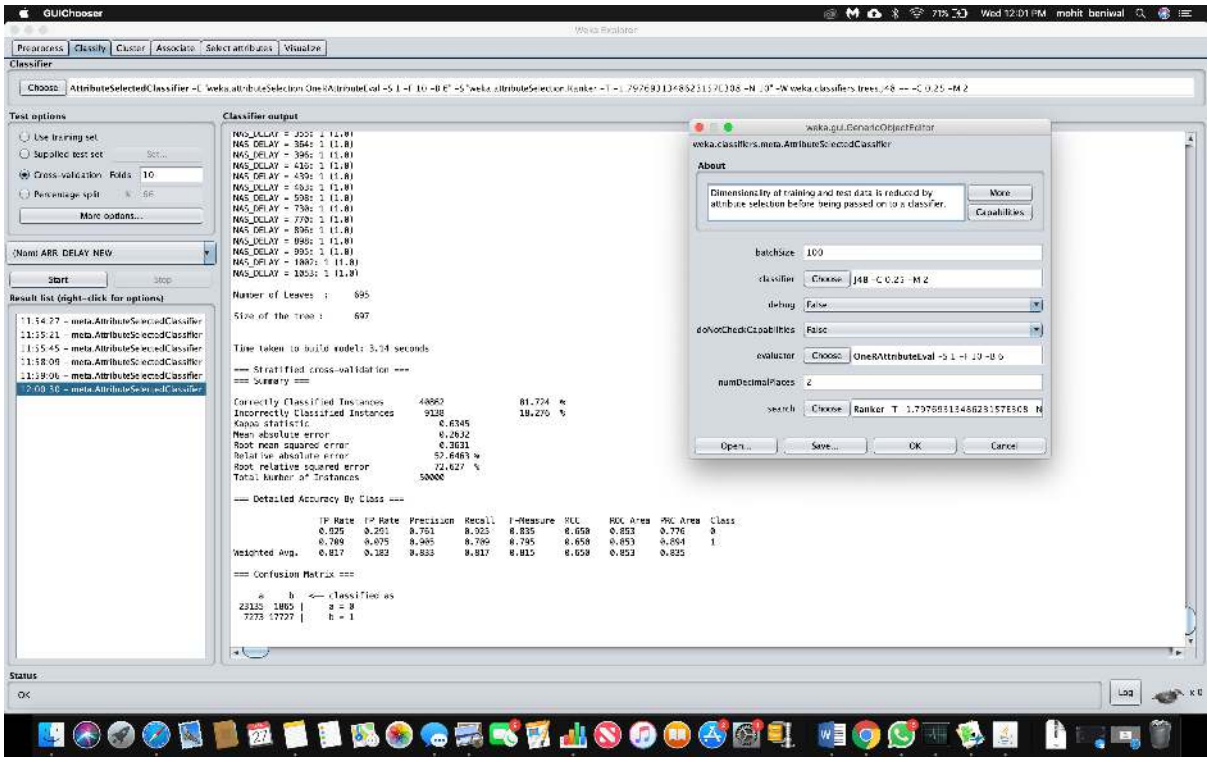
The best model based on ROC is OneR with 0.844.

7.2 J48 Algorithm

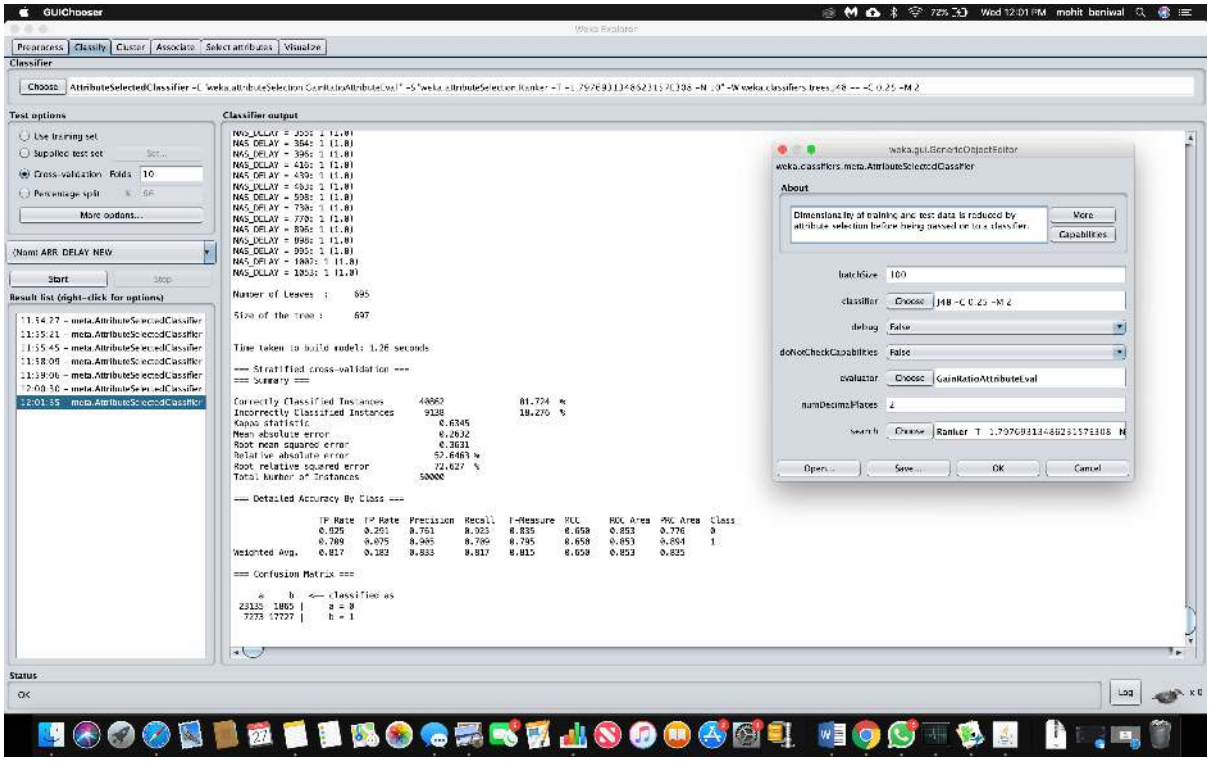
Decision Tree Algorithm is to find out the way the attributes-vector behaves for a number of instances. Also, on the bases of the training instances the classes for the newly generated instances are being found. This algorithm generates the rules for the prediction of the target variable. With the help of tree classification algorithm, the critical distribution of the data is easily understandable. J48 is an extension of ID3. The additional features of J48 are accounting for missing values, decision trees pruning, continuous attribute value ranges, derivation of rules, etc. In the WEKA data mining tool, J48 is an open source Java implementation of the C4.5 algorithm. The WEKA tool provides a number of options associated with tree pruning. In case of potential over fitting pruning can be used as a tool for précising. In other algorithms the classification is performed recursively till every single leaf is pure, that is the classification of the data should be as perfect as possible. This algorithm it generates the rules from which particular identity of that data is generated. The objective is progressively generalization of a decision tree until it gains equilibrium of flexibility and accuracy.

