NAME: NISHI GUPTA
CSC 5800

PROJECT TOPIC:

Predict Student Performance in Secondary Education (high school)

INDEX

Nishi Gupta
Gu1978

OVERVIEW

Educational system is one of the important parts for development of any country. Most of the developed countries have their own educational system and evaluation criteria. In this work, we will analyze real world data from two Portuguese secondary schools. The problem statement is based on paper published at 5th future public technology conference held at Portugal during the year 2008.[1]

Modeling student performance is an important tool for both educators and students, since it can help a better understanding of this phenomenon and ultimately improve it. Although the educational level of the Portuguese population has improved in the last decades. In particular, lack of success in the core classes of Mathematics and the Portuguese language is extremely serious.

As mentioned in the paper[1], they have modeled under three DM goals:

i) Binary classification (pass/fail)

ii) Classification with five levels (from I very good or excellent to V - insufficient)

iii) Regression, with a numeric output that ranges between zero (0%) and twenty (100%)

Here, in this project we have classified students into three categories, "Good", "Fair", and "Poor", on the basis of their final exam performance and created various DM models to predict students' final performance and compared it.

DATASET OVERVIEW -

1. Dataset Information :-

We have used the Student Performance Dataset published by UCI Machine Learning Repository. The dataset has 395 rows and 33 features. We have used only Mathematics dataset for analysis in this project. Each sample consists of student’s background information and grades. Background information consists of some personal information of student (i.e. smoking, study time, etc.), family information of student (i.e. father-mother education, job, etc.). There are three columns of grades given in dataset; G1, G2 and G3. G1 and G2 are periodic grades throughout the year and G3 is the final grades for Mathematics.

File Name: student-mat.csv

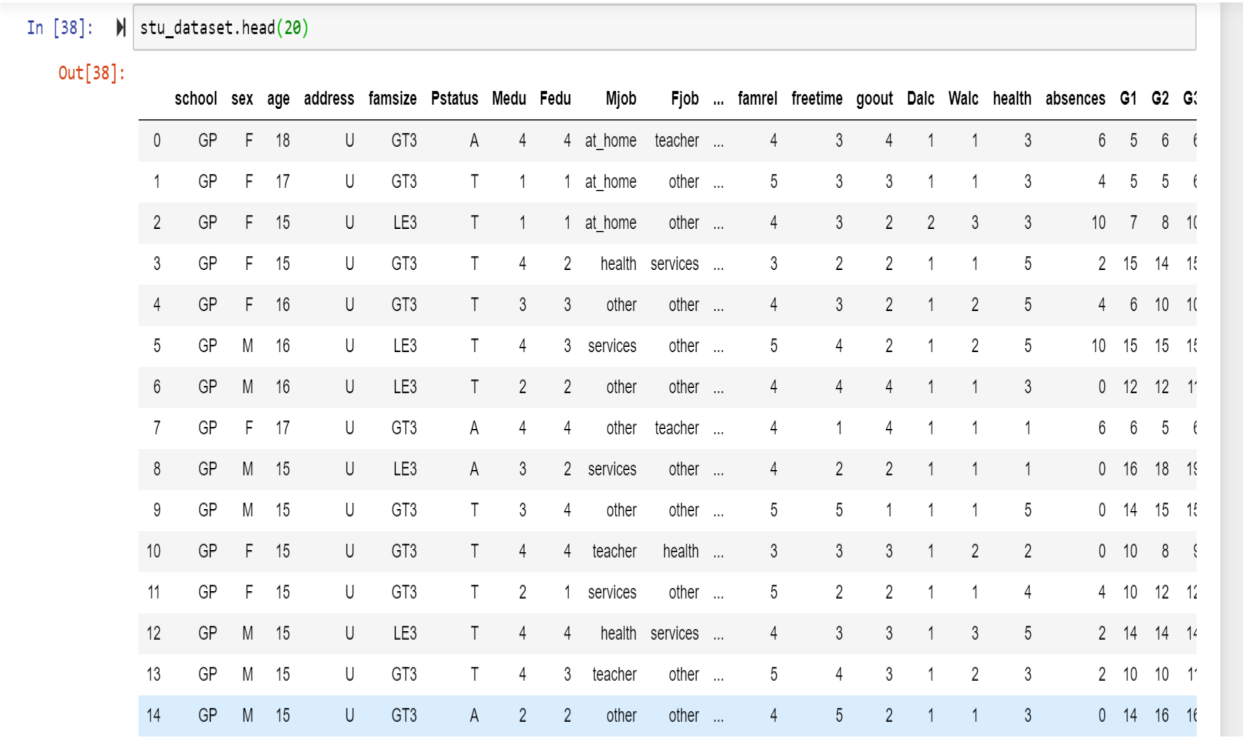
File Size: 57KB

Table 1 : Description of the student attribute taken into build the dataset[1]

Attribute	Description (Domain)
sex	student’s sex (binary: female or male)
age	student’s age (numeric: from 15 to 22)
school	student’s school (binary: Gabriel Pereira or Mousinho da Silveira)
address	student’s home address type (binary: urban or rural)
Pstatus	parent’s cohabitation status (binary: living together or apart)
Medu	mother’s education (numeric: from 0 to 4a)
Mjob	mother’s job (nominalb)
Fedu	father’s education (numeric: from 0 to 4a)
Fjob	father’s job (nominalb) guardian student’s guardian (nominal: mother, father or other)
famsize	family size (binary: ≤ 3 or > 3)
famrel	quality of family relationships (numeric: from 1 – very bad to 5 – excellent)
reason	reason to choose this school (nominal: close to home, school reputation, course preference or other)
traveltime	home to school travel time (numeric: 1 – < 15 min., 2 – 15 to 30 min., 3 – 30 min. to 1 hour or 4 – > 1 hour).
studytime	weekly study time (numeric: 1 – < 2 hours, 2 – 2 to 5 hours, 3 – 5 to 10 hours or 4>10 hours)
failures	number of past class failures (numeric: n if 1 ≤ n < 3, else 4)
schoolsup	extra educational school support (binary: yes or no)
famsup	family educational support (binary: yes or no)
activities	extra-curricular activities (binary: yes or no)
paidclass	extra paid classes (binary: yes or no)
internet	Internet access at home (binary: yes or no)
nursery	attended nursery school (binary: yes or no)
higher	wants to take higher education (binary: yes or no)
romantic	with a romantic relationship (binary: yes or no)
freetime	free time after school (numeric: from 1 – very low to 5 – very high)
goout	going out with friends (numeric: from 1 – very low to 5 – very high)
Walc	weekend alcohol consumption (numeric: from 1 – very low to 5 – very high)
Dalc	workday alcohol consumption (numeric: from 1 – very low to 5 – very high)
health	current health status (numeric: from 1 – very bad to 5 – very good)
absences	number of school absences (numeric: from 0 to 93)
G1	first period grade (numeric: from 0 to 20)
G2	second period grade (numeric: from 0 to 20)
G3	final grade (numeric: from 0 to 20)

