



Mohit Mohit  
Abhishek Rai Sharma

# DATA MINING

Airline Delay Prediction

## Table of Contents

<b>1</b>	<b><i>Introduction</i></b>	2
<b>2</b>	<b><i>Description of Dataset</i></b>	2
<b>2.1</b>	<b>Data Description</b>	2
<b>2.2</b>	<b>Goal</b>	3
<b>2.3</b>	<b>Loading Data in JMP Pro</b>	3
<b>2.4</b>	<b>Creating the Class Attribute</b>	5
<b>2.5</b>	<b>Data Analysis</b>	6
<b>2.6</b>	<b>Is our class imbalanced?</b>	7
<b>2.7</b>	<b>Data Reduction and class balancing</b>	7
<b>2.8</b>	<b>Data Cleaning</b>	8
<b>2.9</b>	<b>Handling Missing Data</b>	10
<b>2.10</b>	<b>Exporting data into csv</b>	12
<b>2.11</b>	<b>Data Loading into Weka</b>	13
<b>2.12</b>	<b>Specifying class attribute</b>	14
<b>2.13</b>	<b>Data Wrangling</b>	15
<b>2.14</b>	<b>Saving the WEKA compatible file [.arff]</b>	16
<b>2.15</b>	<b>Attribute Selection</b>	16
<b>2.16</b>	<b>OneR Attribute Evaluation</b>	17
<b>2.17</b>	<b>Gain Ratio Attribute Evaluation</b>	17
<b>2.18</b>	<b>Classifier Attribute Evaluation</b>	18
<b>2.19</b>	<b>Info Gain Attribute Evaluation</b>	18
<b>2.20</b>	<b>SymmetricUncert Attribute Evaluation</b>	19
<b>3</b>	<b><i>Classification</i></b>	20
<b>3.1</b>	<b>Naïve Bayes Algorithm</b>	20
<b>3.2</b>	<b>J48 Algorithm</b>	23
<b>3.3</b>	<b>Adboost M1 Algorithm</b>	27
<b>3.4</b>	<b>Decision Stump Algorithm</b>	30
<b>4</b>	<b><i>Results</i></b>	34
<b>5</b>	<b><i>Conclusions</i></b>	34
<b>6</b>	<b><i>Work Distribution among team mates</i></b>	34
<b>7</b>	<b><i>Learning Outcomes</i></b>	35
<b>8</b>	<b><i>References</i></b>	35

## 1 Introduction

Delay is painful. We get annoyed when the flights are delayed. However, when we know that the flight will be late, it's 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](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)

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

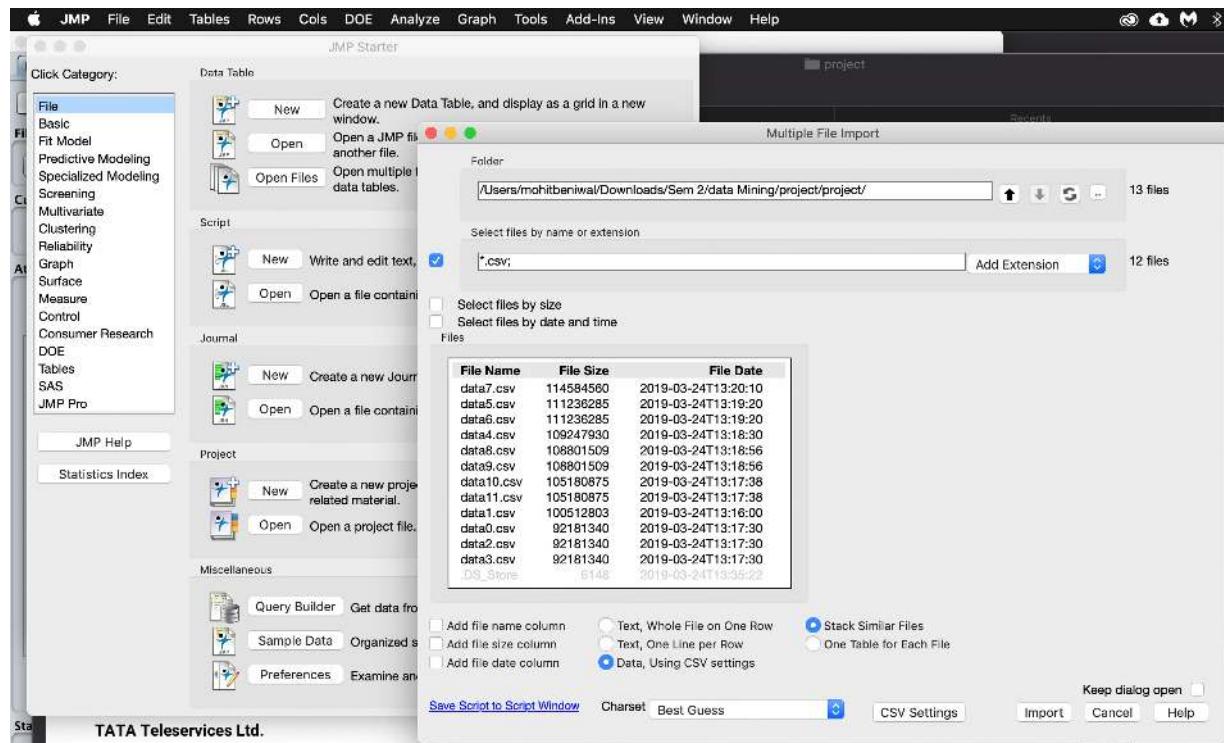
### 4 Data Manipulation in JMP Pro

As our dataset is huge and has multiple files to load from, we decided to use JMP PRO by looking at its efficiency to handle 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

	CRS ARR TIME	ACTUAL ELAPS ED TIME	CARRIER DELAY
1	1745	1745	0
2	0	1254	0
3	0	1148	0
4	0	123733	1
5	20	922	1
6	8	14	1
7	0	918	0
8	0	1818	0
9	12	939	1
10	0	1613	0
11	0	75160	0
12	121	2311	0
13	0	1135	0
14	11	1318	0
15	76	15	0
16	64	300	0
17	72	2198	0
18	47	1537	0
19	0	1218	0
20	0	86682	0
21	-41	65129	0
22	0	1129	0
23	0	1302	0
24	0	31759	0
25	0	23225	0
26	0	15	0
27	0	1119	0
28	6	2116	0
29	0	25996	0
30	0	1810	0
31	48	24291	0
32	0	35145	0
33	0	1058	0
34	0	25065	0
35	15	1822	0
36	0	22069	0
37	0	844	0
38	0	834	0
39	0	19579	0
40	0	1242	0
41	0	14409	0
42	0	950	0
43	0	35670	0
44	0	1411	0
45	0	12655	0
46	0	44	0
47	0	2358	0

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.

Screenshot of JMP software showing the Column Viewer for the 'ARR\_DELAY\_NEW' dataset. The Column Viewer interface includes a tree view of columns on the left, a summary table in the center, and a detailed table on the right. A context menu is open over the summary table, showing options like 'Group to new value...', 'Group to 2', 'Group to 4', etc.

## 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

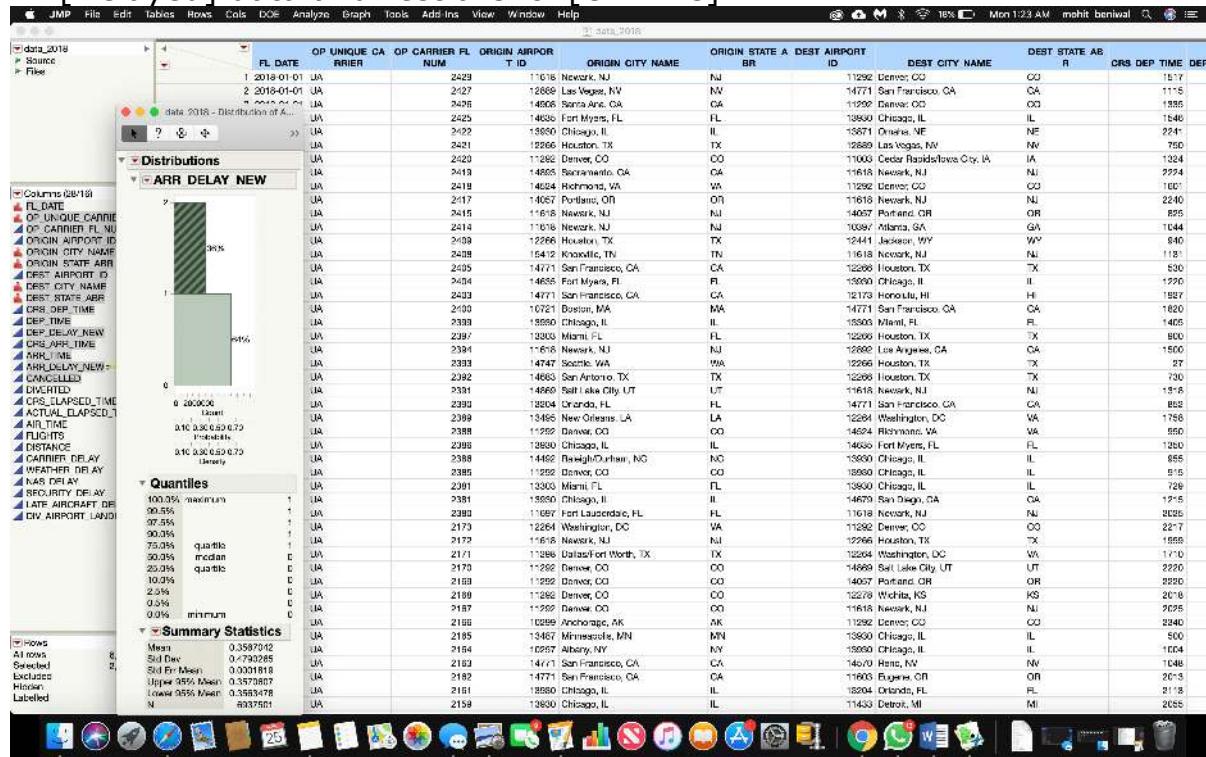
Screenshot of JMP software showing the Column Viewer for the 'ARR\_DELAY\_NEW' dataset. The Column Viewer interface includes a tree view of columns on the left, a summary table in the center, and a detailed table on the right. A context menu is open over the summary table, showing options like 'Group to new value...', 'Group to 2', 'Group to 4', etc.

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 is 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”

The screenshot shows the JMP interface with a data table open. A context menu is displayed over the data, specifically the 'Subset' option under 'Tables'. A modal dialog box for 'Subset' is open, allowing the user to define the subset based on selected rows and columns. The 'Sampling' tab is active, showing 'Random - sample size' set to 25000. Other tabs like 'Rows' and 'Columns' are visible but inactive.

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.

Sun 7:46 PM mahit.benivali Q

	N_STATE_A	DEST_AIRPORT_ID	DEST_CITY_NAME	DEST_STATE_AB	R	CRS_DEP_TIME	DEP_TIME_W	DEP_DELAY_NE	CRS_ARR_TIME	ARR_TIME_W	CANCELLATION_CODE
1	13266	MesaPhoenix/Chandler	TX	TX	14:40	*	*	*	16:01	*	1 B
2	14771	San Francisco	CA	CA	17:44	*	*	*	19:26	*	1 A
3	14678	San Diego	CA	CA	7:28	7:19	0	9:03	11:27	*	0
4	12266	Houston	TX	TX	17:56	*	*	*	18:44	*	1 B
5	12892	Los Angeles	CA	CA	9:10	*	*	*	12:30	*	1 B
6	15841	Wrangell	AK	AK	15:14	15:08	72	15:38	*	*	0
7	12819	Ketchikan	AK	AK	16:23	*	*	*	18:55	*	1 D
8	14266	Petrolia	AK	AK	16:56	*	*	*	11:18	*	1 D
9	15841	Wrangell	AK	AK	9:28	9:18	0	10:11	*	*	0
10	12823	Jurupa	AK	AK	12:04	*	*	*	12:48	*	*
11	12823	Jurupa	AK	AK	6:00	*	*	*	6:41	*	1 A
12	14678	San Diego	CA	CA	5:56	5:41	0	6:46	10:48	*	0
13	12269	Anchorage	AK	AK	5:45	*	*	*	6:52	*	1 A
14	14747	Seattle	WA	WA	5:00	*	*	*	7:18	*	1 A
15	14678	San Diego	CA	CA	7:25	7:22	0	10:17	12:15	*	0
16	14678	San Diego	CA	CA	6:25	6:21	0	6:52	7:15	*	0
17	14678	San Diego	CA	CA	10:25	*	*	*	12:54	*	0
18	14678	San Diego	CA	CA	8:28	8:23	0	9:05	11:16	*	0
19	14678	San Diego	CA	CA	8:05	8:05	0	7:58	7:58	*	0
20	11485	Detroit	MI	MI	8:05	8:05	0	7:58	7:58	*	0
21	11483	Detroit	MI	MI	12:14	12:22	0	1:45	1:45	*	0
22	11042	Cleveland	OH	OH	12:05	12:01	0	11:13	1:33	*	0
23	15842	Knoxville	TN	TN	19:27	19:29	22	23:01	23:01	*	0
24	12823	Baltimore	MD	MD	12:15	*	*	*	13:52	12:52	*
25	12828	Dallas/Ft Worth	TX	TX	8:50	*	*	*	9:48	*	1 A
26	12828	Dallas/Ft Worth	TX	TX	7:00	*	*	*	8:35	*	1 A
27	12828	Dallas/Ft Worth	TX	TX	5:45	*	*	*	7:42	*	1 A
28	12828	Killeen	TX	TX	9:04	*	*	*	10:05	*	1 A
29	12828	Dallas/Ft Worth	TX	TX	10:25	*	*	*	11:25	*	1 A
30	11066	Columbus	OH	OH	8:55	8:59	34	10:53	1:54	*	0
31	11042	Cleveland	OH	OH	8:45	*	*	*	9:15	*	1 C
32	10717	Brownsville	TX	TX	10:05	*	*	*	11:30	*	1 B
33	14824	Richmond	VA	VA	15:30	*	*	*	19:12	*	1 A
34	13066	Laredo	TX	TX	10:05	*	*	*	11:33	*	1 D
35	12966	Houston	TX	TX	12:00	*	*	*	13:19	*	1 B
36	13066	Laredo	TX	TX	14:20	*	*	*	15:40	*	1 D
37	12966	Houston	TX	TX	16:10	*	*	*	17:25	*	1 B
38	12966	Houston	TX	TX	12:03	*	*	*	13:20	*	1 B
39	11065	Cedar Rapids/Iowa City	IA	IA	6:18	11:14	205	14:05	*	*	0
40	12828	Denver	CO	CO	6:35	5:25	0	7:04	*	*	1 A
41	14683	San Antonio	TX	TX	6:00	6:38	38	8:03	*	*	0
42	12264	Orlando	FL	FL	6:00	8:01	121	9:00	*	*	1 B
All Rows	7,065,785										
Selected	43										
Excluded	0										
Hidden	0										
Labelled	0										
Labeled	47										

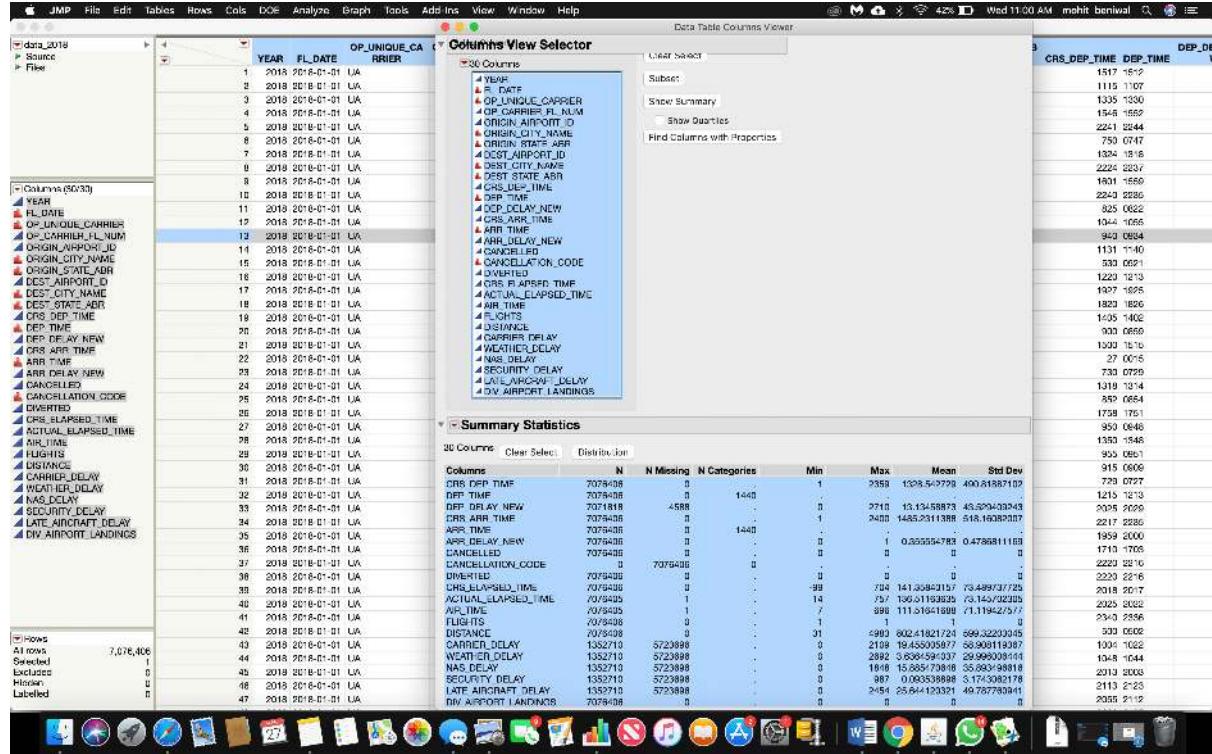
Now we can easily delete all unwanted tuples as shown below.