In this case, we have classified the loan borrower's ability to repay based on Defaulter or Not-Defaulter by designing tree.

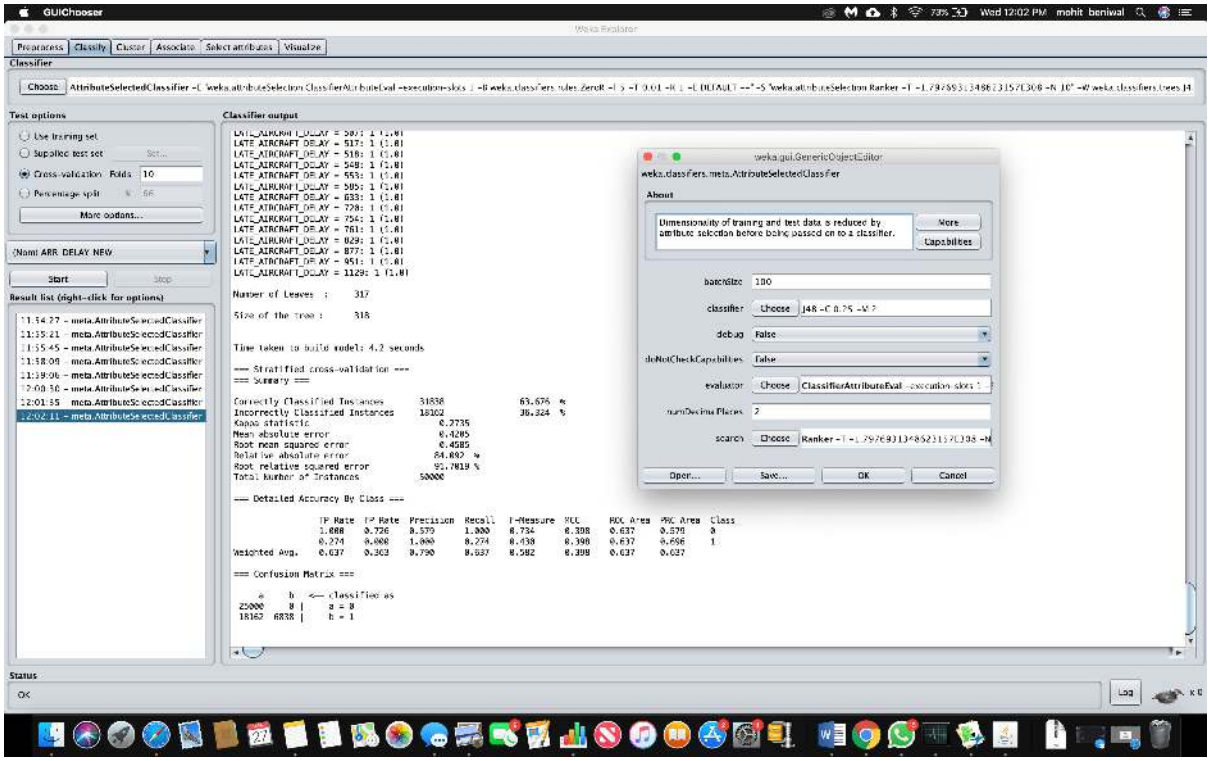
7.2.1 Using OneR Attribute Evaluation



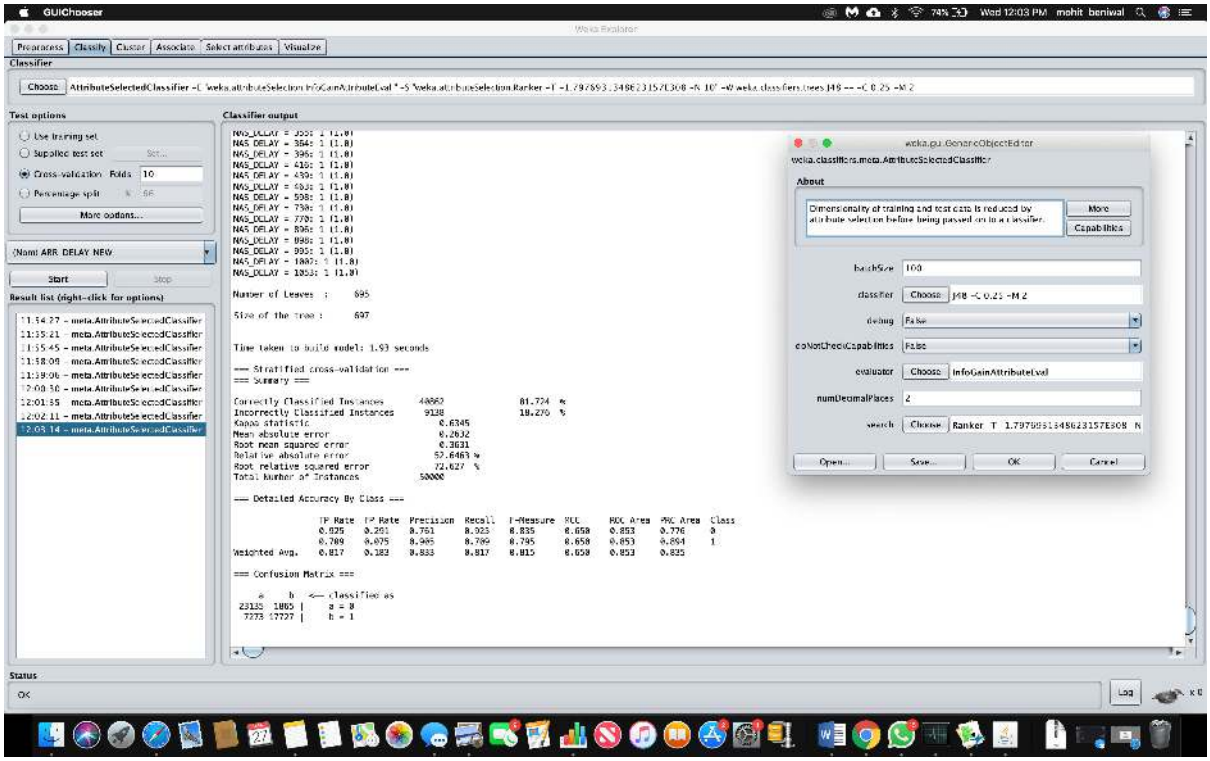
7.2.2 Using Gain Ratio Attribute Evaluation