2. Dataset Exploration :-

We have done dataset exploration in IPython Notebook as well as the attribute information given above provided valuable information regarding the dataset. Some of the important discoveries are found as follows:



```
In [38]: stu_dataset.head(20)
Out[38]:
```

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	famrel	freetime	goout	Dalc	Walc	health	absences	G1	G2	G3
0	GP	F	18	U	GT3	A	4	4	at_home	teacher	...	4	3	4	1	1	3	6	5	6	6
1	GP	F	17	U	GT3	T	1	1	at_home	other	...	5	3	3	1	1	3	4	5	5	6
2	GP	F	15	U	LE3	T	1	1	at_home	other	...	4	3	2	2	3	3	10	7	8	10
3	GP	F	15	U	GT3	T	4	2	health	services	...	3	2	2	1	1	5	2	15	14	15
4	GP	F	16	U	GT3	T	3	3	other	other	...	4	3	2	1	2	5	4	6	10	10
5	GP	M	16	U	LE3	T	4	3	services	other	...	5	4	2	1	2	5	10	15	15	15
6	GP	M	16	U	LE3	T	2	2	other	other	...	4	4	4	1	1	3	0	12	12	11
7	GP	F	17	U	GT3	A	4	4	other	teacher	...	4	1	4	1	1	1	6	6	5	6
8	GP	M	15	U	LE3	A	3	2	services	other	...	4	2	2	1	1	1	0	16	18	19
9	GP	M	15	U	GT3	T	3	4	other	other	...	5	5	1	1	1	5	0	14	15	15
10	GP	F	15	U	GT3	T	4	4	teacher	health	...	3	3	3	1	2	2	0	10	8	9
11	GP	F	15	U	GT3	T	2	1	services	other	...	5	2	2	1	1	4	4	10	12	12
12	GP	M	15	U	LE3	T	4	4	health	services	...	4	3	3	1	3	5	2	14	14	14
13	GP	M	15	U	GT3	T	4	3	teacher	other	...	5	4	3	1	2	3	2	10	10	11
14	GP	M	15	U	GT3	A	2	2	other	other	...	4	5	2	1	1	3	0	14	16	16

Figure 2.1: Overview of Dataset

- There are 395 instances and 33 attributes in this dataset.
- All the 32 attributes except G3 are independent and support in the prediction of G3 which is output label.
- Grades G1, G2 & G3 are given on scale of 1-20.
- The dataset is mixture of numeric and nominal attribute.
- There are **no null values or missing values** for all the attributes.