Sun 7:50 PM mahit.benivali Q

	N_STATE_A	DEST_AIRPORT_ID	DEST_CITY_NAME	DEST_STATE_AB	R	CRS_DEP_TIME	DEP_TIME_W	DEP_DELAY_NE	CRS_ARR_TIME	ARR_TIME_W	CANCELLATION_CODE
1	13266	MesaPhoenix/Chandler	TX	TX	14:40	*	*	*	16:01	*	1 B
2	14771	San Francisco	CA	CA	17:44	*	*	*	19:26	*	1 A
3	14678	San Diego	CA	CA	7:28	7:19	0	9:03	11:27	*	0
4	12266	Houston	TX	TX	17:56	*	*	*	18:44	*	1 B
5	12892	Los Angeles	CA	CA	9:10	*	*	*	12:30	*	1 B
6	15841	Wrangell	AK	AK	15:14	15:08	72	15:38	*	*	0
7	12819	Ketchikan	AK	AK	16:23	*	*	*	18:55	*	1 D
8	14266	Petrolia	AK	AK	9:28	9:18	0	10:11	*	*	0
9	15841	Wrangell	AK	AK	12:04	*	*	*	12:48	*	1 B
10	12823	Jurupa	AK	AK	6:00	*	*	*	6:41	*	1 A
11	12823	Jurupa	AK	AK	12:01	*	*	*	12:54	*	1 A
12	14678	San Diego	CA	CA	5:45	*	*	*	6:52	*	1 A
13	12269	Anchorage	AK	AK	5:45	*	*	*	6:52	*	1 A
14	14747	Seattle	WA	WA	5:00	*	*	*	7:18	*	1 A
15	14678	San Diego	CA	CA	7:25	7:22	0	10:17	12:15	*	0
16	14678	San Diego	CA	CA	6:25	6:21	0	6:52	7:15	*	0
17	14678	San Diego	CA	CA	10:25	*	*	*	12:54	*	0
18	14678	San Diego	CA	CA	8:28	8:23	0	9:05	11:16	*	0
19	14678	San Diego	CA	CA	8:05	8:05	0	7:58	7:58	*	0
20	11485	Detroit	MI	MI	8:05	8:05	0	7:58	7:58	*	0
21	11483	Detroit	MI	MI	12:14	12:22	0	1:45	1:45	*	0
22	11042	Cleveland	OH	OH	12:05	12:01	551	6	16:09	18:59	*
23	15842	Knoxville	TN	TN	19:27	19:29	22	23:01	23:01	*	0
24	12823	Baltimore	MD	MD	12:15	2:15	*	*	13:52	12:52	*
25	12828	Dallas/Ft Worth	TX	TX	8:50	*	*	*	9:48	*	1 A
26	12828	Dallas/Ft Worth	TX	TX	7:00	*	*	*	7:42	*	1 A
27	12828	Dallas/Ft Worth	TX	TX	5:45	*	*	*	6:05	*	1 A
28	12828	Killeen	TX	TX	9:04	*	*	*	10:05	*	1 A
29	12828	Dallas/Ft Worth	TX	TX	10:25	*	*	*	11:25	*	1 A
30	11066	Columbus	OH	OH	8:55	8:59	34	10:53	1:54	*	0
31	11045	Cincinnati	OH	OH	8:45	*	*	*	9:15	*	1 B
32	10447	Brownsville	TX	TX	10:00	*	*	*	11:00	*	1 B
33	14824	Rutherford	VA	VA	15:00	*	*	*	16:10	*	1 A
34	12001	Laredo	TX	TX	10:05	*	*	*	11:03	*	1 B
35	12288	Houston	TX	TX	12:00	*	*	*	13:19	*	1 B
36	12001	Laredo	TX	TX	14:20	*	*	*	15:40	*	1 B
37	12288	Houston	TX	TX	16:10	*	*	*	17:25	*	1 D
38	12966	Houston	TX	TX	12:03	*	*	*	13:00	*	1 B
39	11065	Cedar Rapids/Iowa City	IA	IA	6:19	11:14	288	14:05	*	*	0
40	12828	Denver	CO	CO	5:35	5:55	35	7:04	*	*	1 A
41	14683	San Antonio	TX	TX	6:00	6:36	36	8:03	*	*	0
42	12264	Washington	DC	DC	6:00	8:01	-21	9:00	*	*	1 B
All Rows	7,065,785										
Selected	43										
Excluded	0										
Hidden	0										
Labelled	0										
Labeled	47										

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



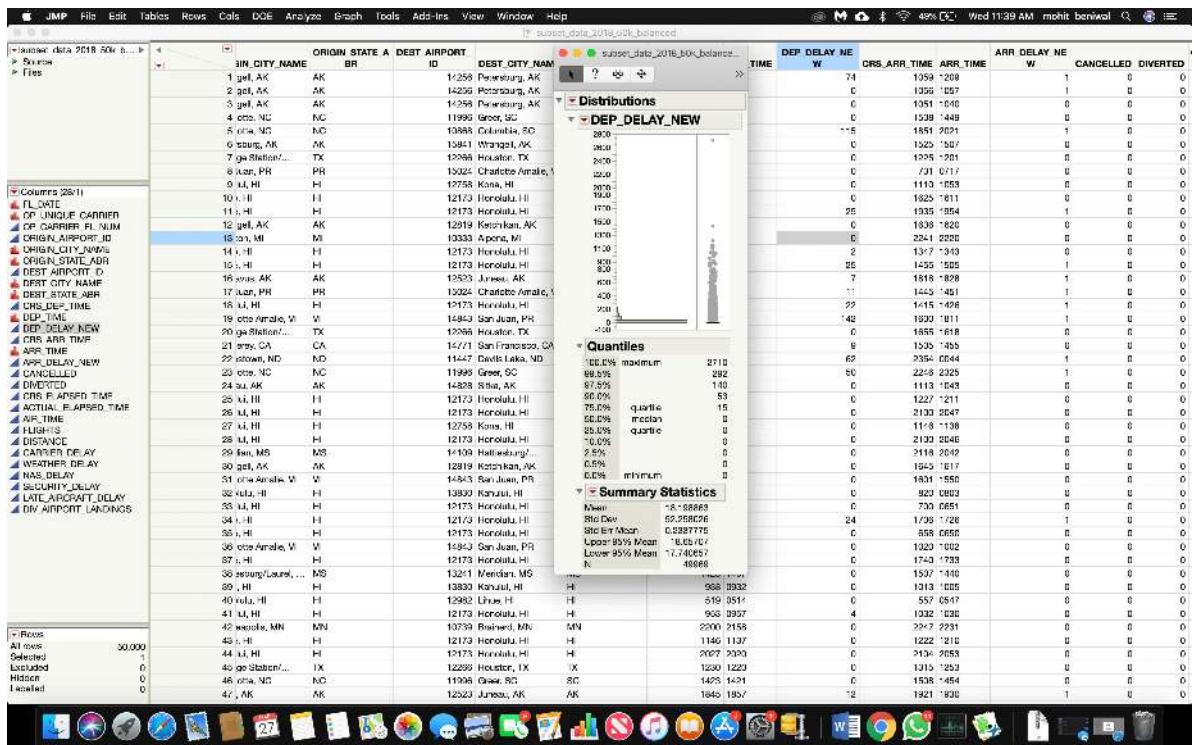
By looking at the data we realized that the YEAR column is not giving any added information as we already have FL\_DATE column. So, we have decided to delete the YEAR column.

Right Click -> Delete Column

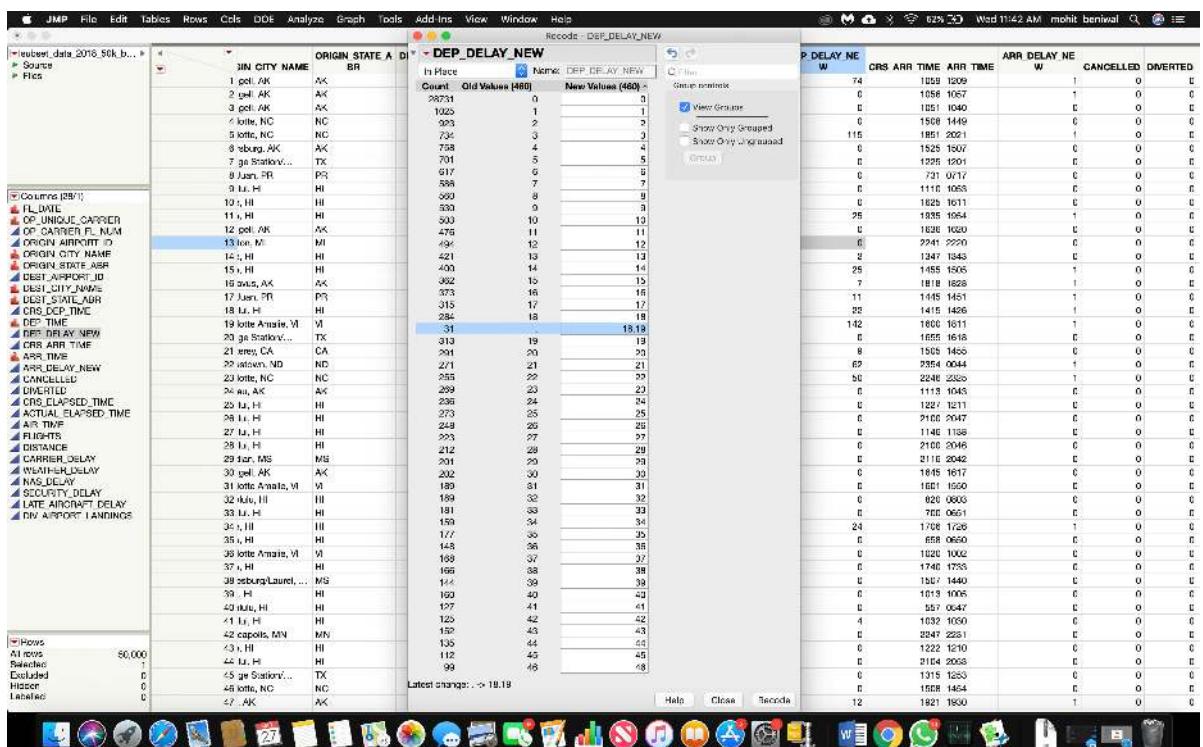
## 4.6 Handling Missing Data

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.



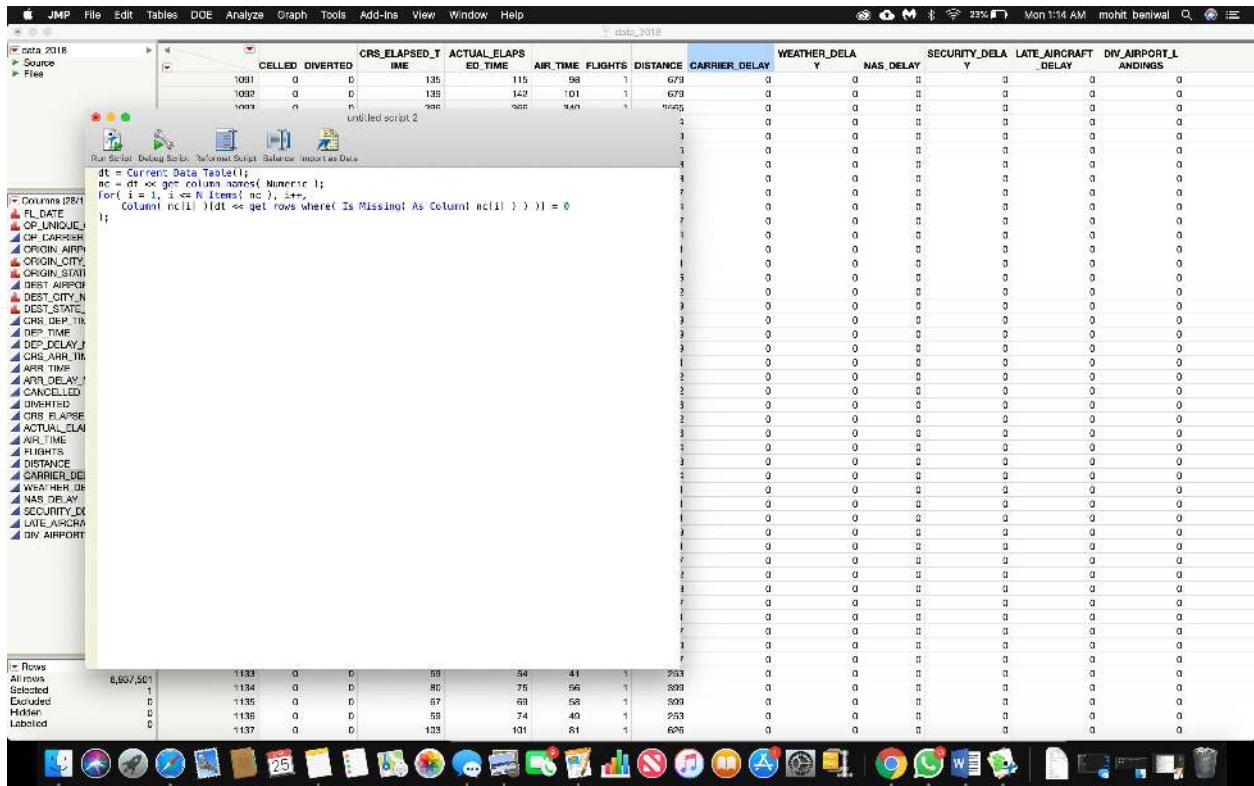
As we can see that the mean is 18.19, we will set the missing value with the mean.