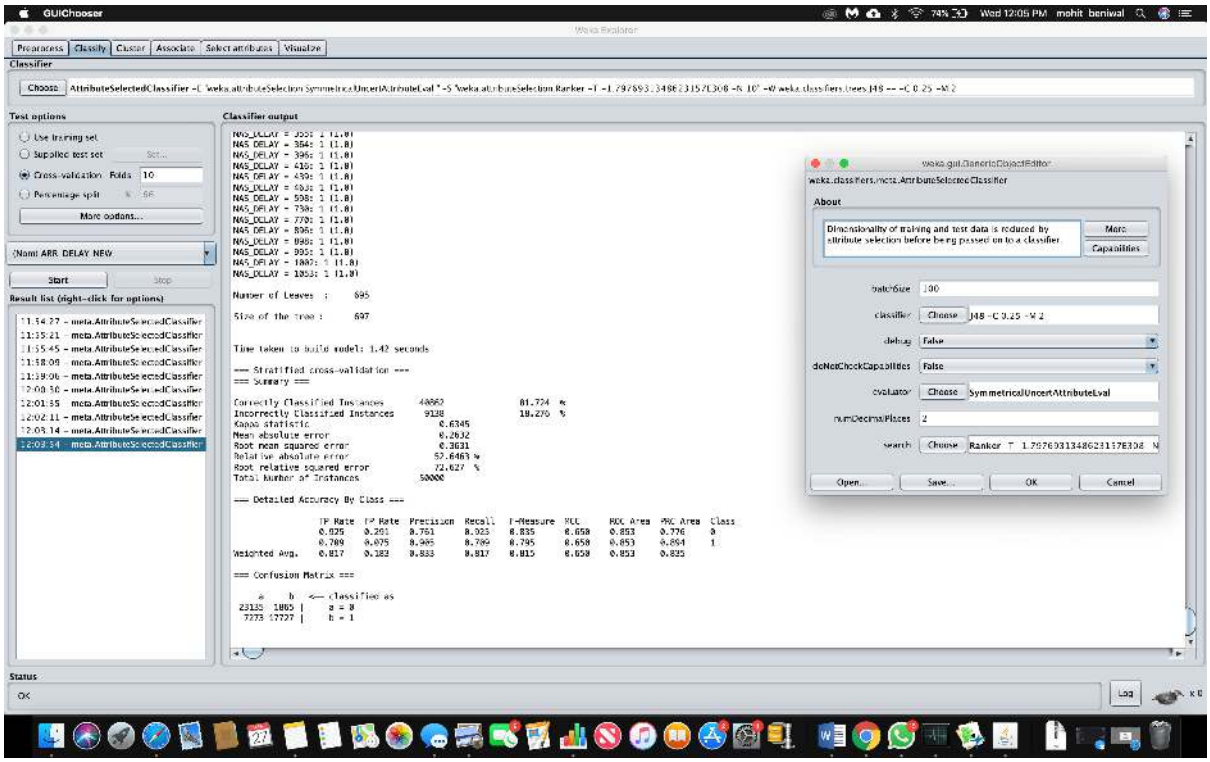
7.2.3 Using Classifier Attribute Evaluation



7.2.4 Using InfoGain Attribute Evaluation



7.2.5 Using SymmetricalUncert Attribute Evaluation



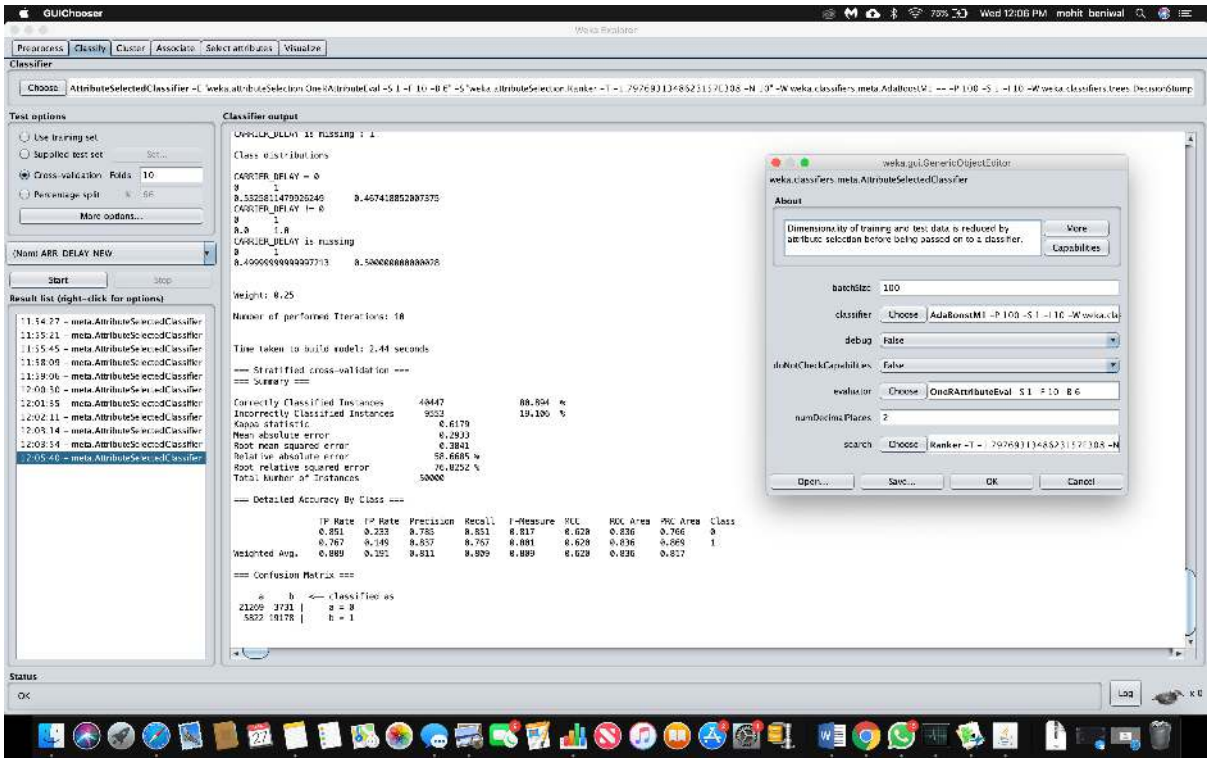
Attribute selection Algorithm	Correctly Classified Instances %	TP Rate	FP Rate	ROC Area	F-Measure	Precision	RMS Error
1R	81.724	0.817	0.183	0.853	0.815	0.833	0.363
Gain Ratio	81.724	0.817	0.183	0.853	0.815	0.833	0.363
Classifier Attribute	63.676	0.637	0.363	0.637	0.582	0.790	0.458
Info Gain	81.724	0.817	0.183	0.853	0.815	0.833	0.363
Symmetrical Uncert	81.724	0.817	0.183	0.853	0.815	0.833	0.363

The best model based on ROC is OneR with 0.853 as four of our classifiers had the exactly same results so we choose the first one.