Nishi Gupta
Gu1978

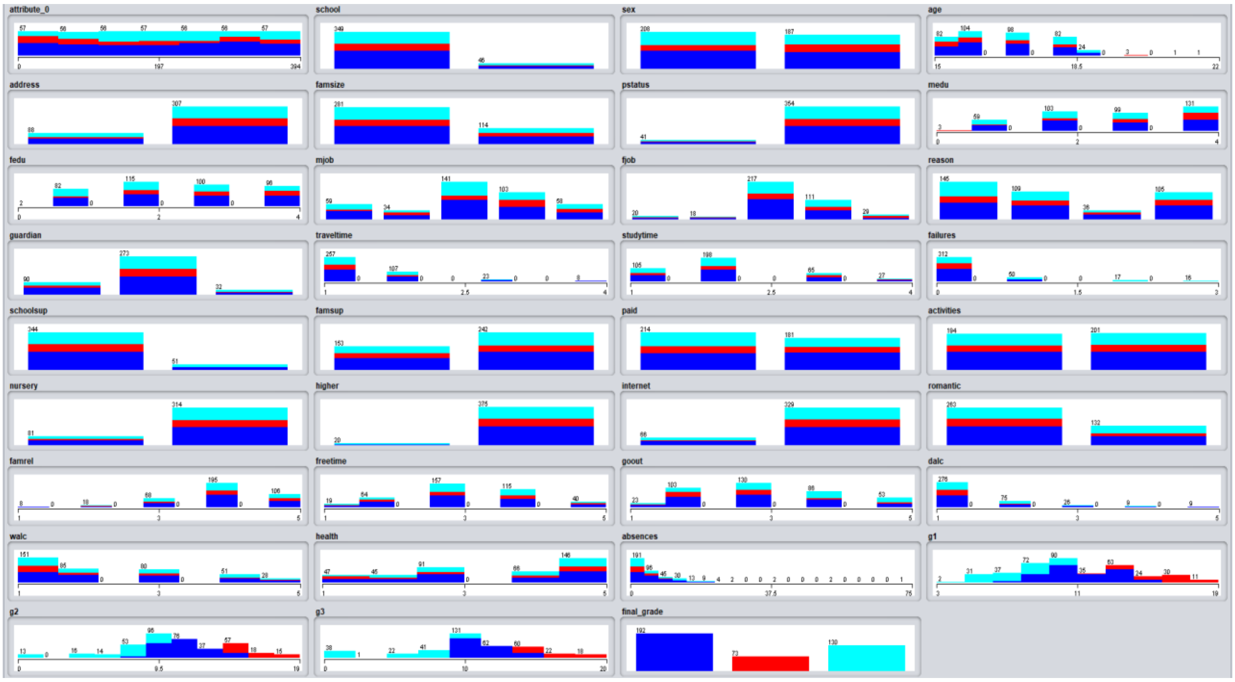


Figure 2.2: Histogram Plot for some of the features

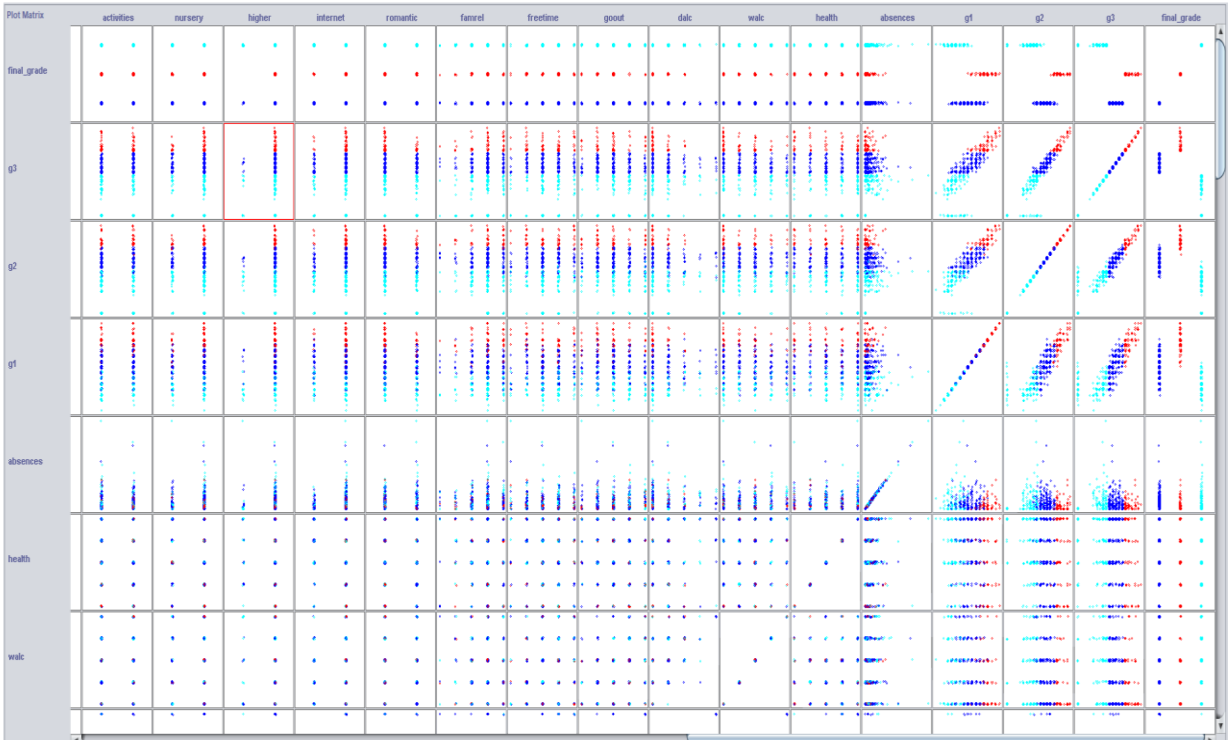
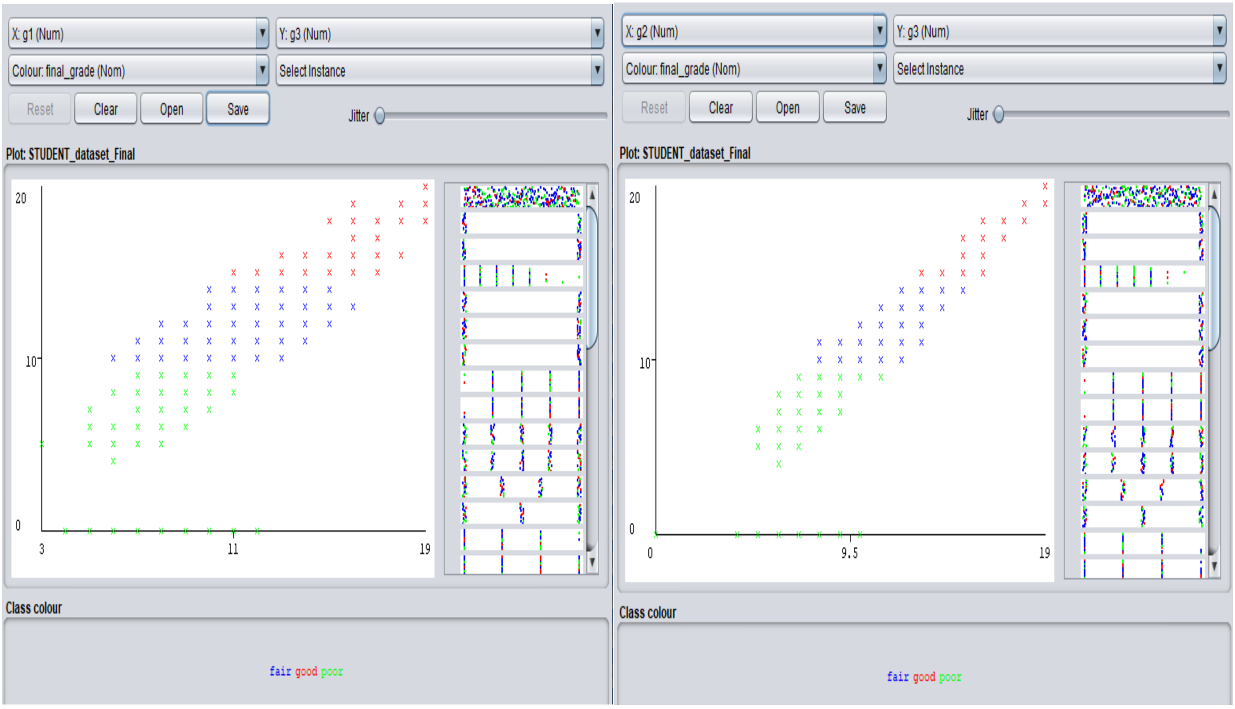


Figure 2.3: Plotting of some features



X: g1 (Num)
Y: g3 (Num)
Colour: final_grade (Nom)
Select Instance
Plot: STUDENT_dataset_Final

X: g2 (Num)
Y: g3 (Num)
Colour: final_grade (Nom)
Select Instance
Plot: STUDENT_dataset_Final

Figure 2.4: Snapshot of Correlation between G1 & G3, G2 & G3 respectively

3. Data Mining Software Tools:

- **IPython Notebook** – Python in IPython Notebook for data exploration and preprocessing.
- **WEKA** 3.8.2 - Waikato Environment for Knowledge Analysis (WEKA) tool for classification models.

DATA PRE-PROCESSING –

Data preprocessing is a data mining technique that involves transforming raw data that is often incomplete, inconsistent, lacking in certain trends, into an understandable format, so that we can further process the data to apply Data mining algorithms.