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 original prefs
Eval( Parse( "pref(" || current_pref || ")" ) );

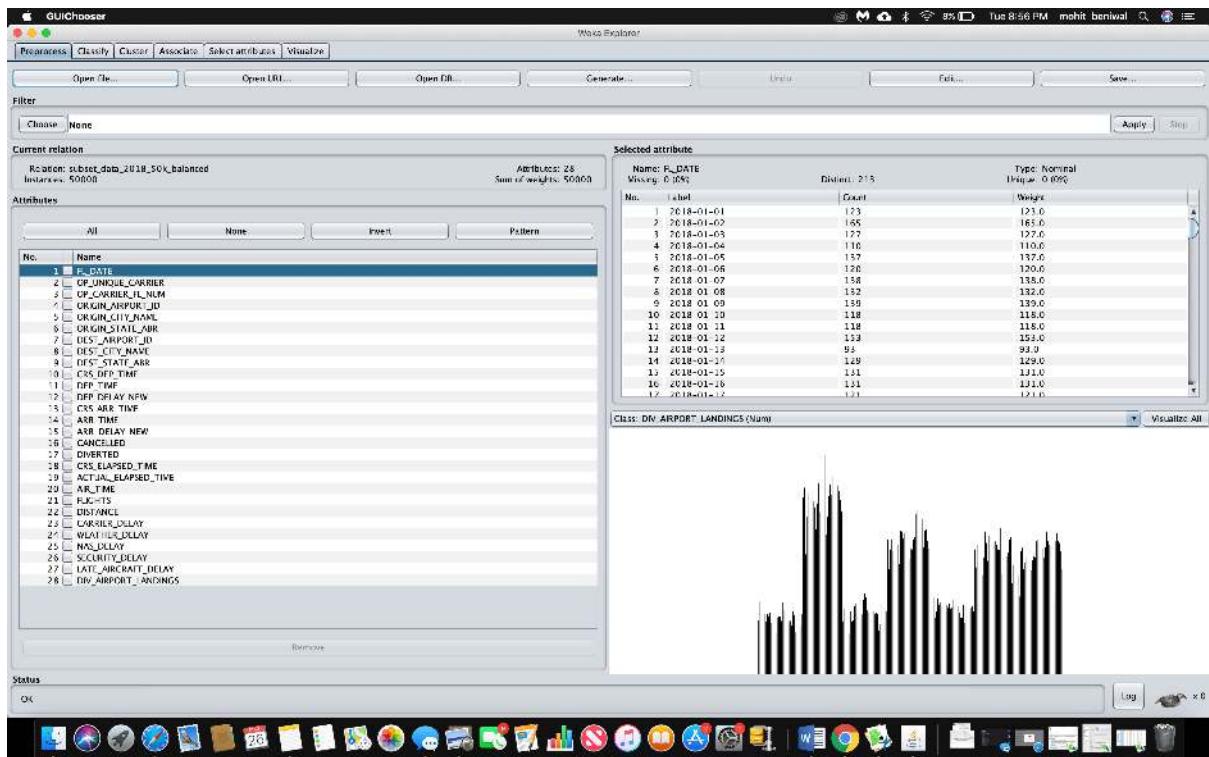
```

CITY_NAME	ORIGIN_STATE_A	DEST_AIRPORT_ID	DEST_CITY_NAME	DEST_STATE_AB	CRS_DEP_TIME	DEP_TIME	DEP_DELAY_NE		
FL	CD	13336 Chicago, IL	IL	1457	1511	-4			
	CO	11292 Denver, CO	CO	649	693	5			
CA	CA	14747 Seattle, WA	WA	1903	1914	-1			
AZ	AZ	12721 Boston, MA	MA	1711	1655	0			
NY	NY	14831 San Jose, CA	CA	1733	1725	0			
IA	MA	14100 Philadelphia, PA	PA	2117	2124	7			
DeLand, FL	TX	11270 Washington, DC	VA	1125	1139	3			
L	L	11288 Dallas/Fort ...	TX	1553	1553	0			
is, MN	MN	11775 Sioux Falls, SD	SD	933	898	0			
I	AL	13232 Chicago, IL	IL	1533	1529	0			
Chs, UT	UT	10387 Atlanta, GA	GA	845	841	0			
D	CO	15916 St. Louis, MO	MO	2315	2319	3			
WA	WA	11282 Denver, CO	CO	715	714	0			
Springs, CO	CO	13330 Chicago, IL	IL	2333	2326	0			
newark/... WA	WA	11282 Denver, CO	CO	1215	1219	0			
U	NJ	11289 Newark, NJ	CO	1383	1388	5			
A	SA	12693 Los Angeles, CA	CA	1133	1204	-14			
	MD	10381 Milwaukee, WI	WI	1025	1119	-4			
TS	TX	15140 Albuquerque, NM	NM	830	868	0			
JX	NC	11287 Charlotte, NC	NC	1239	1159	0			
	TX	13851 Oklahoma City, OK	OK	849	844	-4			
ny, MD	MD	13796 Oakland, CA	CA	833	816	-16			
L	L	11433 Detroit, MI	MI	1743	1741	-1			
ne, IA	IA	12181 Houston, TX	TX	2020	2001	-1			
KY	KY	10801 Baltimore, MD	MD	755	753	0			
MO	MO	13385 New Orleans, LA	LA	833	801	0			
,SD	SD	13330 Chicago, IL	IL	1375	1373	0			
IX	TX	11279 Washington, DC	VA	950	902	0			