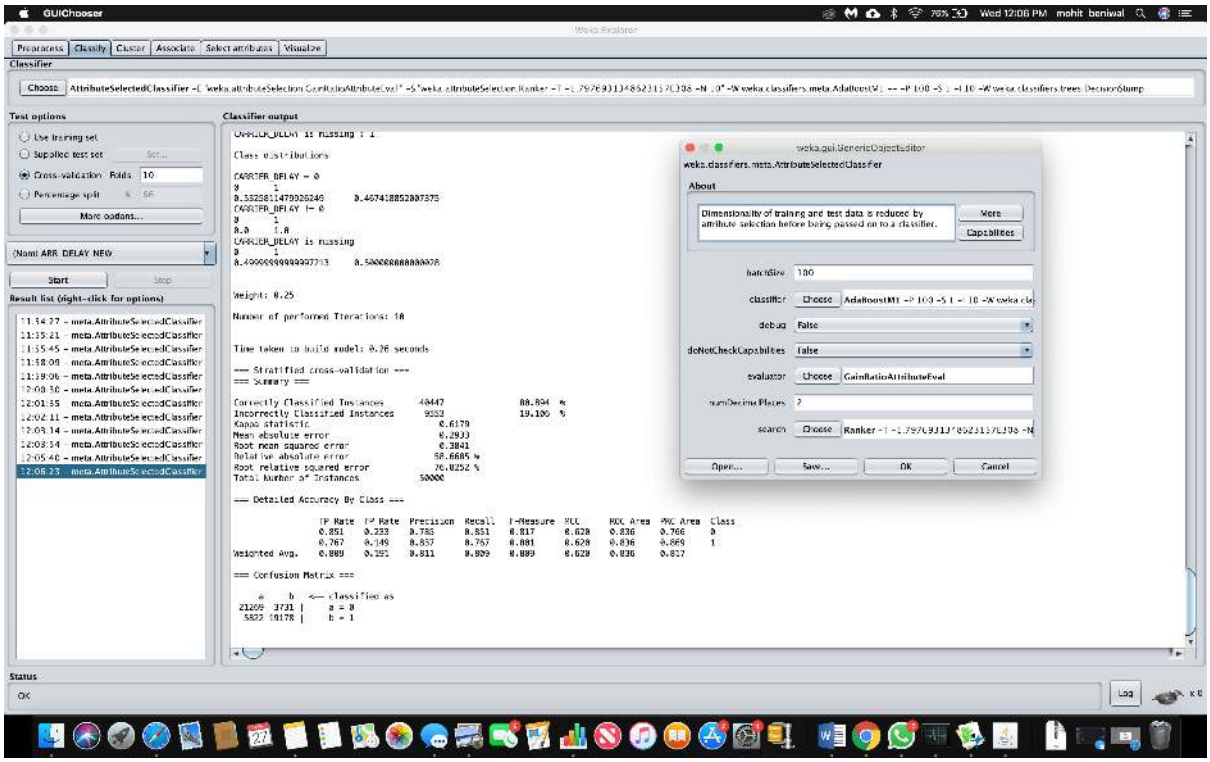
7.3 Adboost M1 Algorithm

AdaBoost, short for Adaptive Boosting, is a machine learning meta-algorithm formulated by Yoav Freund and Robert Schapire, who won the 2003 Gödel Prize for their work. It can be used in conjunction with many other types of learning algorithms to improve performance. The output of the other learning algorithms ('weak learners') is combined into a weighted sum that represents the final output of the boosted classifier. AdaBoost is adaptive in the sense that subsequent weak learners are tweaked in favor of those instances misclassified by previous classifiers. AdaBoost is sensitive to noisy data and outliers. In some problems it can be less susceptible to the overfitting problem than other learning algorithms. The individual learners can be weak, but as long as the performance of each one is slightly better than random guessing, the final model can be proven to converge to a strong learner.