Steps Performed:-

- Performed pre-processing with *PYTHON in IPython notebook* by using numpy and pandas library. we have checked in each column for missing values. But in given dataset, there are no missing values available. So, there is no need of data cleaning in given dataset.
- Since, most of the columns are categorical attributes, if we apply variance filter, then it might remove some important features. So, we are not applying variance filter.

- Initially, the target output class (G3) ranges from 0 to 20 then mapped into Final_grades attribute class by applying categorical value-

Good	15-20
Fair	10-14
Poor	0-9

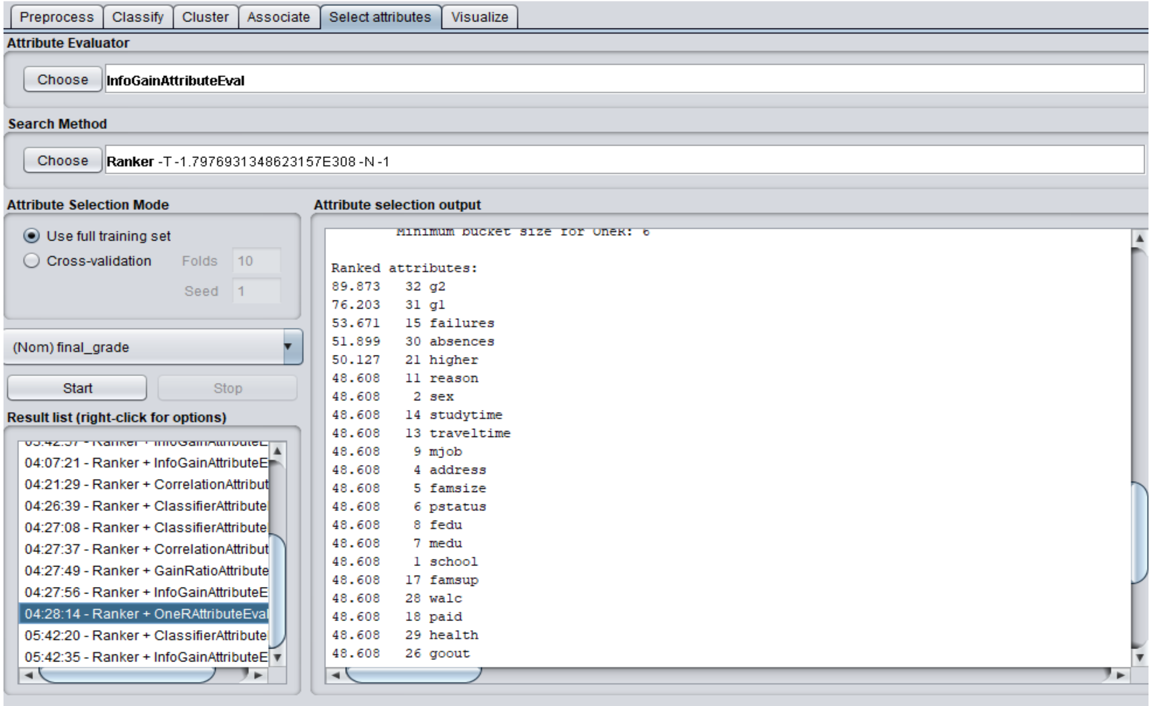
- Label Encoder method is applied on Final_grades attribute class which encode target labels; 0,1 & 2 are encoded label.
- Created data frame named dfd for classification; where G3 column is dropped for future prediction.
- As the downloaded data came in csv format. Thus, in order to use the dataset in *WEKA*, we converted the CSV file into ARFF.

ALGORITHM SELECTION-

As mentioned before, our goal is mainly focused on classification. we have created various data mining models to predict student final performance and compared model performance based on sample accuracy score.

After going through the paper[4] as mentioned that they have performed classification with five class and eight random attributes and their maximum accuracy was 76.73%. So, my goal was to increase the accuracy and thought to remove redundant variables.

We are selecting the best attribute on the basis of attribute's evaluator method namely CorrelationAttributeEval, GainRatioAttributeEval and InfoGainAttributeEval with Ranker Search Method in WEKA. After finding the result of the implementing attribute evaluator algorithm, we are selected 5 different attribute which mostly affect our prediction result.



Attribute Evaluator

InfoGainAttributeEval

Search Method

Ranker -T -1.7976931348623157E308 -N -1

Attribute Selection Mode

● Use full training set

○ Cross-validation Folds 10 Seed 1

(Nom) final_grade

Result list (right-click for options)

```
04:07:21 - Ranker + InfoGainAttributeE
04:21:29 - Ranker + CorrelationAttribut
04:26:39 - Ranker + ClassifierAttribute
04:27:08 - Ranker + ClassifierAttribute
04:27:37 - Ranker + CorrelationAttribut
04:27:49 - Ranker + GainRatioAttribute
04:27:56 - Ranker + InfoGainAttributeE
04:28:14 - Ranker + OneRAttributeEval
05:42:20 - Ranker + ClassifierAttribute
05:42:35 - Ranker + InfoGainAttributeE
```

Attribute selection output

```
        Minimum bucket size for OneR: 6

Ranked attributes:
89.873   32 g2
76.203   31 g1
53.671   15 failures
51.899   30 absences
50.127   21 higher
48.608   11 reason
48.608    2 sex
48.608   14 studytime
48.608   13 traveltime
48.608    9 mjob
48.608    4 address
48.608    5 famsize
48.608    6 pstatus
48.608    8 fedu
48.608    7 medu
48.608    1 school
48.608   17 famsup
48.608   28 walc
48.608   18 paid
48.608   29 health
48.608   26 goout
```

We have implemented total eight models out of which we have selected only top five model on the basis of accuracy. The lowest accuracy models are lazy IBK, Kstar, ZeroR having accuracy in between 45-70% accuracy. Here are the selected model for our dataset-