IL	FL	13303 Miami, FL	FL	1030	957	0			
va, PA	PA	10387 Atlanta, GA	GA	1835	1801	0			
ted, VI	VI	13262 Miami, FL	FL	945	890	0			
ansle, FL	FL	13336 Chicago, IL	IL	1313	1339	0			
unian, NC	NC	14100 Philadelphia, PA	PA	737	745	0			
H	H	11288 Dallas/Fort ...	TX	1845	1808	0			
F	TX	10423 Austin, TX	TX	1055	1052	0			
ca, CA	CA	12478 New York, NY	NY	933	928	0			
		10387 Atlanta, GA	GA	910	811	-1			
30 2018-01-01 DL		14749 Honolulu, HI	HI	2288	2222	0			
30 2018-01-01 DL		12173 Honolulu, HI	HI	15018 St. Louis, MO	MO	2213	2159	0	
40 2018-01-01 DL		13087 Atlanta, GA	GA	1453	1448	0			
41 2018-01-01 DL		12982 Los Angeles, CA	CA	12264 Washington, DC	VA	2133	2145	0	
42 2018-01-01 DL		14399 10387 Atlanta, GA	GA	11282 Denver, CO	CO	943	899	0	
43 2018-01-01 DL		11433 Detroit, MI	MI	10387 Atlanta, GA	GA	1543	1541	0	
44 2018-01-01 DL		22007 12982 Los Angeles, CA	CA	14747 Seattle, WA	WA	1033	1001	-1	
45 2018-01-01 DL		23803 13087 Atlanta, GA	GA	12270 Wichita, KS	KS	913	913	0	
46 2018-01-01 DL		24683 10387 Atlanta, GA	GA	11618 Newark, NJ	NJ	1837	1801	0	
47 2018-01-01 DL		10921							

Click Run .

## 5 Data Loading into Weka

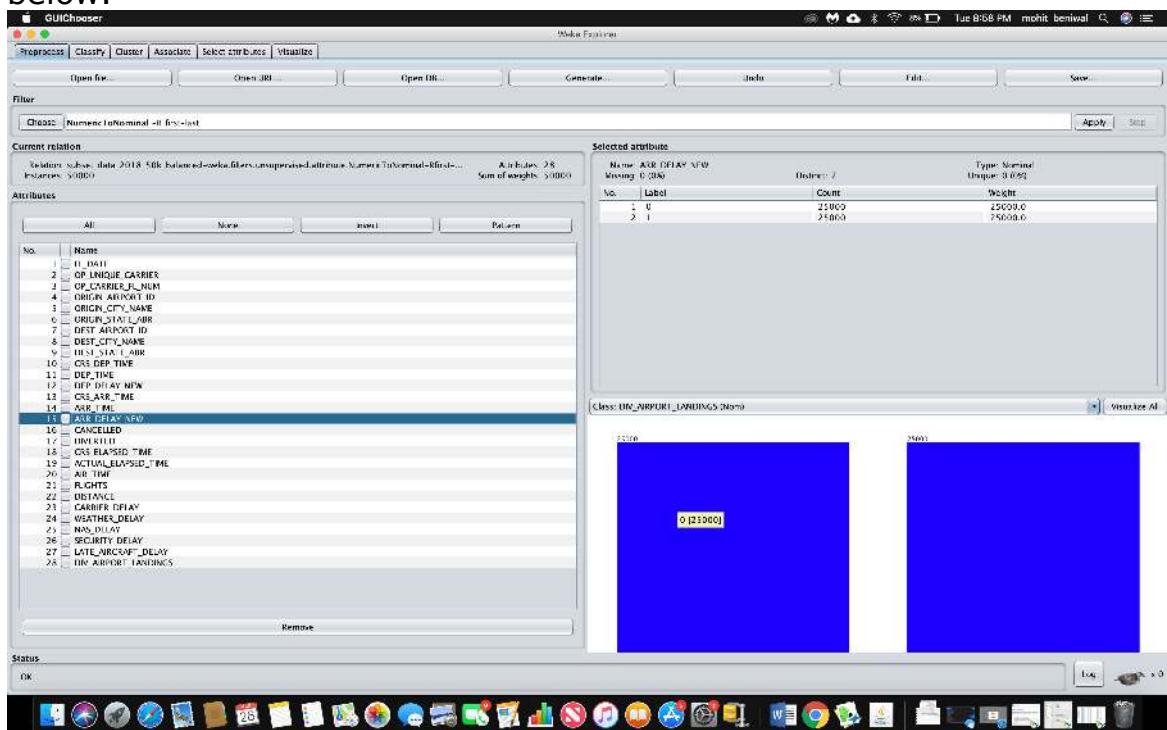
We imported the subset data with 50,000 tuples into Weka.



## 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.

The screenshot shows the 'Selected attribute' panel in the GUI Chooser application. The 'Type: Nominal' section shows 'Missing: 0 (0%)' and 'Weight' column values. The 'ARR\_DELAY\_NEW' attribute is highlighted in blue. A context menu is open over the table, with the 'Add instance...' option highlighted.