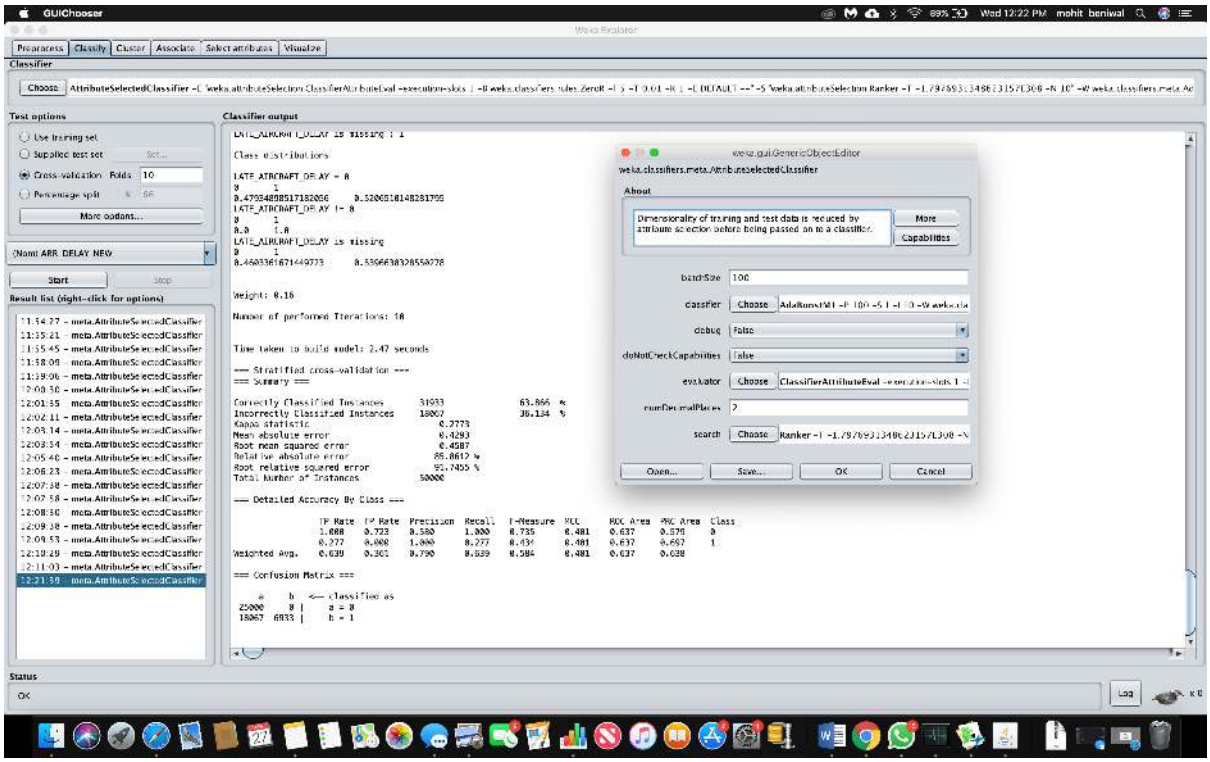
7.3.1 Using OneR Attribute Evaluation



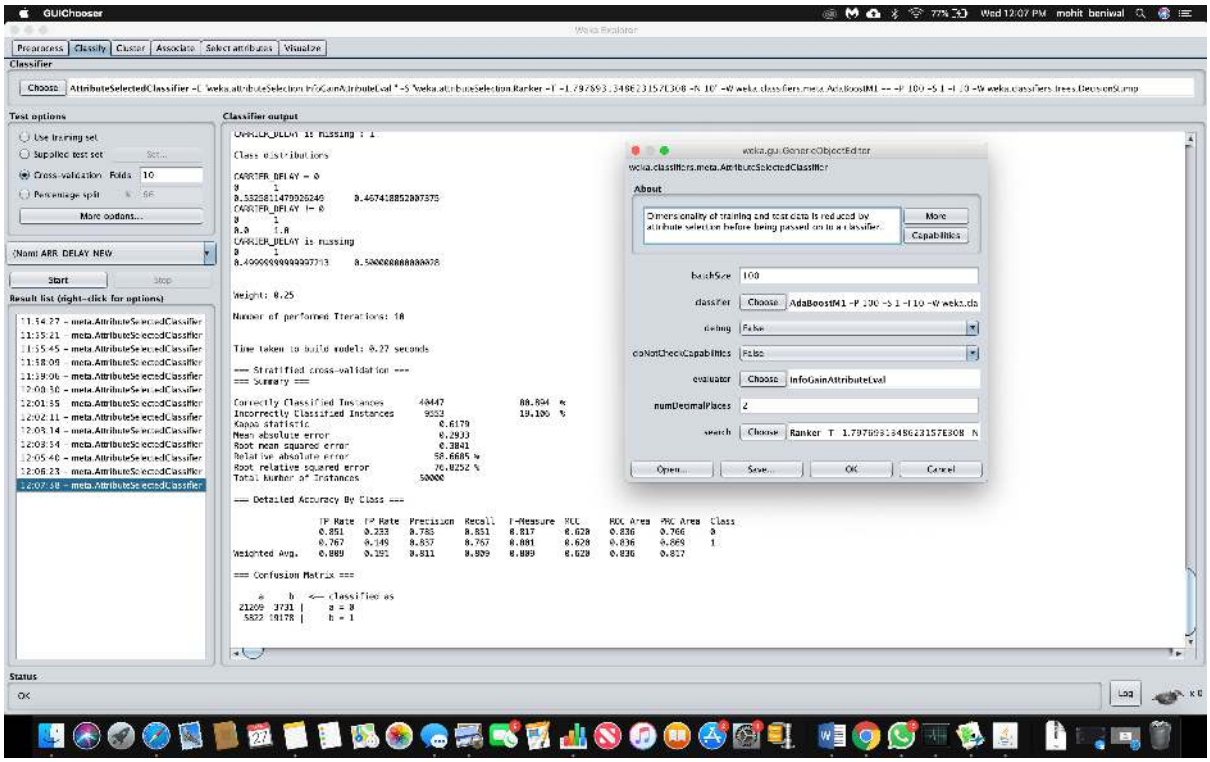
7.3.2 Using Gain Ratio Attribute Evaluation



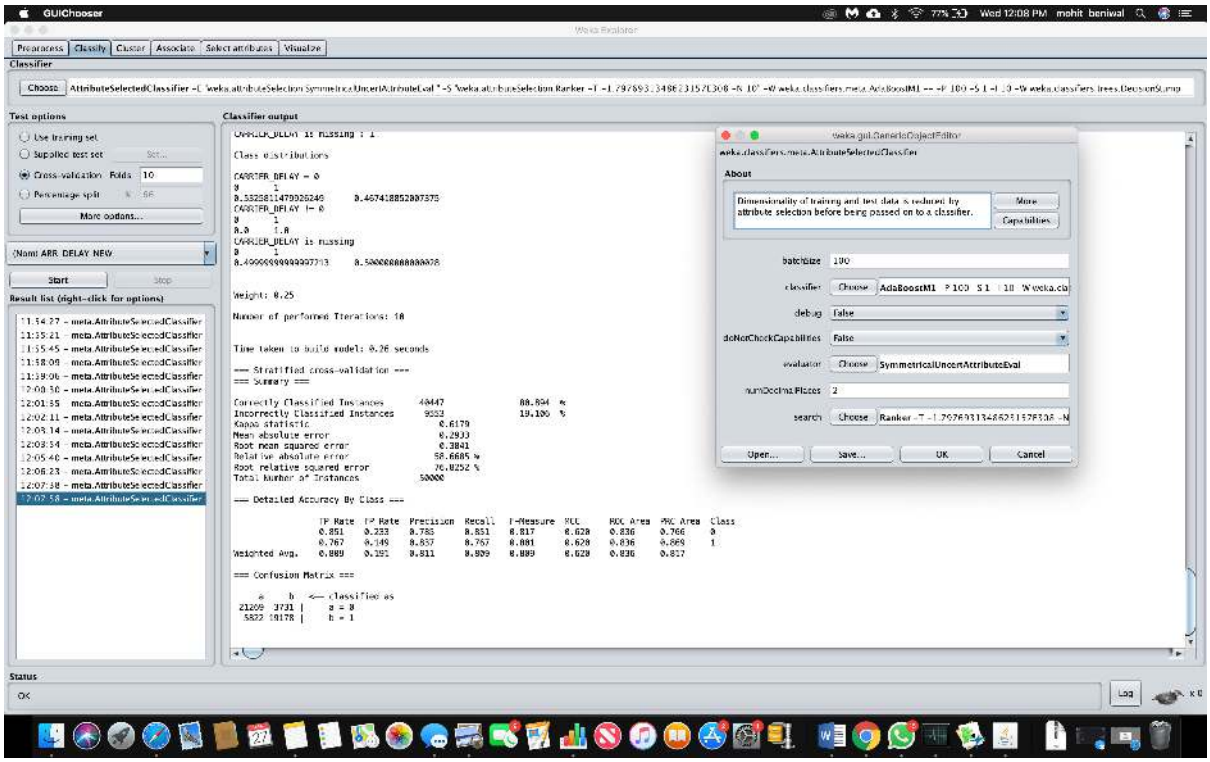
7.3.3 Using Classifier Attribute Evaluation



7.3.4 Using InfoGain Attribute Evaluation



7.3.5 Using SymmetricalUncert Attribute Evaluation



Attribute selection Algorithm	Correctly Classified Instances %	TP Rate	FP Rate	ROC Area	F-Measure	Precision	RMS Error
1R	80.894	0.809	0.191	0.836	0.809	0.811	0.384
Gain Ratio	80.894	0.809	0.191	0.836	0.809	0.811	0.384
Classifier Attribute	63.866	0.639	0.361	0.637	0.584	0.790	0.458
Info Gain	80.894	0.809	0.191	0.836	0.809	0.811	0.384
Symmetrical Uncert	80.894	0.809	0.191	0.836	0.809	0.811	0.384

The best model based on ROC is OneR with 0.836 as four of our classifiers had the exactly same results so we choose the first one.