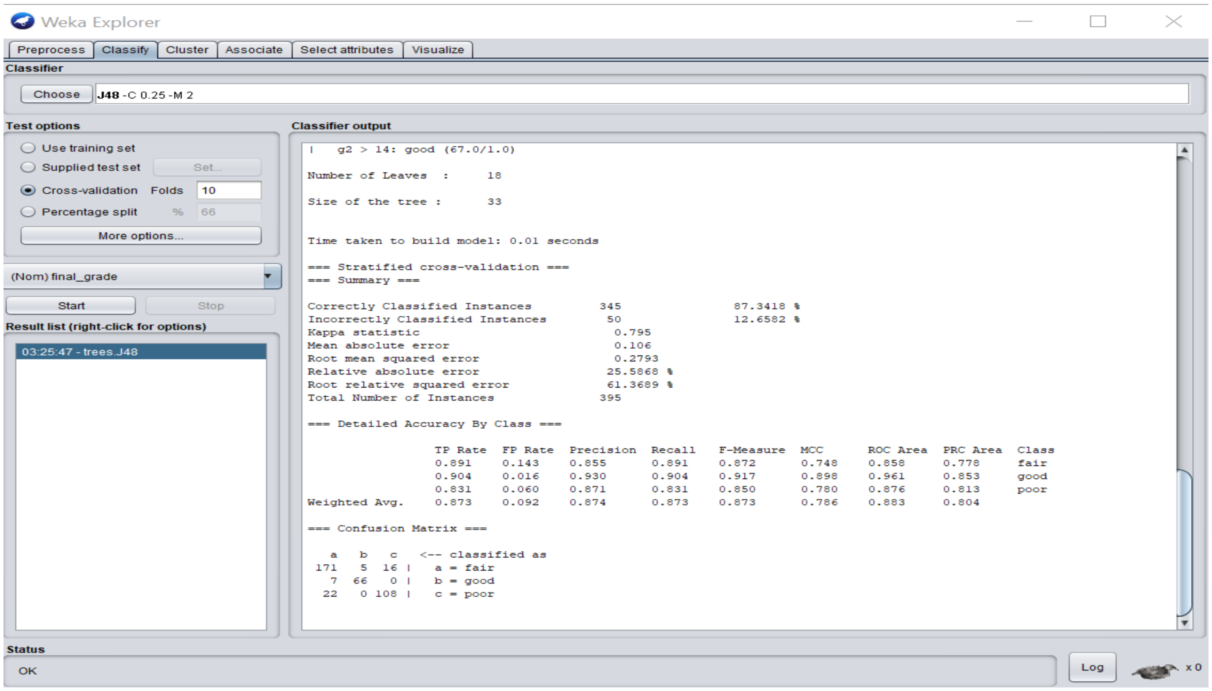
I. Decision Tree (J48) classifier
II. Naïve Bayes Classifier
III. Random Forest Classifier
IV. JRip Classifier
V. Multilayer Perceptron

NOTE :- For implementation of all the classification models, we have tested on 10-fold cross validation.

1. DECISION TREE (J48):

- Decision Tree is one of the most popular Data Mining algorithm. Decision Tree is used for both classification and regression problems. Here, we used it as a classification problem
- J48 showed accuracy 87.3418 % when experimented with all 32 attributes.

- When we experimented with randomly 5 selected attributes and set minNumObj to six and numFolds to five, then the accuracy is increased to 89.8734 %, which is in increment of 2.5316%.



Classifier: J48 -C 0.25 -M 2

Test options: ○ Use training set ○ Supplied test set ● Cross-validation Folds 10 ○ Percentage split % 66

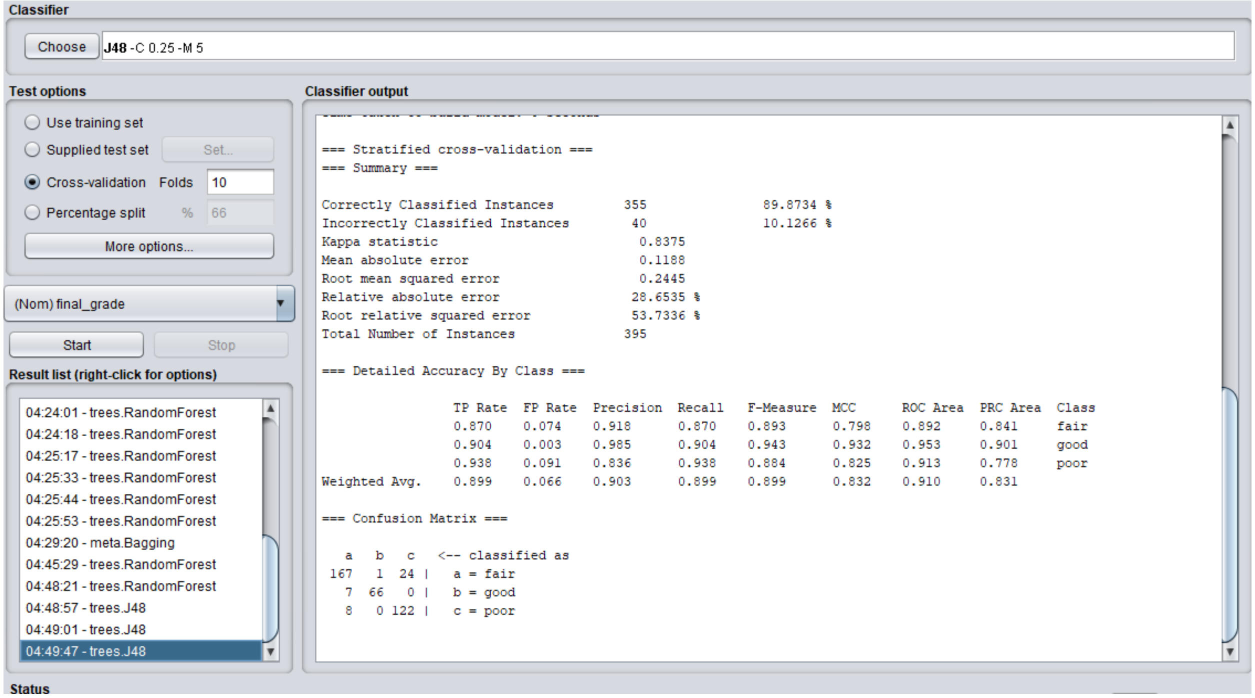
(Nom) final_grade

Result list (right-click for options): 03:25:47 - trees.J48

Status: OK

```
|   g2 > 14: good (67.0/1.0)

Number of Leaves  :     18

Size of the tree :      33


Time taken to build model: 0.01 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         345               87.3418 %
Incorrectly Classified Instances        50               12.6582 %
Kappa statistic                          0.795
Mean absolute error                      0.106
Root mean squared error                  0.2793
Relative absolute error                 25.5868 %
Root relative squared error             61.3689 %
Total Number of Instances              395

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.891    0.143    0.855      0.891    0.872      0.748    0.858     0.778     fair
                 0.904    0.016    0.930      0.904    0.917      0.898    0.961     0.853     good
                 0.831    0.060    0.871      0.831    0.850      0.780    0.876     0.813     poor
Weighted Avg.    0.873    0.092    0.874      0.873    0.873      0.786    0.883     0.804

=== Confusion Matrix ===

   a   b   c   <-- classified as
 171   5  16 |   a = fair
   7  66   0 |   b = good
  22   0 108 |   c = poor
```