No.	Name	ARR_DELAY_NEW	Actual	Nominal	Count	Weight
1	FL_DATE	1900	-1914	16	18	2.18
2	OP_UNIQUE_CARRIER	1711	-851	0	1057	11.80
3	OP_CARRIER_FL_NUM	2310	-2353	0	645	6.28
4	ORIGIN_AIRPORT_ID	1738	-1723	0	2120	20.9
5	ORIGIN_CITY_NAME	PA	-2117	7	2255	22.3
6	ORIGIN_STATE_AIR	VA	-1135	3	1315	13.45
7	DEST_AIRPORT_ID	72	-1810	0	1446	14.9
8	DEST_CITY_NAME	SD	-961	8	1055	10.8
9	DEP_DELAY_NEW	1810	-1500	0	1055	10.8
10	CRS_DEP_TIME	GA	-645	6	857	8.57
11	DEP_TIME	MD	-2015	3	17	24.00
12	DEP_DELAY	CD	-715	0	327	8.19
13	DEP_DELAY_NEW	IL	-2240	0	505	5.05
14	ARR_DELAY	CG	-1215	0	1305	12.6
15	ARR_DELAY_NEW	CO	-1350	0	1711	16.1
16	CANCELLED	CA	-1150	0	1317	13.0
17	DIVERTED	MI	-1090	1	1090	1.09
18	DEP_DELA迟延	TX	-2028	0	2313	2.31
19	ACTUAL_ELAPSED_TIME	TS	-2028	0	2313	2.31
20	ARR_TIME	LA	-630	1	820	8.2
21	FLIGHTS	IL	-1125	0	1629	16.29
22	DISTANCE	VA	-657	0	1118	11.18
23	CARRIER_DELAY	FL	-1008	0	1133	11.33
24	METHTER_DELAY	GA	-1205	0	2037	20.37
25	MSL_DELAY	FL	-843	0	1057	10.57
26	SECURITY_DELAY	FL	-843	0	1057	10.57
27	LATE_AIRCRAFT_DELAY	FL	-843	0	1057	10.57
28	DIV_AIRPORT_LANDINGS	FL	-843	0	1057	10.57
29	ARR_DELAY_NEW	FL	-843	0	1057	10.57

## 5.2 Data Wrangling

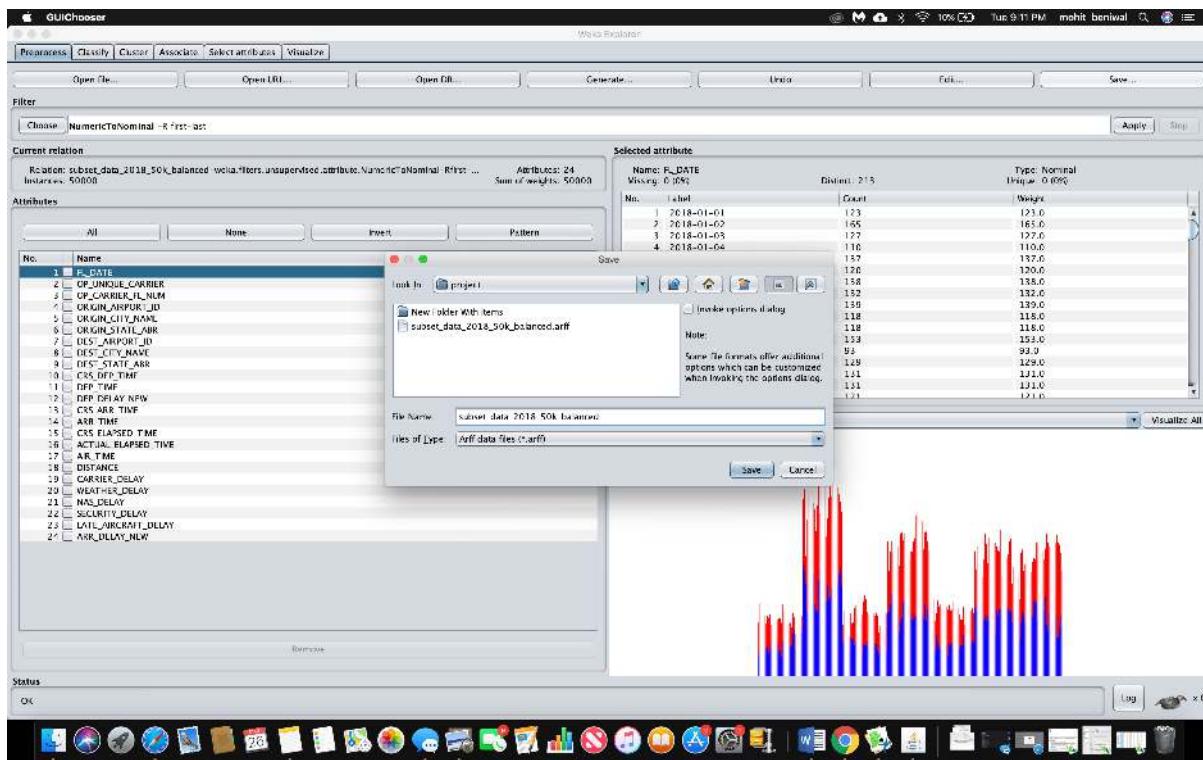
Now after looking through all the attributes we found that the CANCELLED, DIVERTED, FLIGHTS and DIV\_AIRPORT\_LANDINGS have only single values either '0' or '1'. Hence, we will delete them.