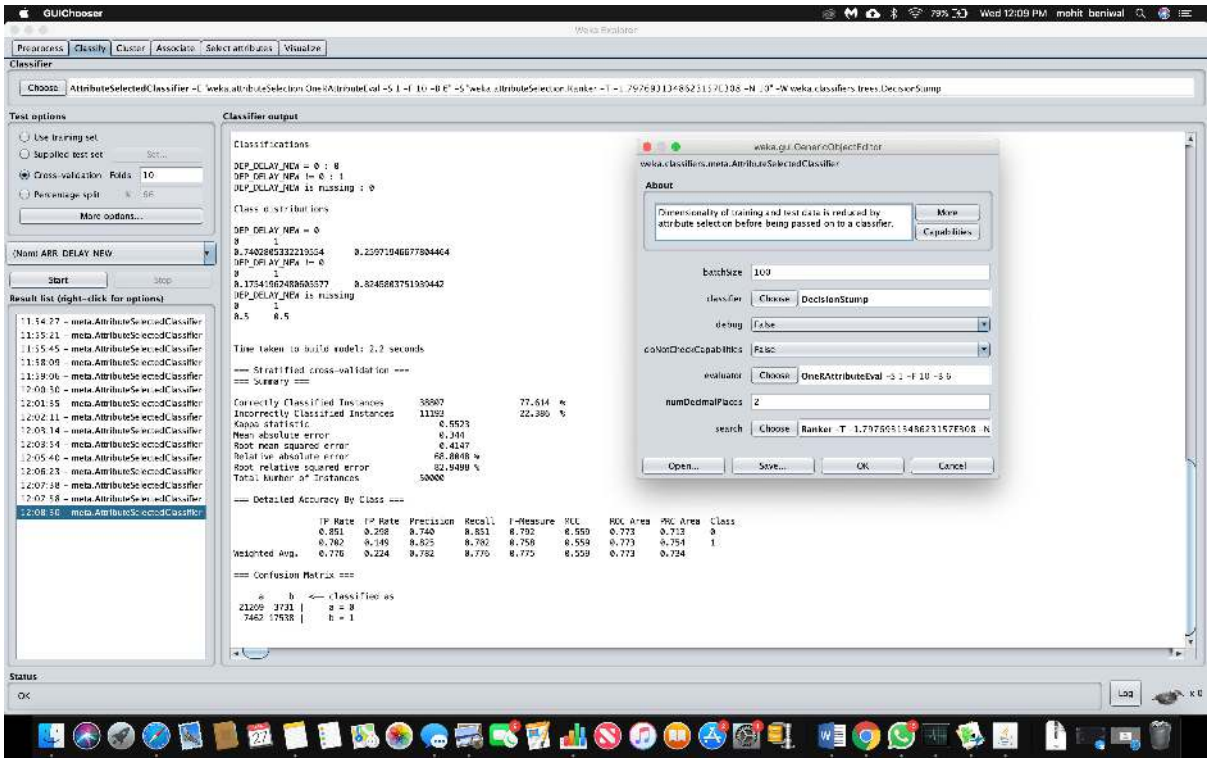
7.4 Decision Stump Algorithm

A decision stump is a machine learning model consisting of a one-level decision tree. That is, it is a decision tree with one internal node (the root) which is immediately connected to the terminal nodes (its leaves). A decision stump makes a prediction based on the value of just a single input feature. Sometimes they are also called 1-rules.

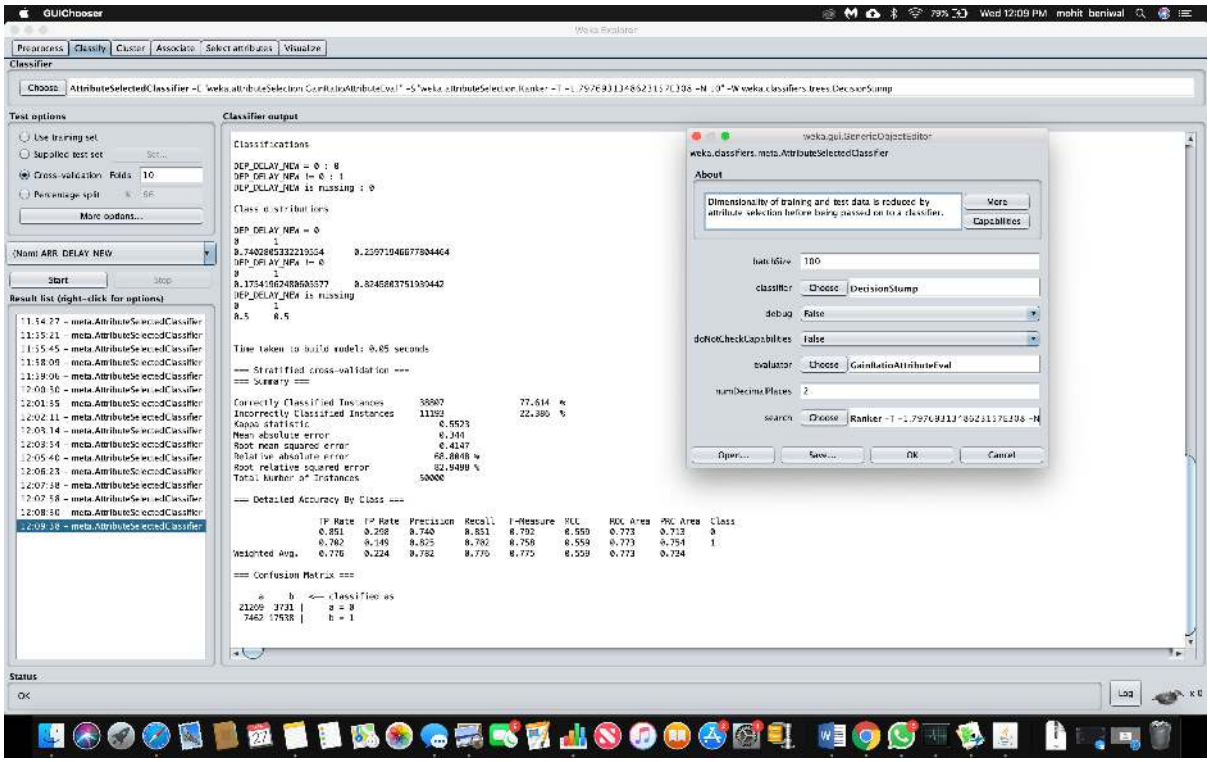
Depending on the type of the input feature, several variations are possible. For nominal features, one may build a stump which contains a leaf for each possible feature value or a stump with the two leaves, one of which corresponds to some chosen category, and the other leaf to all the other categories. For binary features these two schemes are identical. A missing value may be treated as a yet another category.