Figure1.1: J48 with 32 attributes



Classifier: J48 -C 0.25 -M 5

Test options: ○ Use training set ○ Supplied test set ● Cross-validation Folds 10 ○ Percentage split % 66

(Nom) final_grade

Result list (right-click for options):
04:24:01 - trees.RandomForest
04:24:18 - trees.RandomForest
04:25:17 - trees.RandomForest
04:25:33 - trees.RandomForest
04:25:44 - trees.RandomForest
04:25:53 - trees.RandomForest
04:29:20 - meta.Bagging
04:45:29 - trees.RandomForest
04:48:21 - trees.RandomForest
04:48:57 - trees.J48
04:49:01 - trees.J48
04:49:47 - trees.J48

Status

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         355               89.8734 %
Incorrectly Classified Instances        40               10.1266 %
Kappa statistic                          0.8375
Mean absolute error                      0.1188
Root mean squared error                  0.2445
Relative absolute error                 28.6535 %
Root relative squared error             53.7336 %
Total Number of Instances              395

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.870    0.074    0.918      0.870    0.893      0.798    0.892     0.841     fair
                 0.904    0.003    0.985      0.904    0.943      0.932    0.953     0.901     good
                 0.938    0.091    0.836      0.938    0.884      0.825    0.913     0.778     poor
Weighted Avg.    0.899    0.066    0.903      0.899    0.899      0.832    0.910     0.831