The screenshot shows the 'Selected attribute' panel in the GUI Chooser application. The 'Type: Nominal' section shows 'Missing: 0 (0%)' and 'Weight' column values. The 'CANCELLED' attribute is highlighted in blue. A context menu is open over the table, with the 'Add instance...' option highlighted.

No.	Name	Count	Weight
1	CANCELLED	50000	50000.0

### 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.

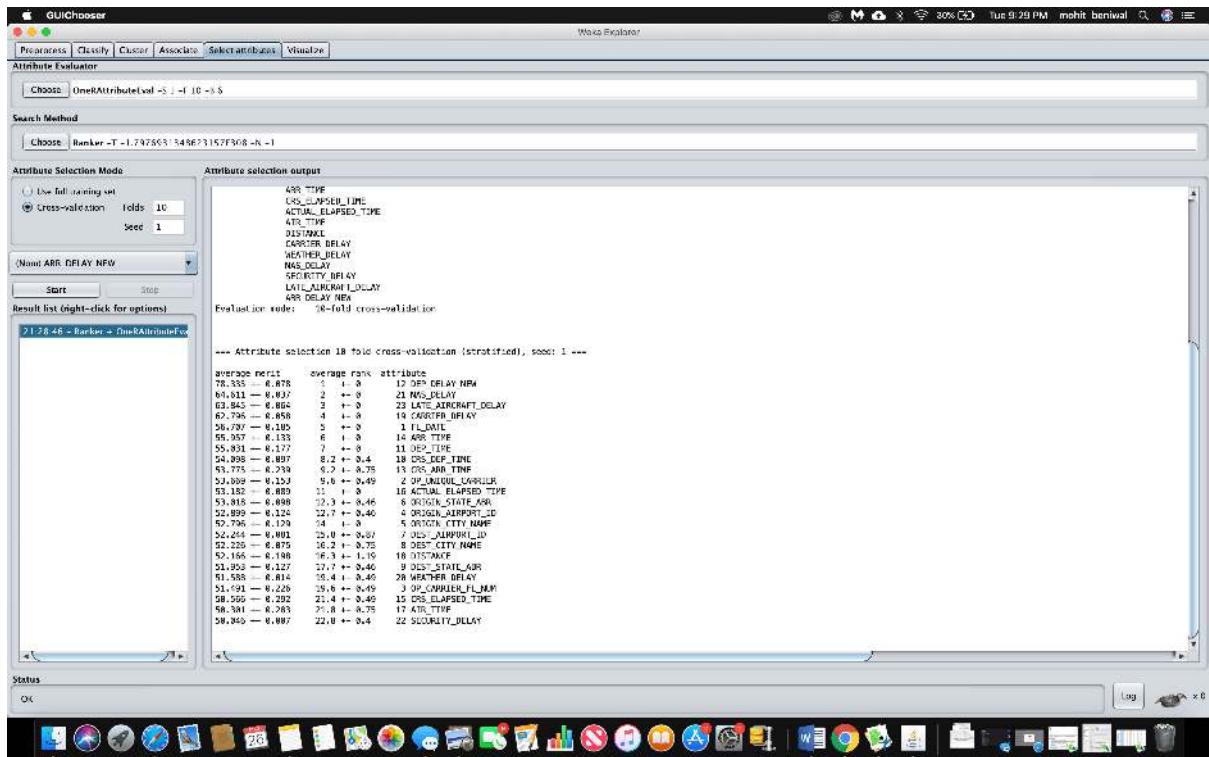


## 6 Attribute Selection

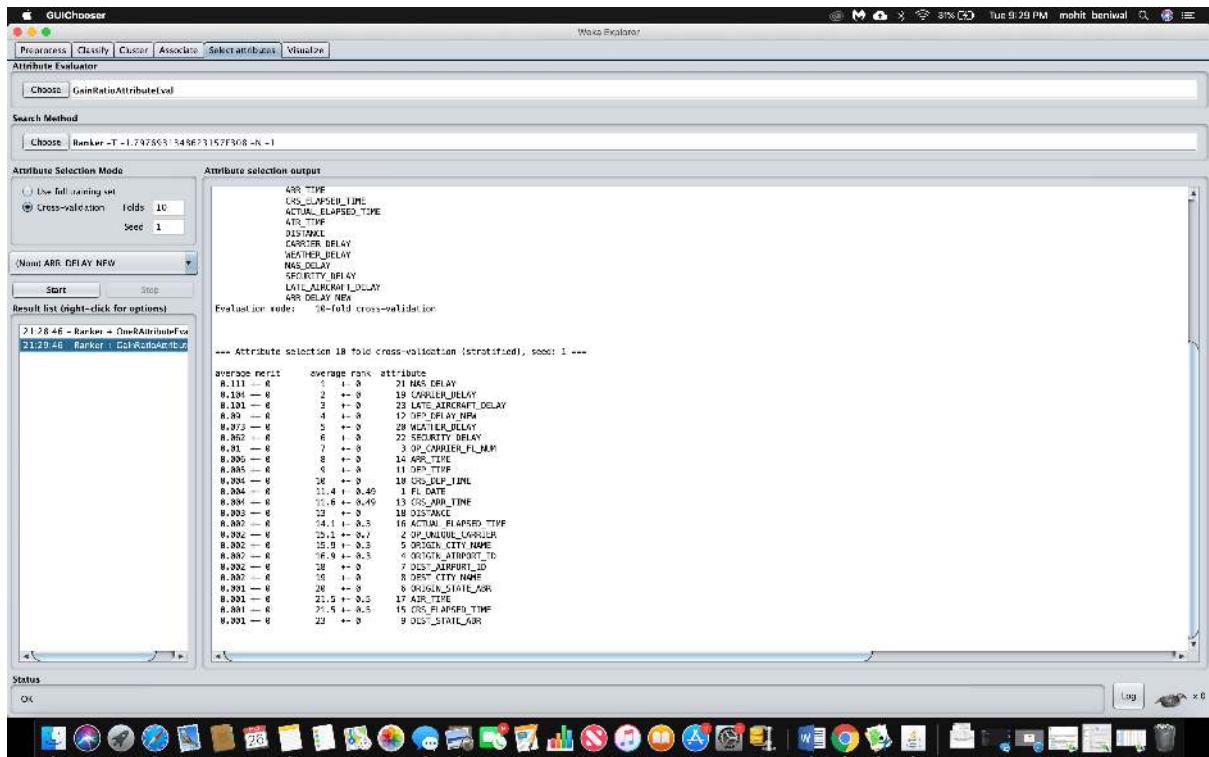
After running through all the available attribute selection algorithms, we chose these 5 evaluations to predict best set of pairs which would be considered for our model building. Here we chose Ranker Search method for the selection.

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

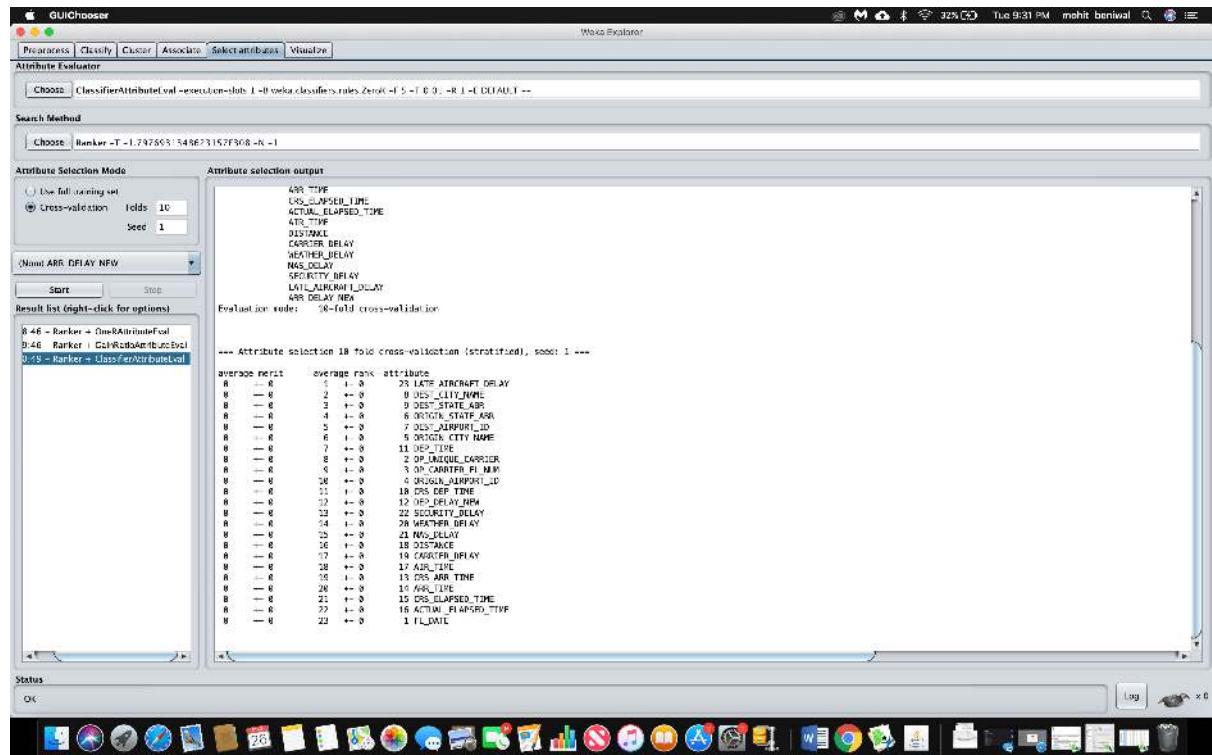
## 6.1 OneR Attribute Evaluation



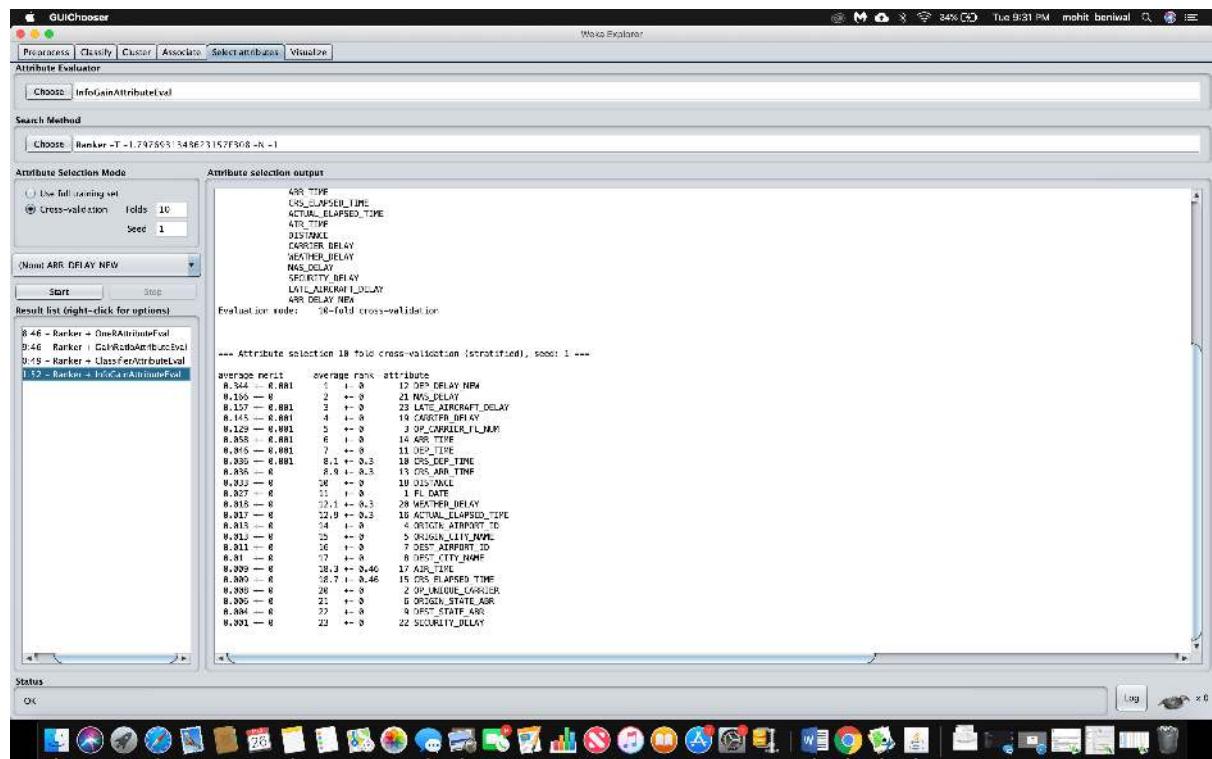
## 6.2 Gain Ratio Attribute Evaluation



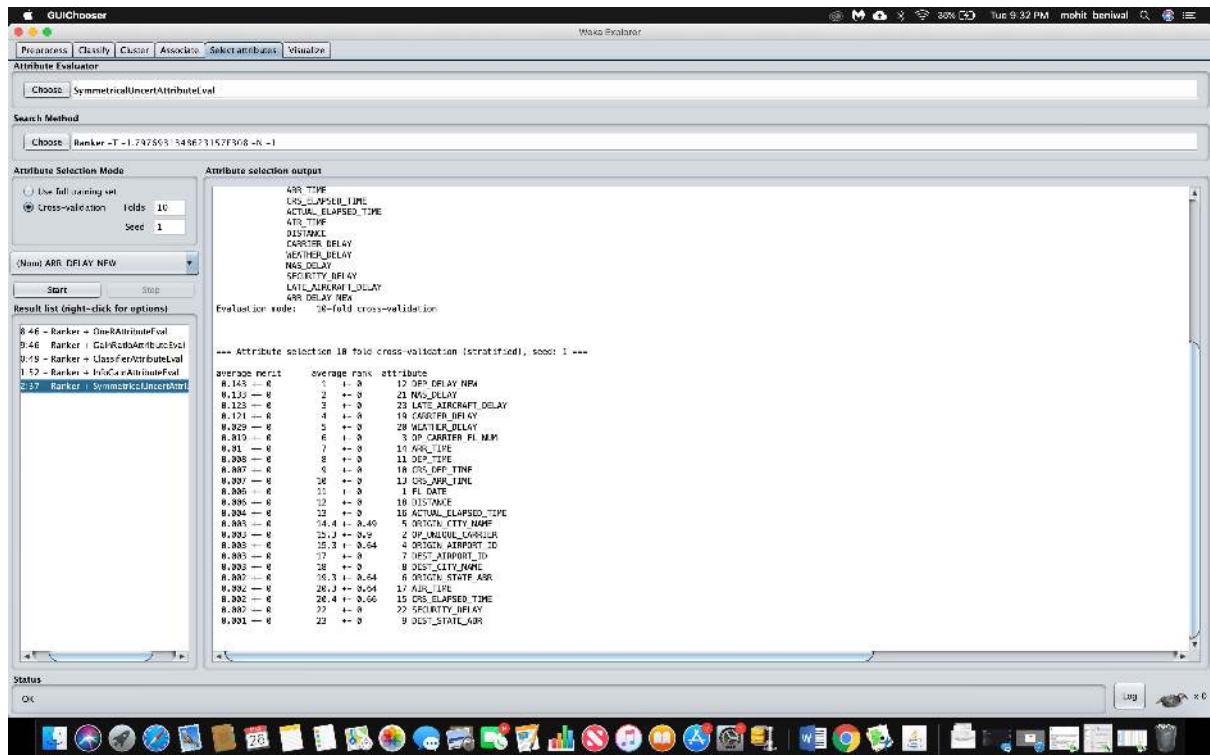
## 6.3 Classifier Attribute Evaluation



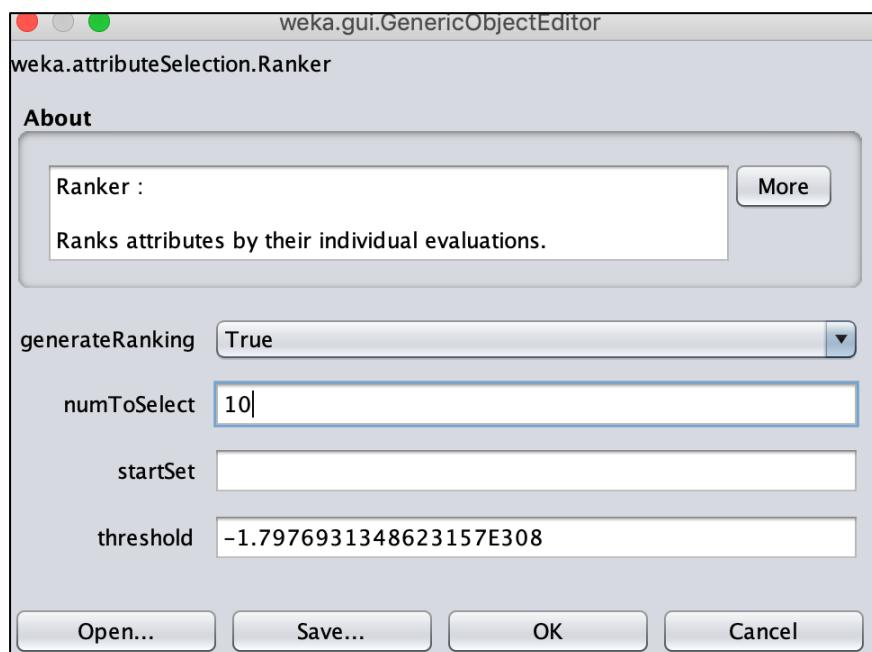
## 6.4 Info Gain Attribute Evaluation



## 6.5 SymmetricalUncert Attribute Evaluation



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



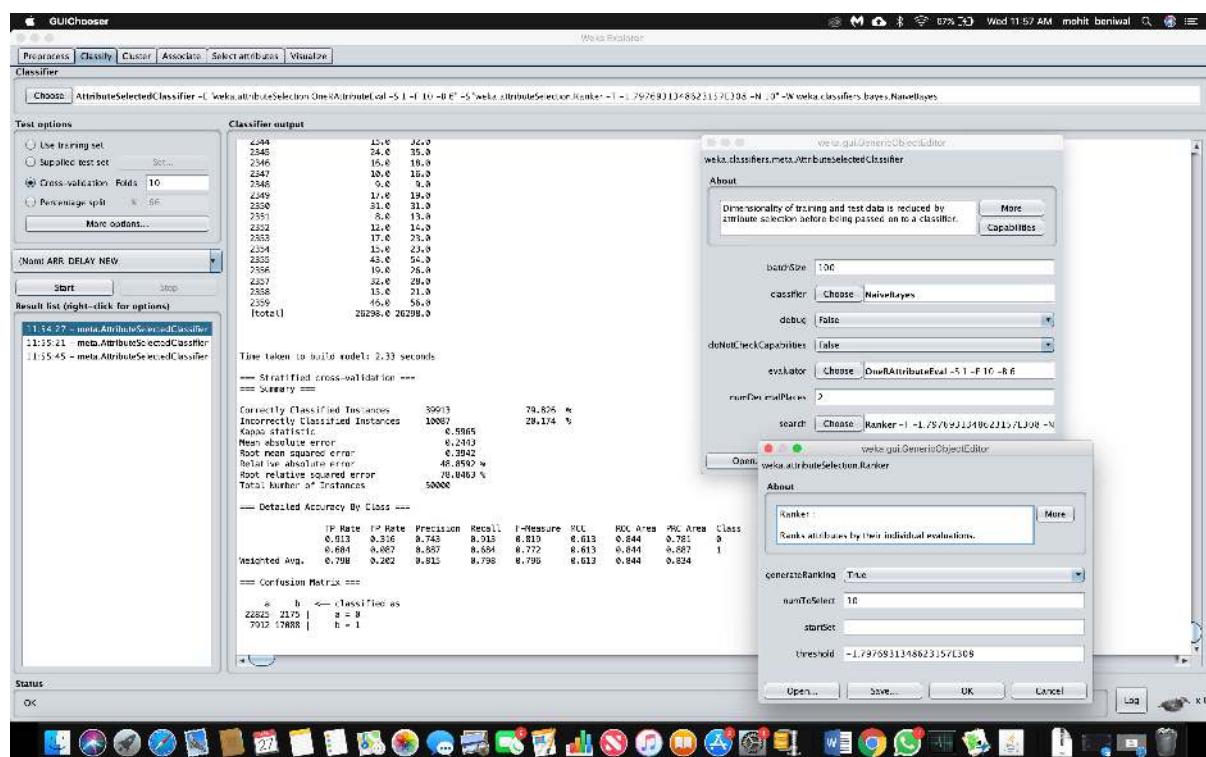
## 7 Classification

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.

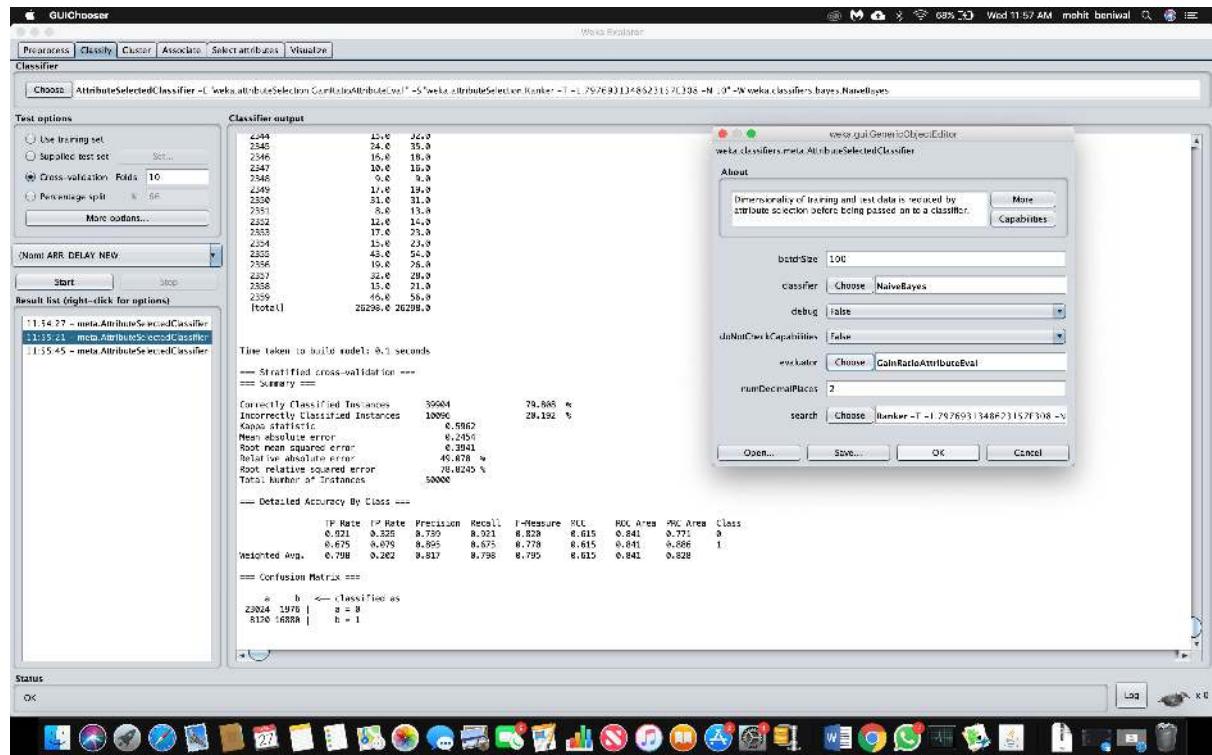
### 7.1 Naïve Bayes Algorithm

Naïve 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. Naïve Bayes classifiers have been especially popular for text classification and are a traditional solution for classification problems.