Decision stumps are often used as components (called "weak learners" or "base learners") in machine learning ensemble techniques such as bagging and boosting.

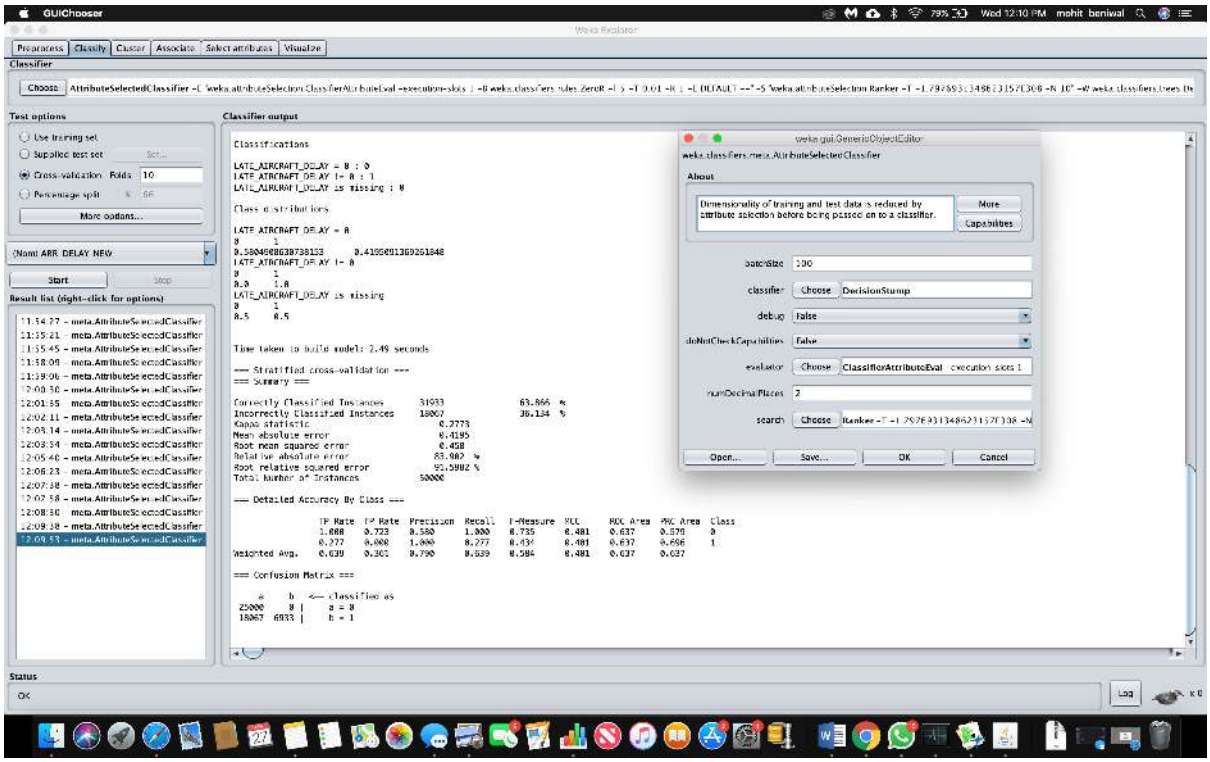
7.4.1 Using OneR Attribute Evaluation



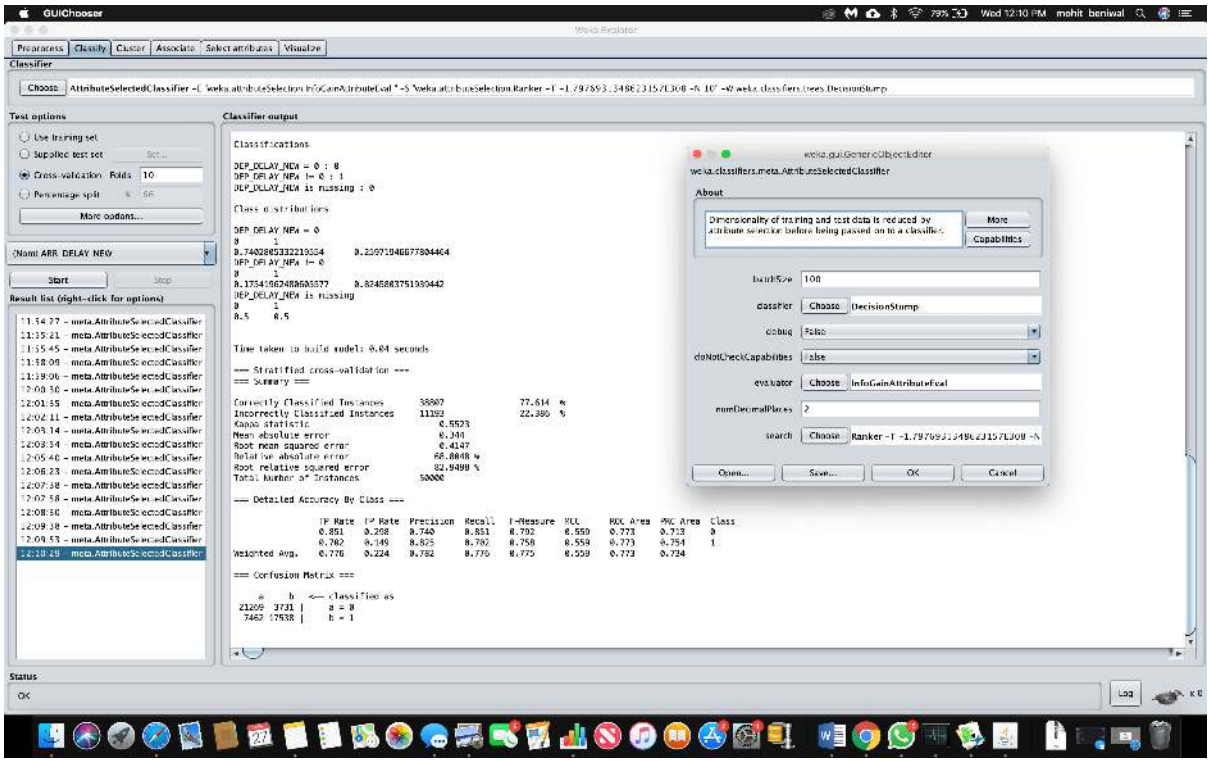
7.4.2 Using Gain Ratio Attribute Evaluation