=== Confusion Matrix ===

   a   b   c   <-- classified as
 167   1  24 |   a = fair
   7  66   0 |   b = good
   8   0 122 |   c = poor
```

Figure 1.2: J48 with random 5 attributes

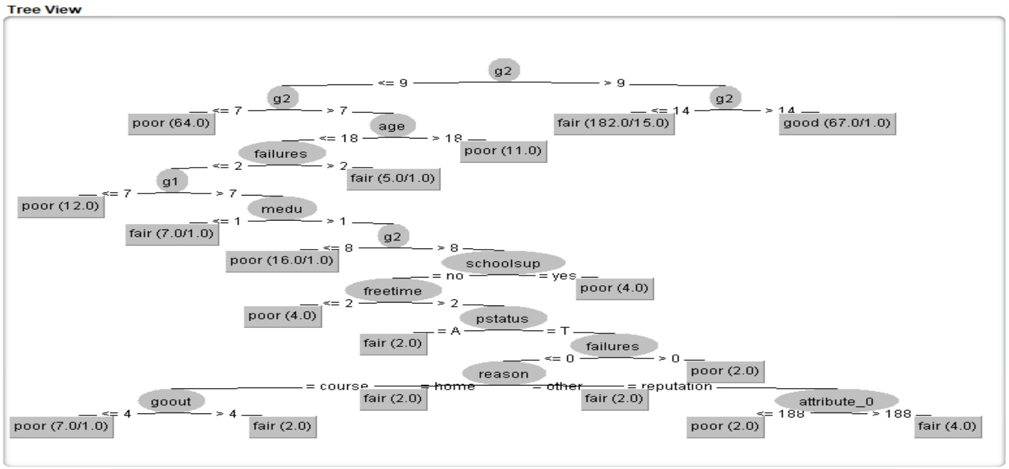
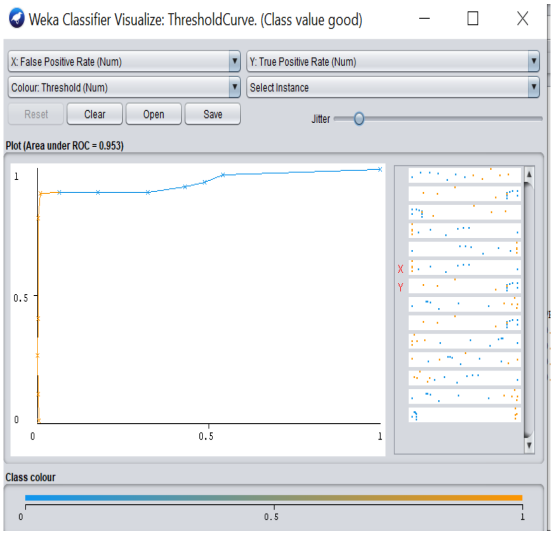


Figure 1.3: Tree


Weka Classifier Visualize: ThresholdCurve. (Class value good)

X: False Positive Rate (Num)

Y: True Positive Rate (Num)

Colour: Threshold (Num)

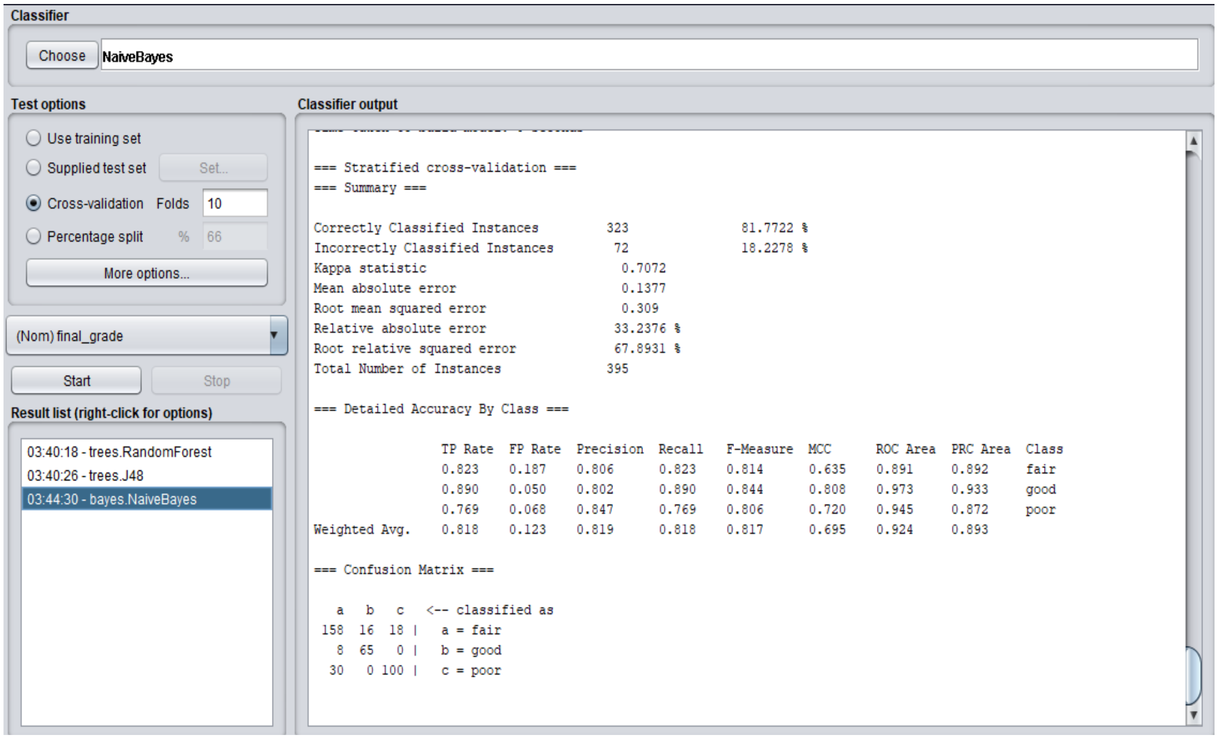
Plot (Area under ROC = 0.953)

Figure 1.4: ROC curve for J48(class value good)

2. NAIVEBAYES CLASSIFICATION :

- Naïve Bayes classifier is based on Bayes theorem to classify objects.
- When we experimented with all 32 attributes, Naïve Bayes showed the accuracy of 81.77%.
- With the randomly selected five attributes, accuracy increased to 83.038%, this shows accuracy of Naive Bayes is increased to 1.268%.
- For better performance of this model, we set kernelEstimator to True and then model performance was quite good which is 85.3165%.

Nishi Gupta
Gu1978



Classifier: NaiveBayes

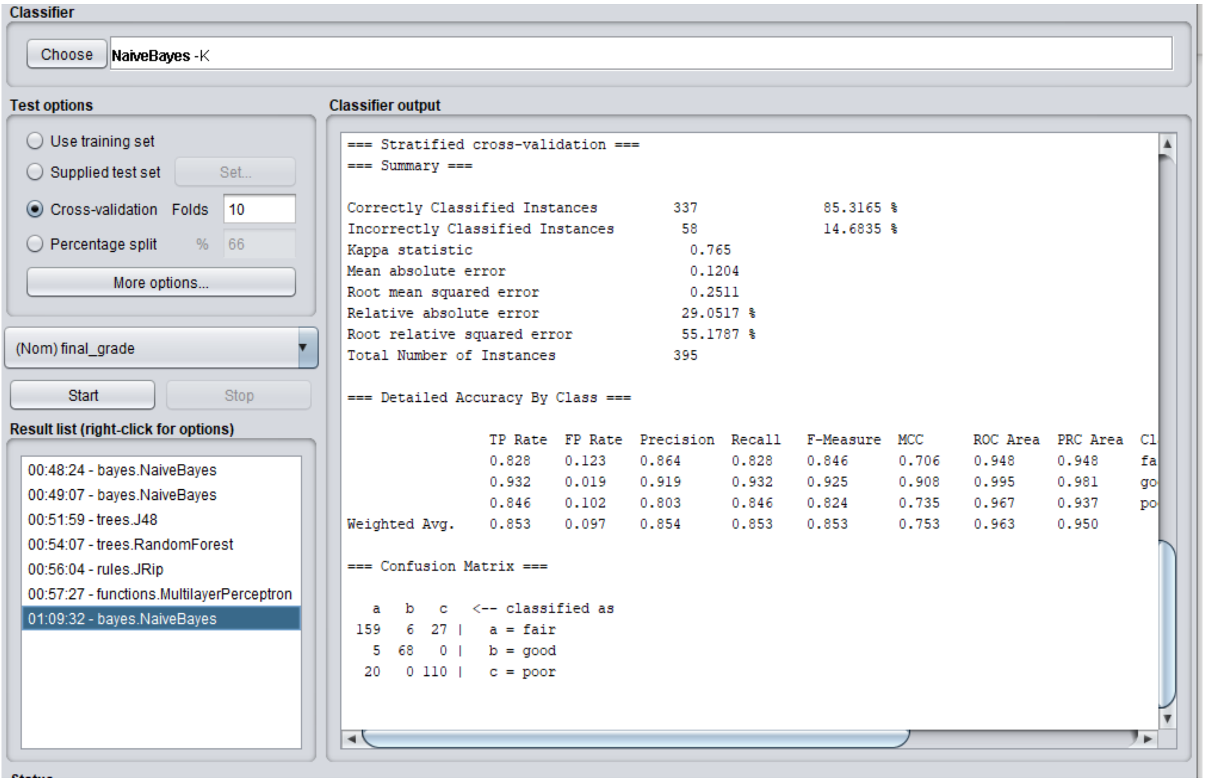
Test options:
- ○ Use training set
- ○ Supplied test set
- ● Cross-validation Folds 10
- ○ Percentage split % 66

(Nom) final_grade

Result list (right-click for options)
- 03:40:18 - trees.RandomForest
- 03:40:26 - trees.J48
- 03:44:30 - bayes.NaiveBayes

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         323               81.7722 %
Incorrectly Classified Instances        72               18.2278 %
Kappa statistic                          0.7072
Mean absolute error                      0.1377
Root mean squared error                  0.309
Relative absolute error                 33.2376 %
Root relative squared error             67.8931 %
Total Number of Instances              395

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.823    0.187    0.806      0.823    0.814      0.635    0.891     0.892     fair
                 0.890    0.050    0.802      0.890    0.844      0.808    0.973     0.933     good
                 0.769    0.068    0.847      0.769    0.806      0.720    0.945     0.872     poor
Weighted Avg.    0.818    0.123    0.819      0.818    0.817      0.695    0.924     0.893

=== Confusion Matrix ===

   a   b   c   <-- classified as
 158  16  18 |   a = fair
   8  65   0 |   b = good
  30   0 100 |   c = poor
```