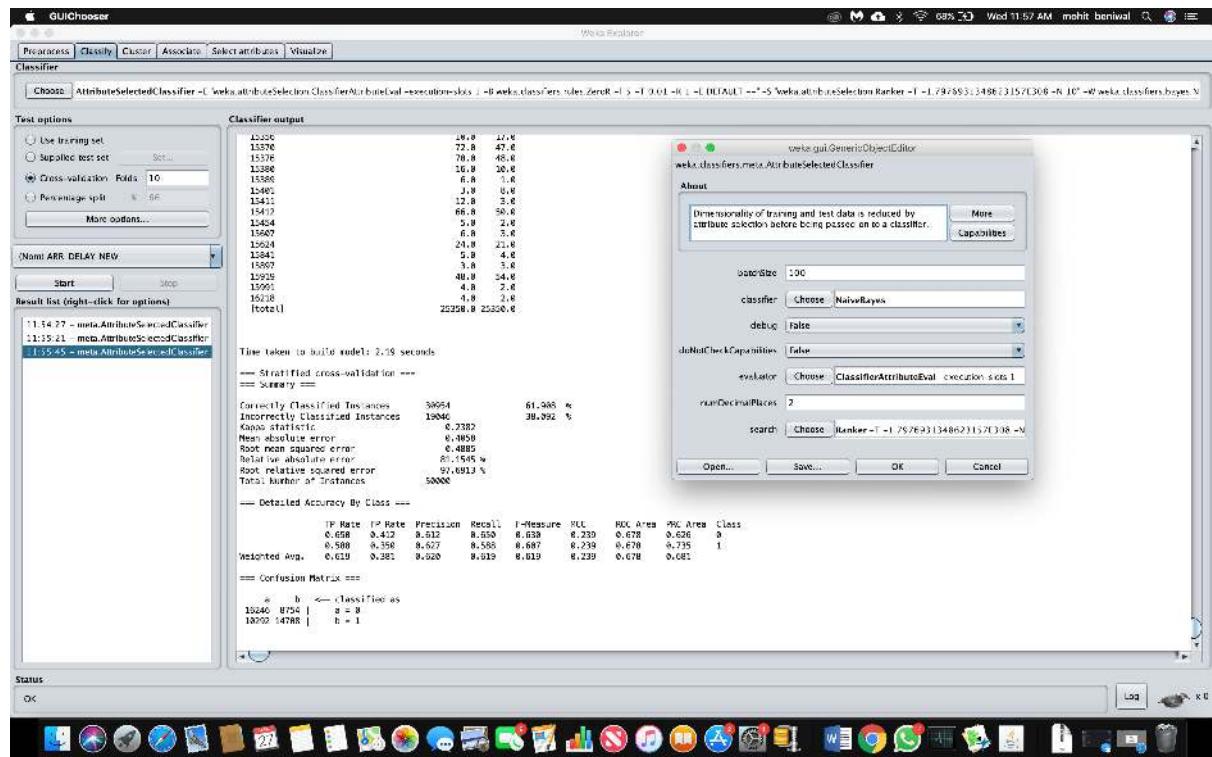
#### 7.1.1 Using OneR Attribute Evaluation



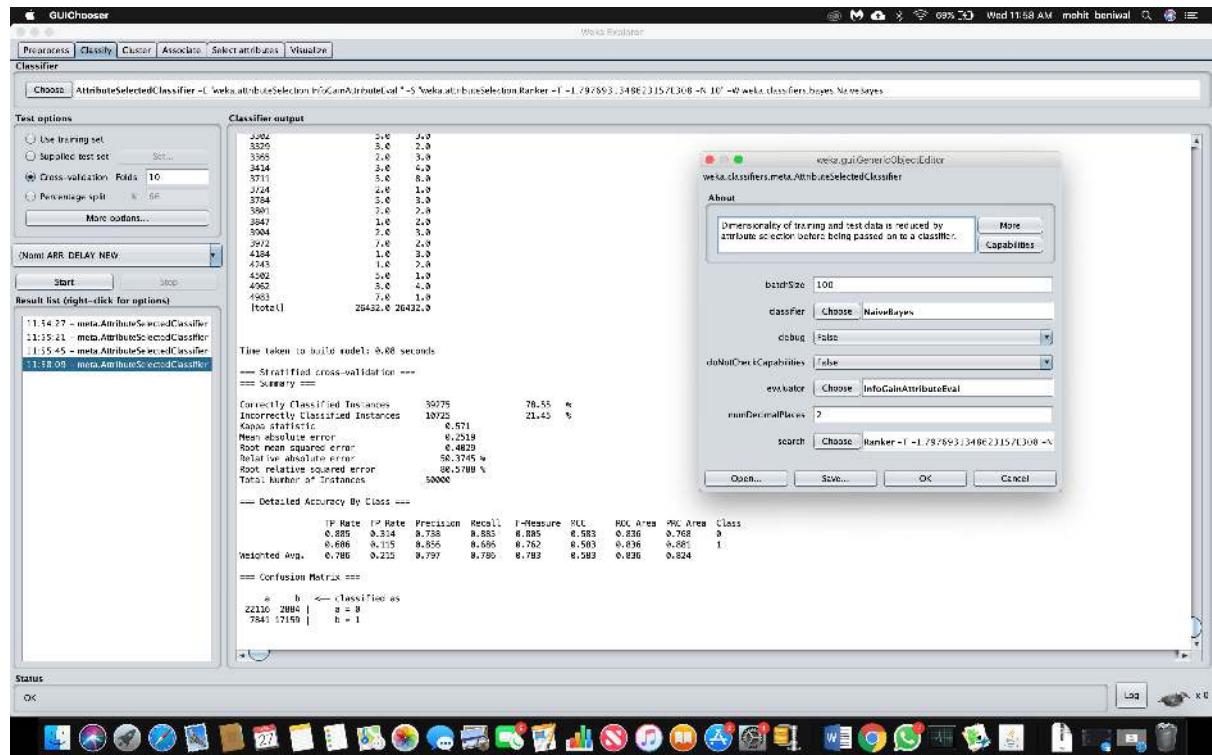
## 7.1.2 Using Gain Ratio Attribute Evaluation



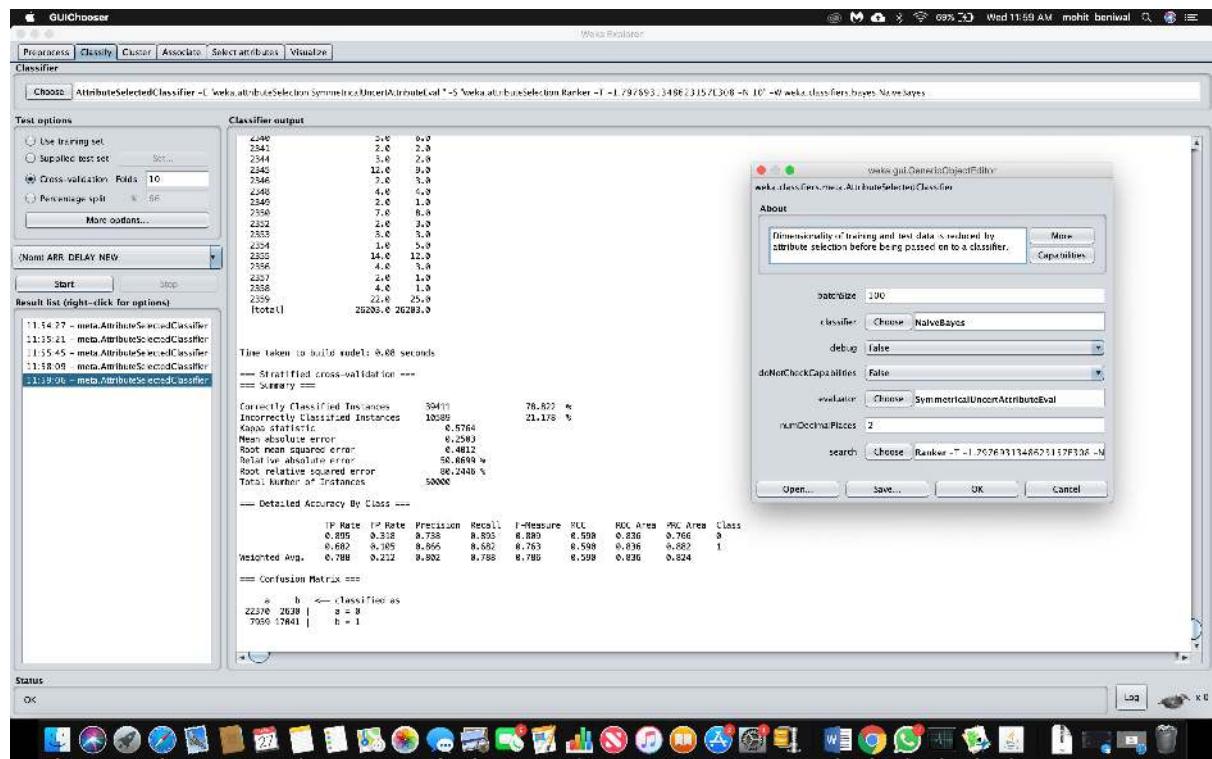
## 7.1.3 Using Classifier Attribute Evaluation



## 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

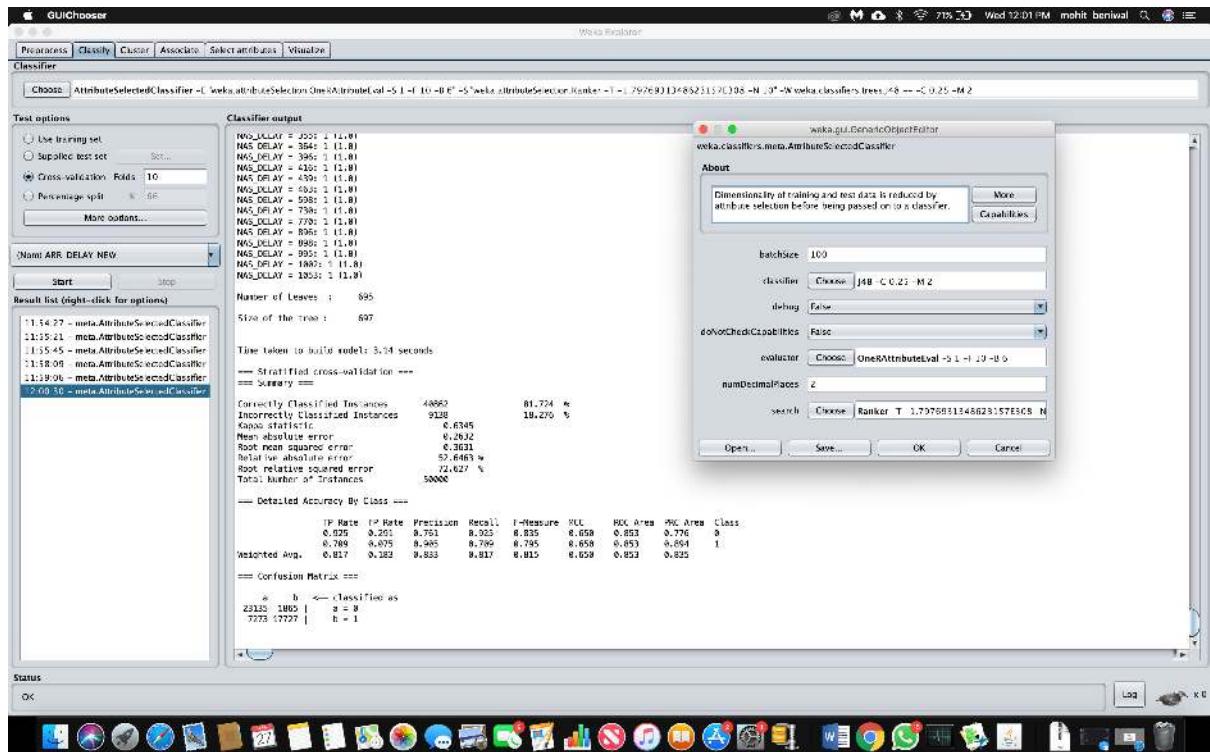
The best model based on ROC is OneR with 0.844.

## 7.2 J48 Algorithm

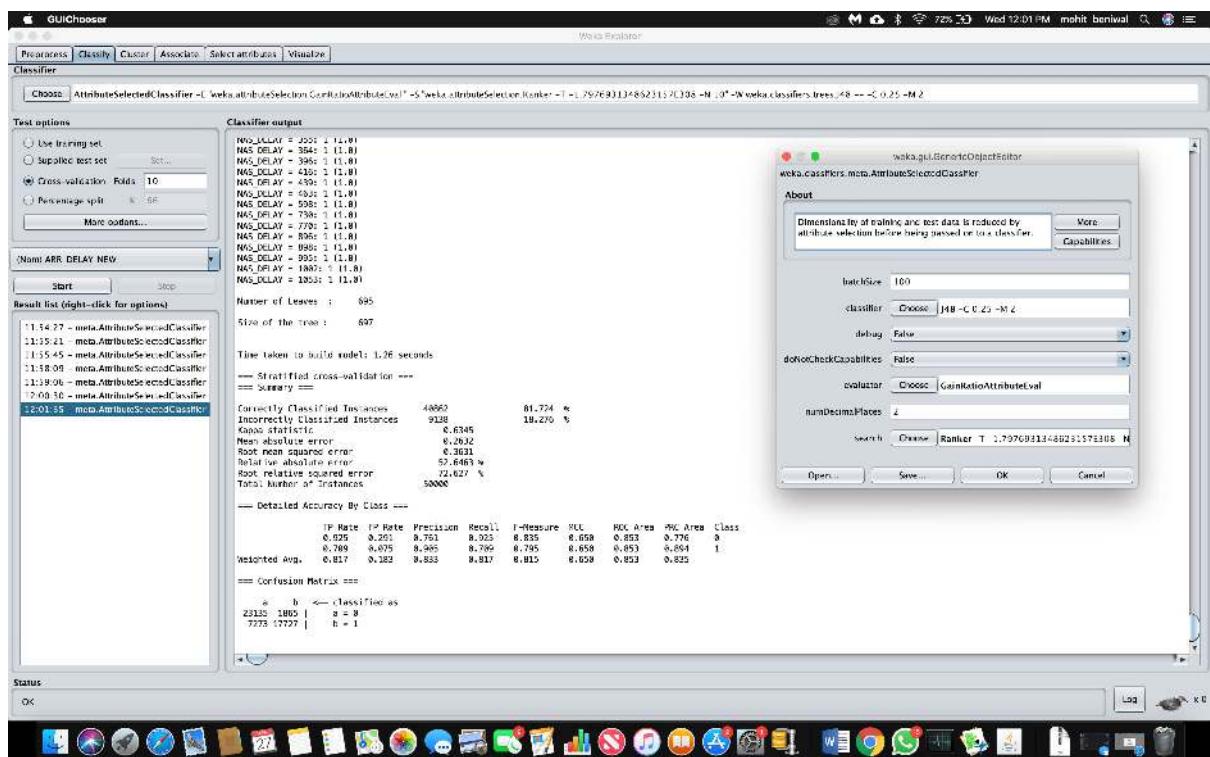
Decision Tree Algorithm is to find out the way the attributes-vector behaves for a number of instances. Also, on the bases of the training instances the classes for the newly generated instances are being found. This algorithm generates the rules for the prediction of the target variable. With the help 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.

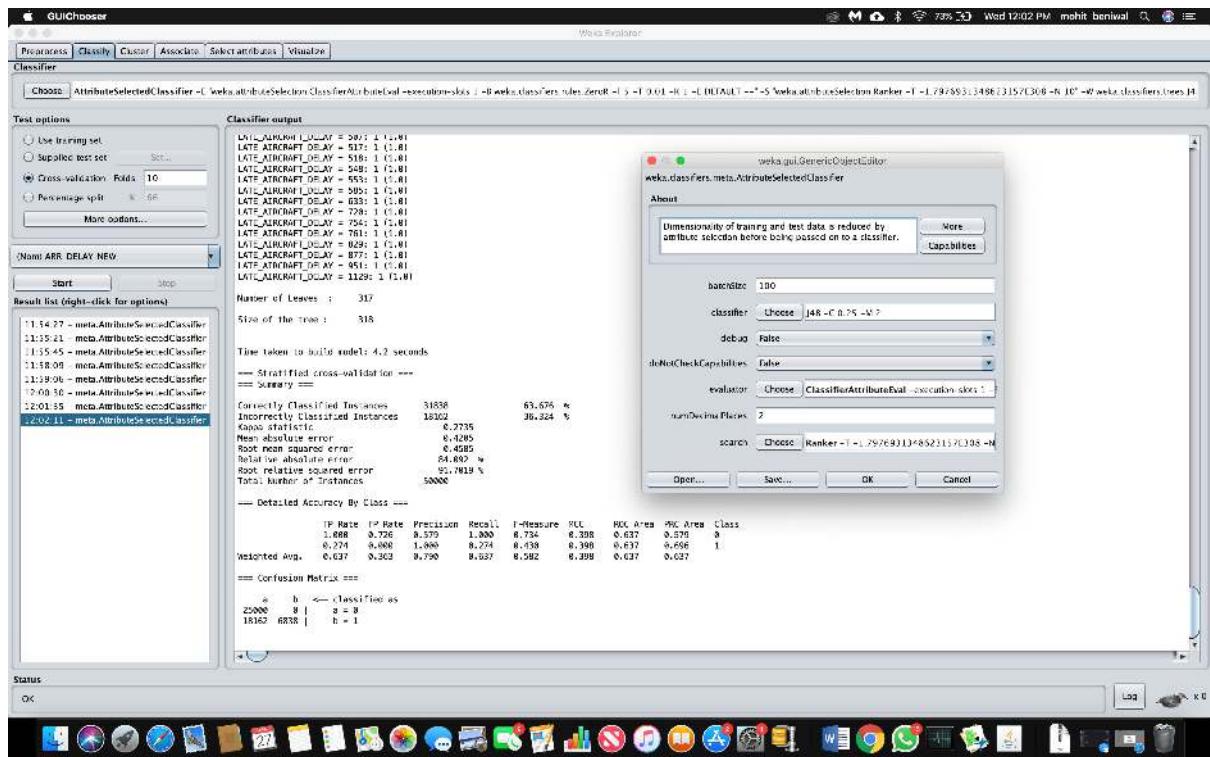
## 7.2.1 Using OneR Attribute Evaluation



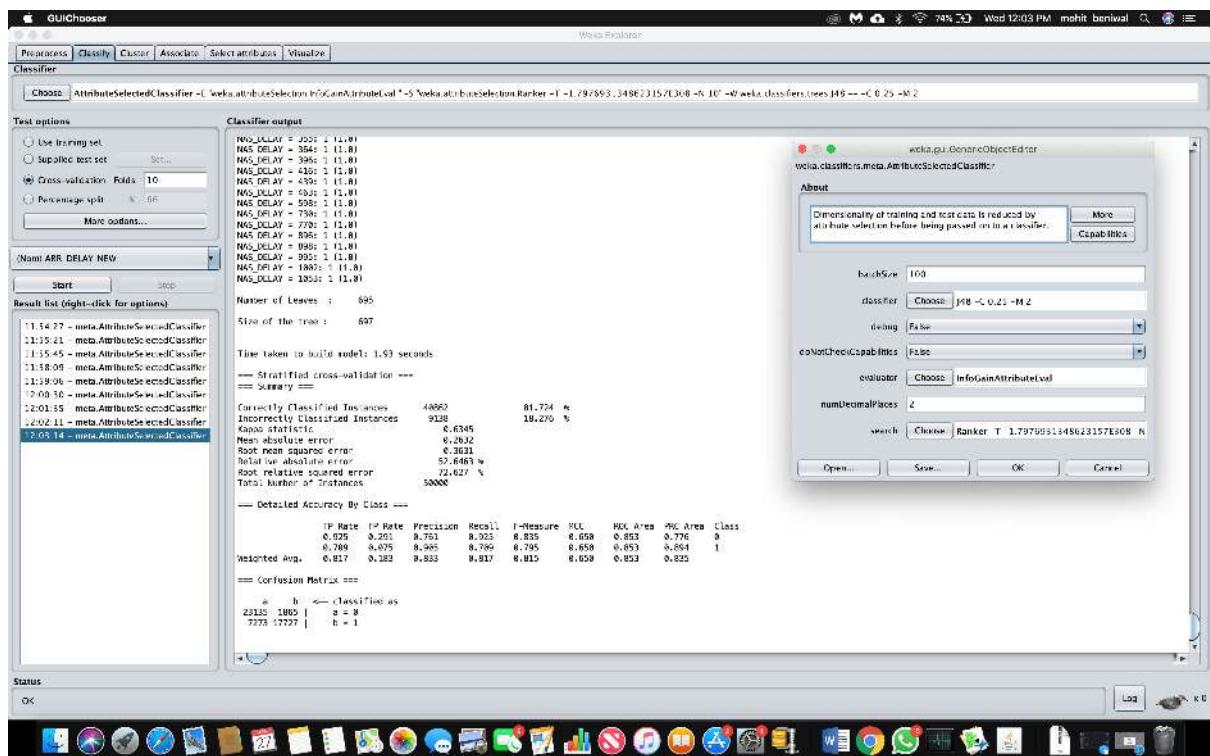
## 7.2.2 Using Gain Ratio Attribute Evaluation



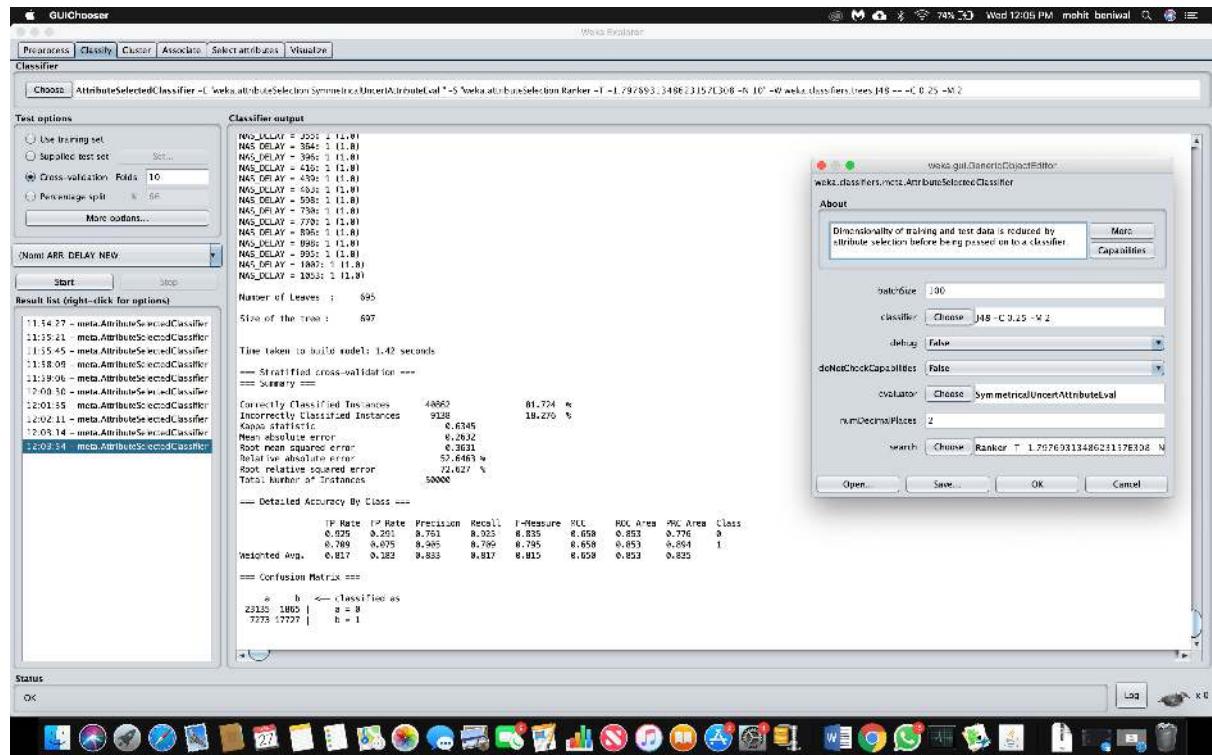
## 7.2.3 Using Classifier Attribute Evaluation



## 7.2.4 Using InfoGain Attribute Evaluation



## 7.2.5 Using SymmetricalUncert Attribute Evaluation



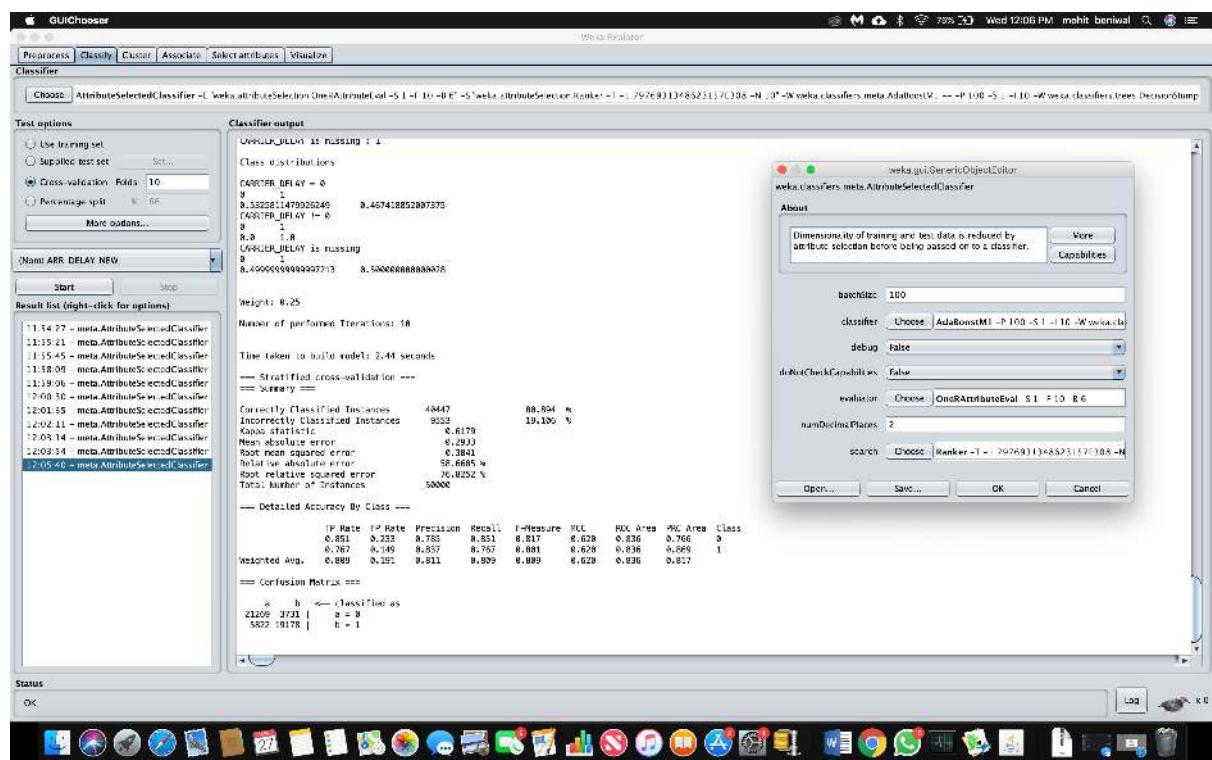
Attribute selection Algorithm	Correctly Classified Instances %	TP Rate	FP Rate	ROC Area	F-Measure	Precision	RMS Error
1R	81.724	0.817	0.183	0.853	0.815	0.833	0.363
Gain Ratio	81.724	0.817	0.183	0.853	0.815	0.833	0.363