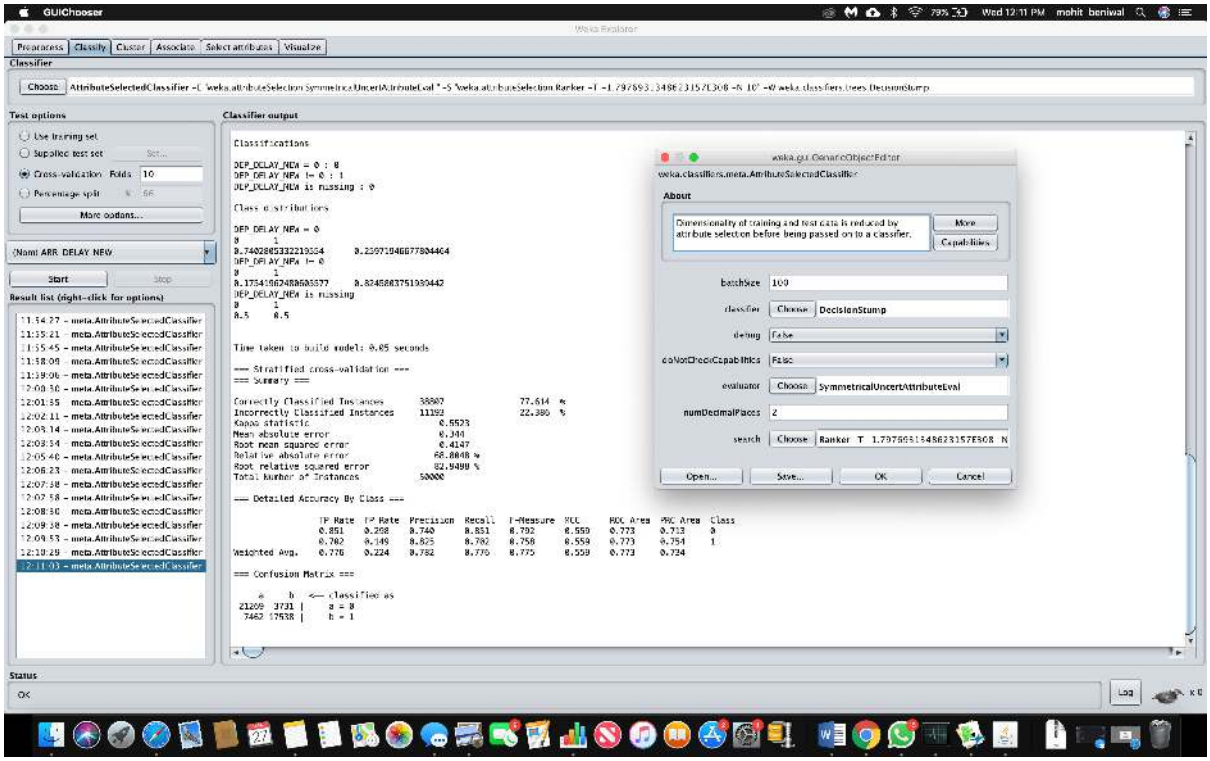
7.4.3 Using Classifier Attribute Evaluation



7.4.4 Using InfoGain Attribute Evaluation



7.4.5 Using SymmetricalUncert Attribute Evaluation



Attribute selection Algorithm	Correctly Classified Instances %	TP Rate	FP Rate	ROC Area	F-Measure	Precision	RMS Error
1R	77.614	0.776	0.224	0.773	0.775	0.782	0.414
Gain Ratio	77.614	0.776	0.224	0.773	0.775	0.782	0.414
Classifier Attribute	64.004	0.640	0.360	0.637	0.586	0.791	0.457
Info Gain	77.614	0.776	0.224	0.773	0.775	0.782	0.414
Symmetrical Uncert	77.614	0.776	0.224	0.773	0.775	0.782	0.414

The best model based on ROC is OneR with 0.773 as four of our classifiers had the exactly same results so we choose the first one.

8 Results

After all the preprocessing and deploying the model in WEKA, there were several metrics evolving across the 20 models. The major metrics that should be taken into consideration are ROC area, Precision and Root mean square error.

But considering ROC area as a major parameter to select the best model, J48 with OneR Attribute selection gives the highest ROC area (0.853). The other models weren't compared based on the class accuracy because sometimes there may be a case where the class is biased, or the TP rate may not have been accurate.

9 Conclusions

In this project, we have developed a model to predict if there will be a delay for a given flight detail.

10 Work Distribution among team mates

The project was divided into 3 phases:

1. Data Gathering
2. Data Processing
3. Model building and Testing
4. Documentation

The data was gathered by Mohit. We both had further used that data for preprocessing using JMP Pro. During the initial phase project there were several discussions on which models to be used that could contribute towards a better model.

Then came the model building and testing part. We contemplate over which attribute selection algorithm and which classification to use. So, we distributed the load equally and ran all the 20 combinations took screenshots. We then cross validated the results and then went for documentation.

The documentation which is a very important task was shared between Abhishek and Mohit. Mohit had documented the clubbing and preprocessing part. Abhishek then led the effort to write about the models, and together we wrote the summary of results.

11 Learning Outcomes

The project was a unique learning experience for us. We never had worked on such a humongous dataset. Cleaning and processing the data was very tedious task and we finally able to do so. Meanwhile, we learnt some interesting idea of how the raw data needs to be processed and then build the model. Moreover, the data engineering part was a tough task. We learnt a lot about what features are and how would they be a part for a good model.

This project has given us the ideas of working on dataset and performing various data science tasks. There were several challenges that we faced during the project building.

1. Working with large data.
2. Preparing the data for ready to use form.
3. Choosing the attributes to work with.
4. Underlying working of Data Mining algorithms in projects.
5. Comparing the performance of models.

12 References

1.Naïve Bayes: Devin Soni, "Introduction to Naïve Bayes Classification" (source: https://towardsdatascience.com/introduction-to-naive-bayes-classification-4cffabb1ae54)

2.J48: Gaganjot Kaur, "Improved J48 Classification Algorithm for the Prediction of Diabetes" (source:http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.678.9273&rep=rep1&type=pdf)

3. AdaBoost M1 (source: https://en.wikipedia.org/wiki/AdaBoost)
4. Decision Stump: (source: https://en.wikipedia.org/wiki/Decision_stump)