Classifier Attribute	63.676	0.637	0.363	0.637	0.582	0.790	0.458
Info Gain	81.724	0.817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

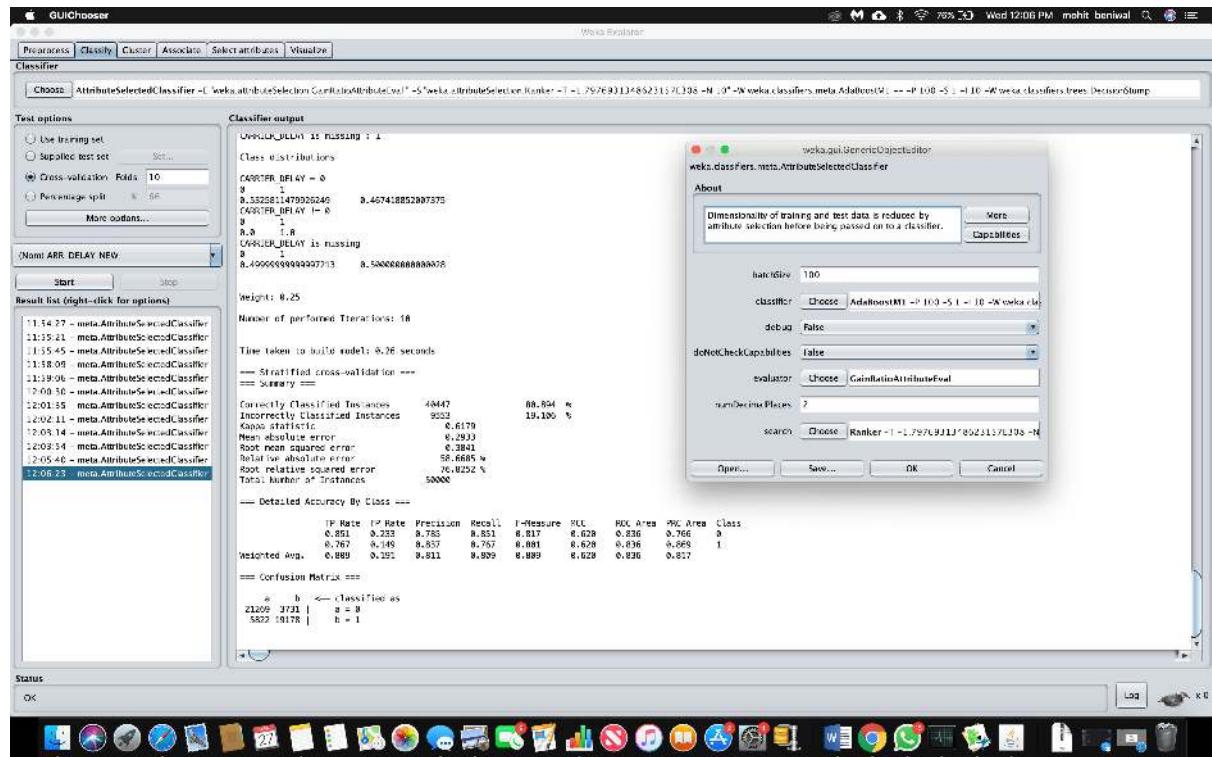
## 7.3 Adaboost M1 Algorithm

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.

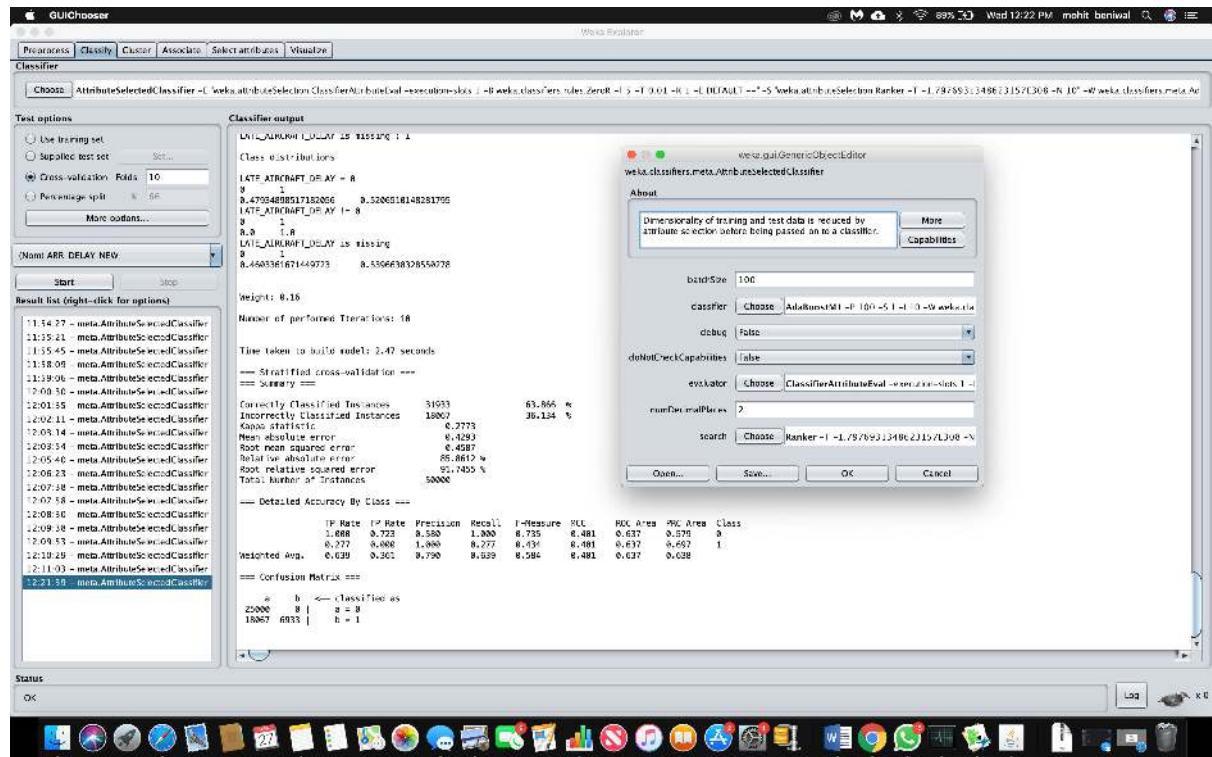
### 7.3.1 Using OneR Attribute Evaluation



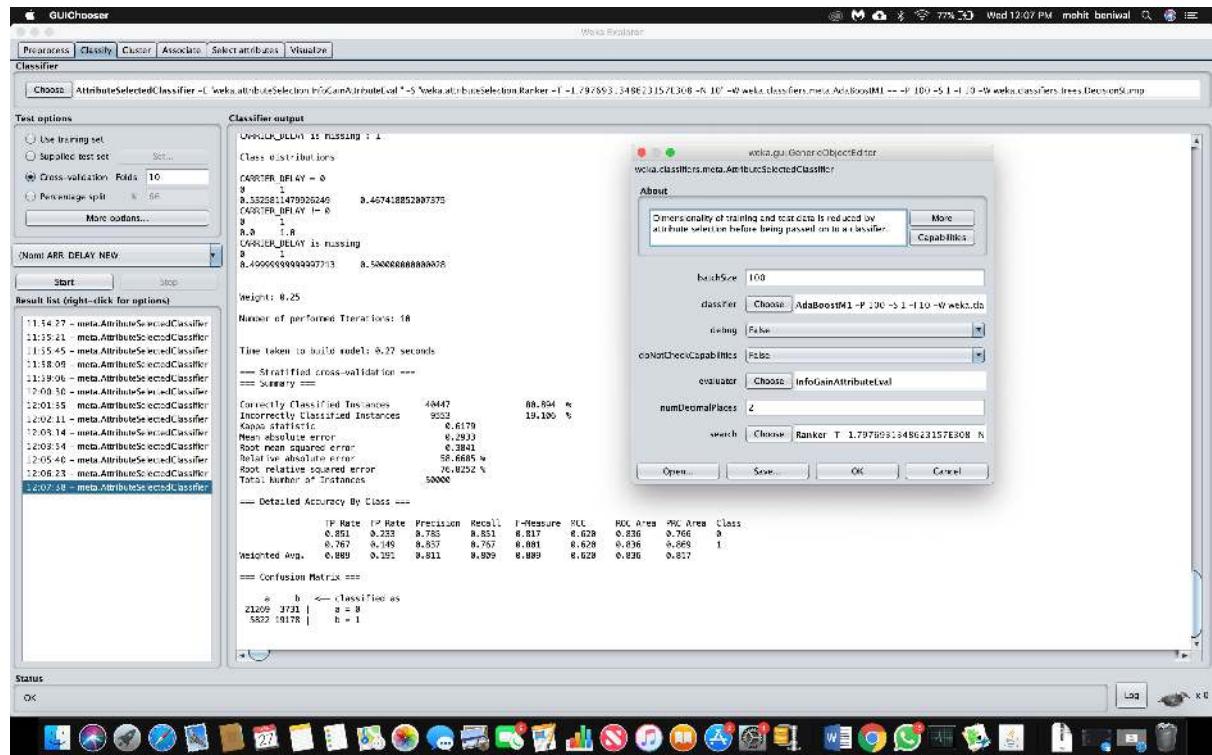
### 7.3.2 Using Gain Ratio Attribute Evaluation



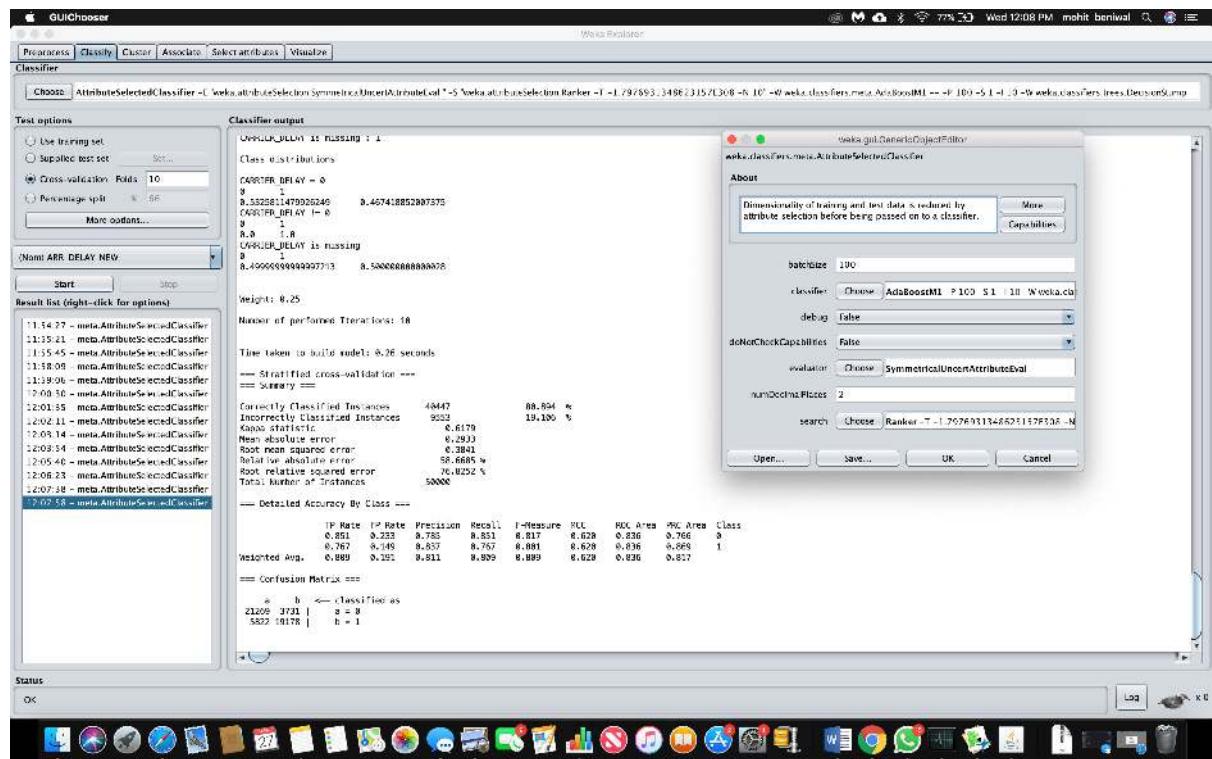
### 7.3.3 Using Classifier Attribute Evaluation



### 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.

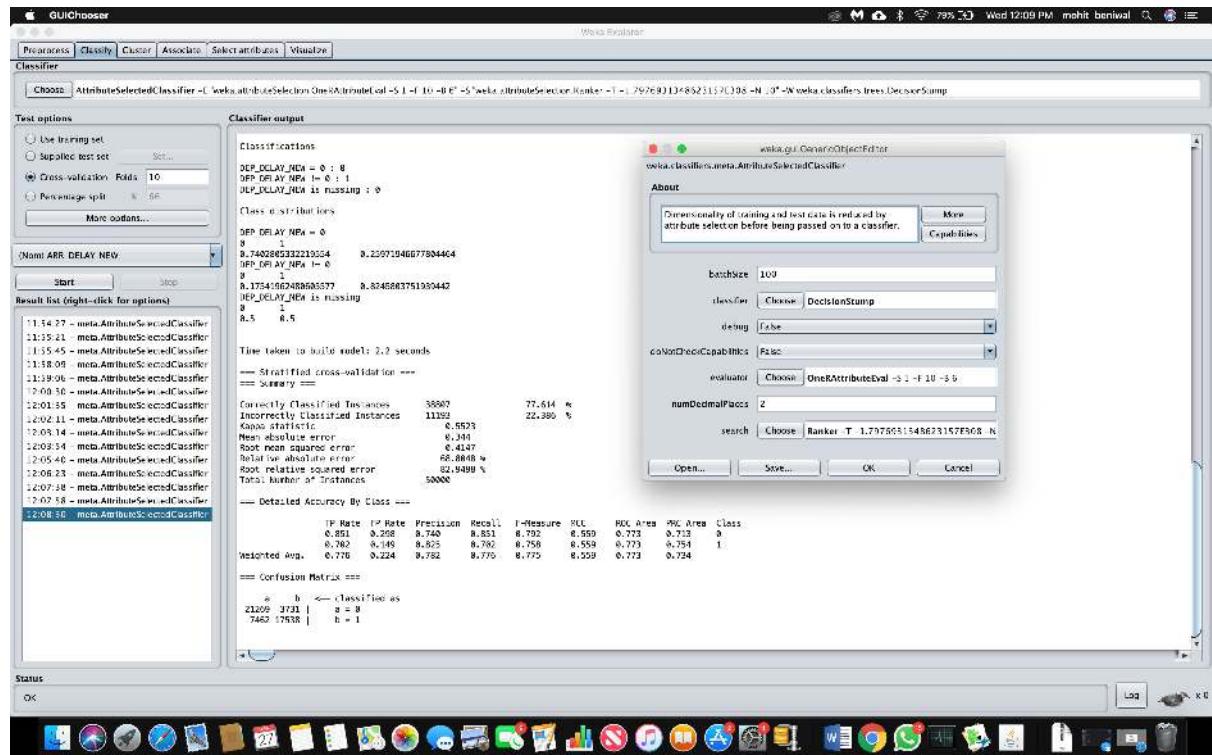
## 7.4 Decision Stump Algorithm

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.

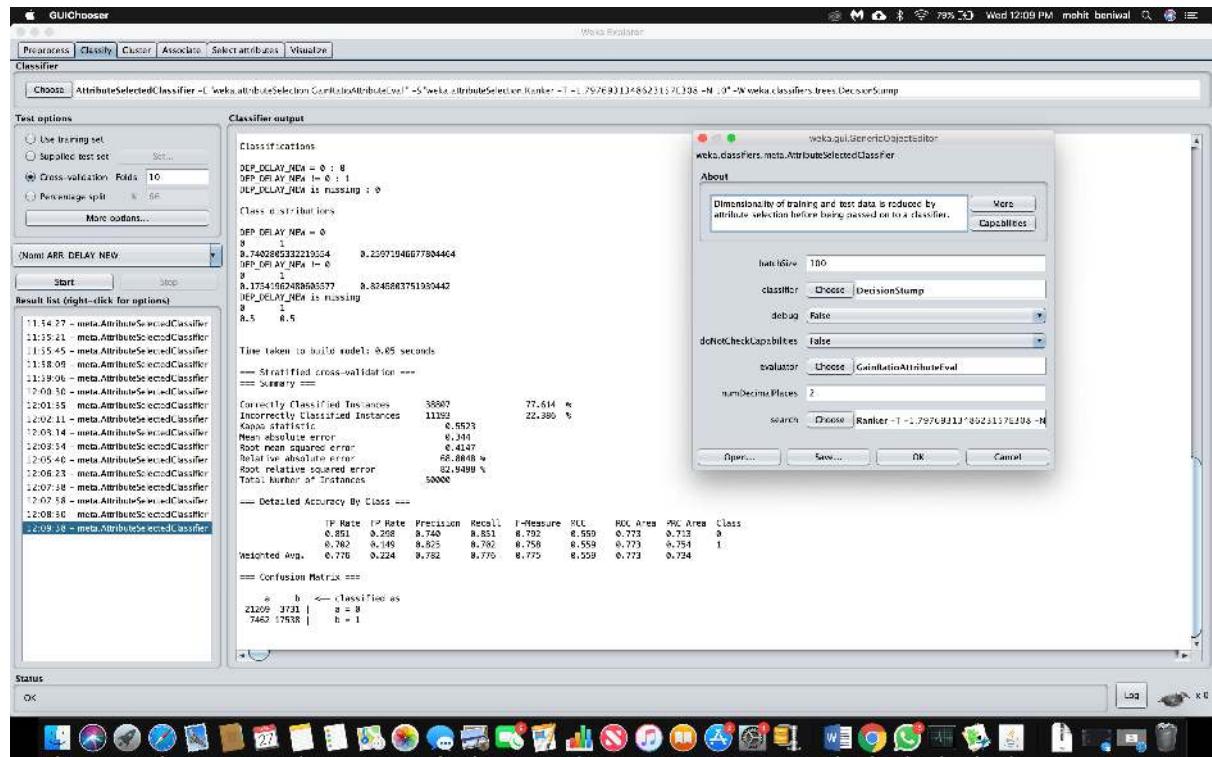
Depending on the type of the input feature, several variations are possible. For nominal features, one may build a stump which contains a leaf for each possible feature value or a stump with the two leaves, one of which corresponds to some chosen category, and the other leaf to all the other categories. 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.

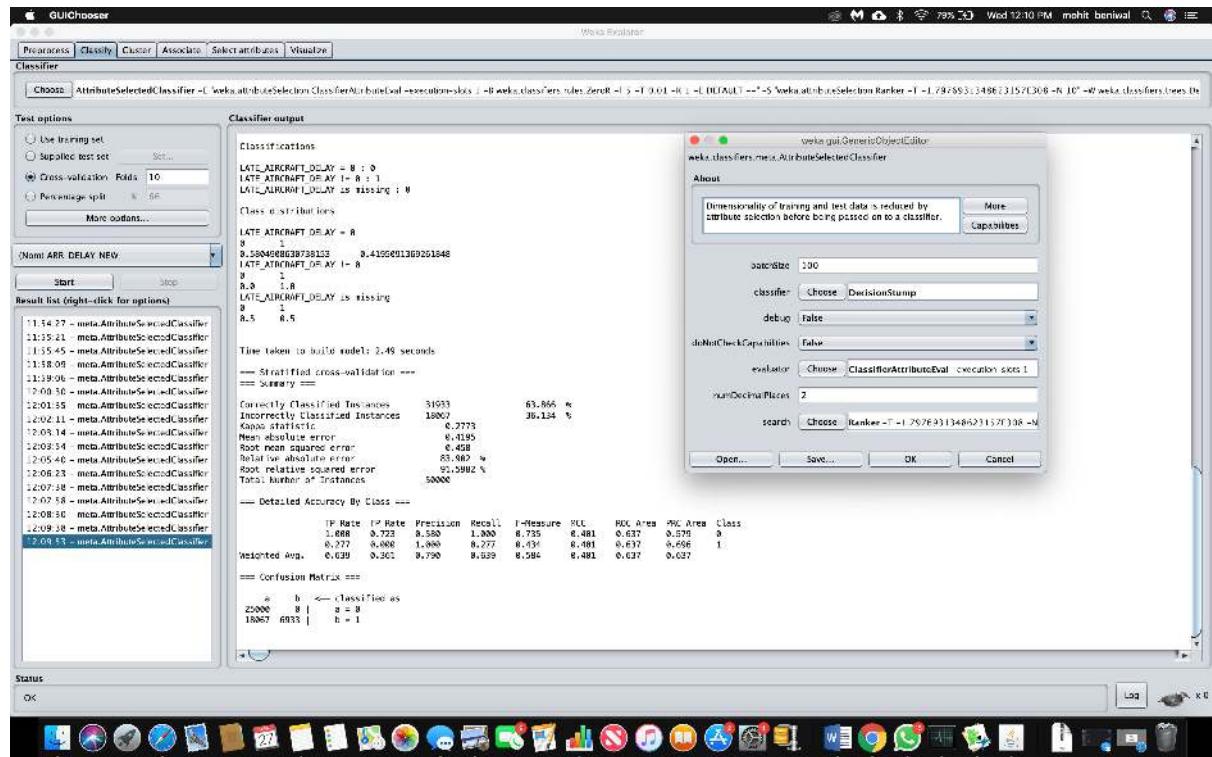
## 7.4.1 Using OneR Attribute Evaluation



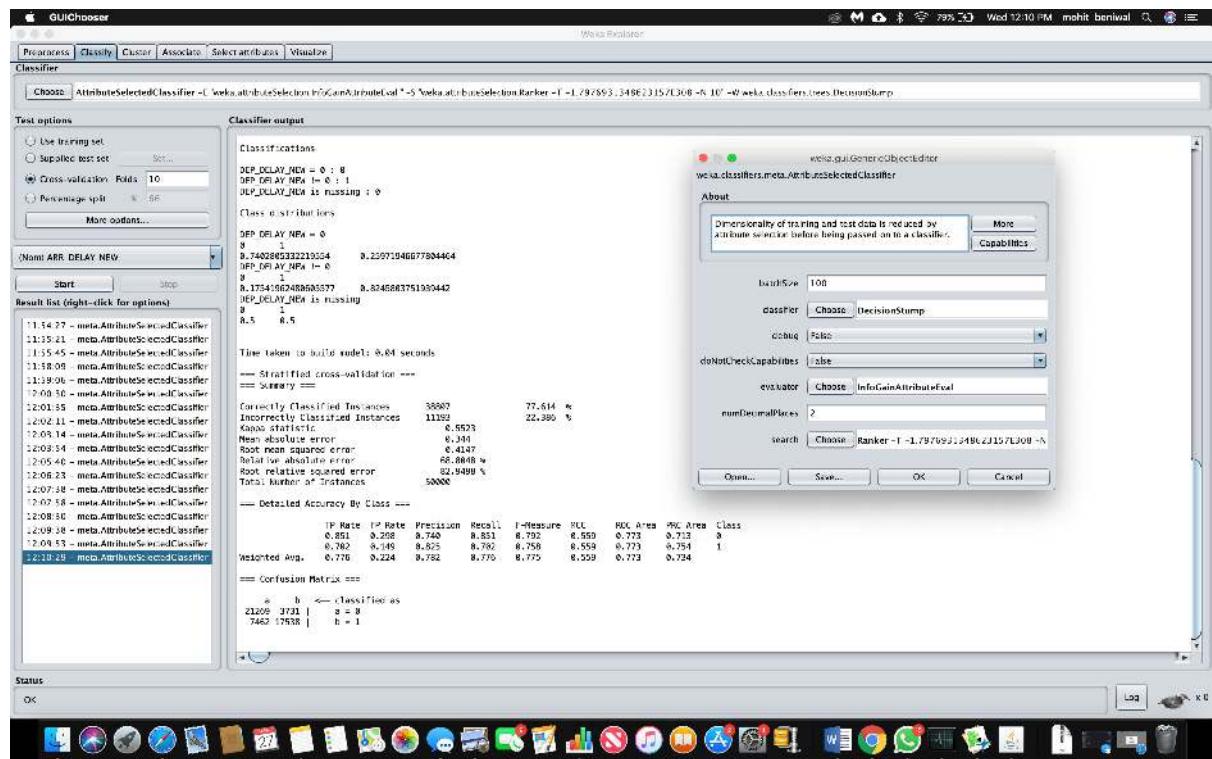
## 7.4.2 Using Gain Ratio Attribute Evaluation



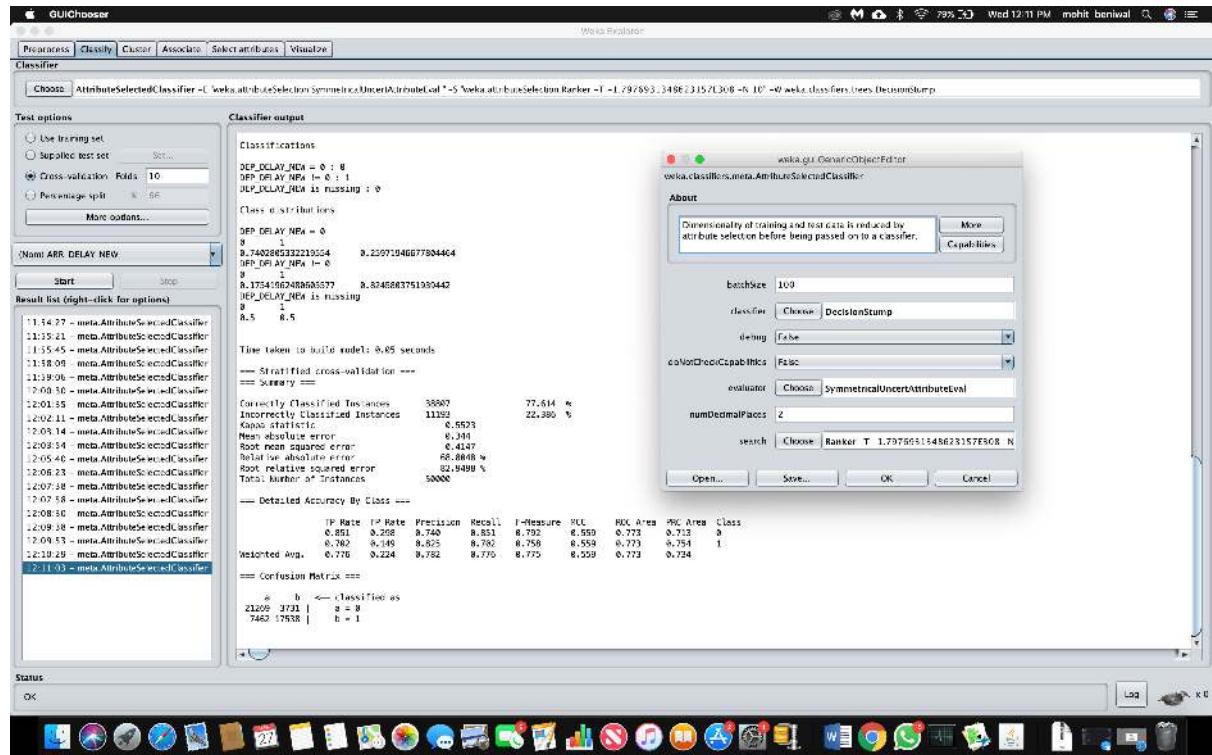
### 7.4.3 Using Classifier Attribute Evaluation



### 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](https://en.wikipedia.org/wiki/Decision_stump))