Figure 2.1: NaiveBayes with 32 attributes



Classifier: NaiveBayes -K

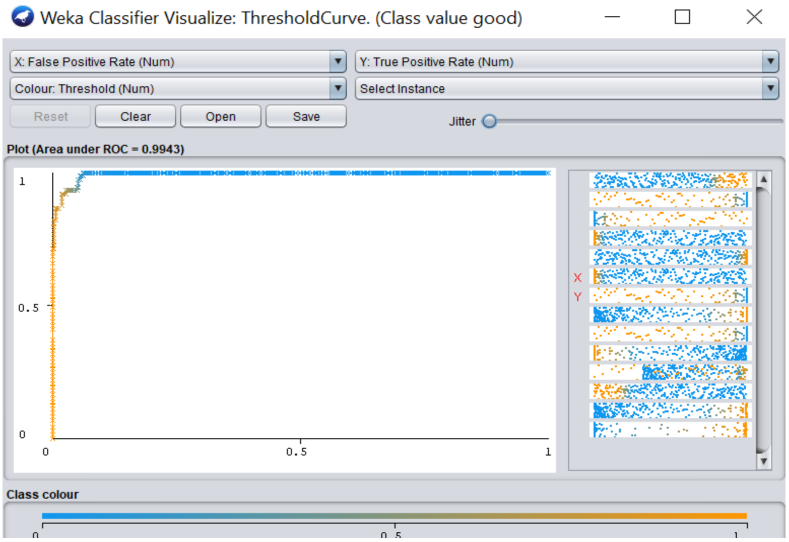
Test options:
- ○ Use training set
- ○ Supplied test set
- ● Cross-validation Folds 10
- ○ Percentage split % 66

(Nom) final_grade

Result list (right-click for options)
- 00:48:24 - bayes.NaiveBayes
- 00:49:07 - bayes.NaiveBayes
- 00:51:59 - trees.J48
- 00:54:07 - trees.RandomForest
- 00:56:04 - rules.JRip
- 00:57:27 - functions.MultilayerPerceptron
- 01:09:32 - bayes.NaiveBayes

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         337               85.3165 %
Incorrectly Classified Instances        58               14.6835 %
Kappa statistic                          0.765
Mean absolute error                      0.1204
Root mean squared error                  0.2511
Relative absolute error                 29.0517 %
Root relative squared error             55.1787 %
Total Number of Instances              395

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Cl
                 0.828    0.123    0.864      0.828    0.846      0.706    0.948     0.948     fa
                 0.932    0.019    0.919      0.932    0.925      0.908    0.995     0.981     go
                 0.846    0.102    0.803      0.846    0.824      0.735    0.967     0.937     po
Weighted Avg.    0.853    0.097    0.854      0.853    0.853      0.753    0.963     0.950

=== Confusion Matrix ===

   a   b   c   <-- classified as
 159   6  27 |   a = fair
   5  68   0 |   b = good
  20   0 110 |   c = poor
```

Figure 2.2: NaiveBayes with 5 random attributes and KernelEstimator set TRUE



Weka Classifier Visualize: ThresholdCurve. (Class value good)

X: False Positive Rate (Num)

Y: True Positive Rate (Num)

Colour: Threshold (Num)

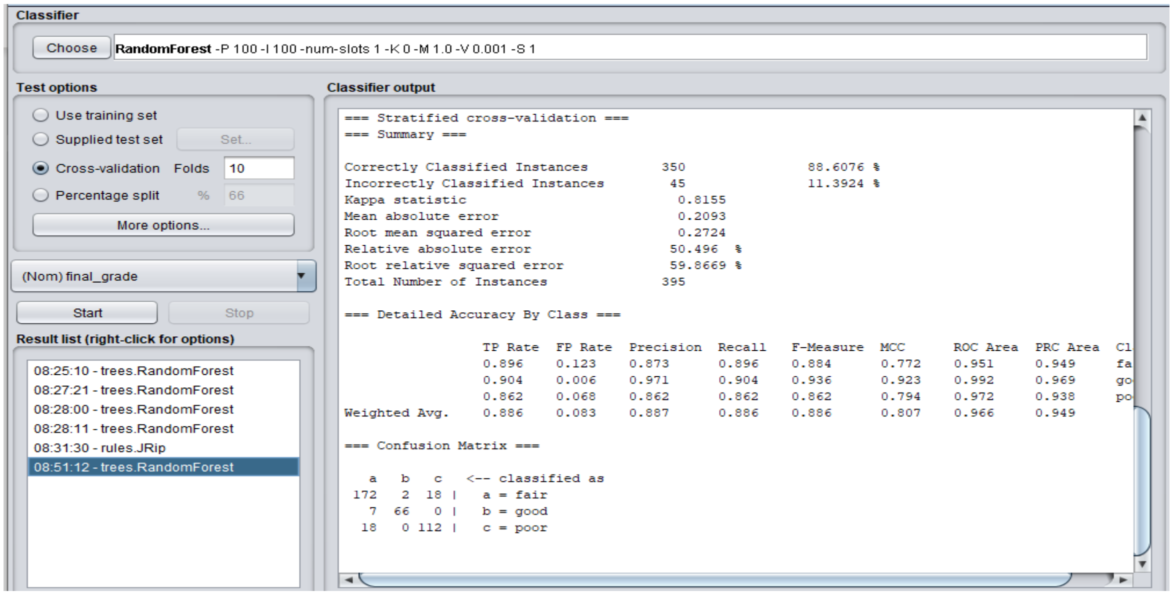
Select Instance

Plot (Area under ROC = 0.9943)

Figure 2.3: ROC curve for Naïve Bayes(class value good)

3. RANDOM FOREST :

- Random Forest algorithm is used as a supervised learning and known for its powerfulness. In addition to powerful decision tree representation, it is capable of generalizing well.
- It showed accuracy with all 32 attributes is 88.6076 %.
- When we randomly select 5 attributes and change the maxDepth to 3, accuracy become 89.8734 % which is slight increase in the accuracy i.e 1.2658%.



Classifier

RandomForest -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1

Test options

○ Use training set
○ Supplied test set
● Cross-validation Folds 10
○ Percentage split % 66

(Nom) final_grade

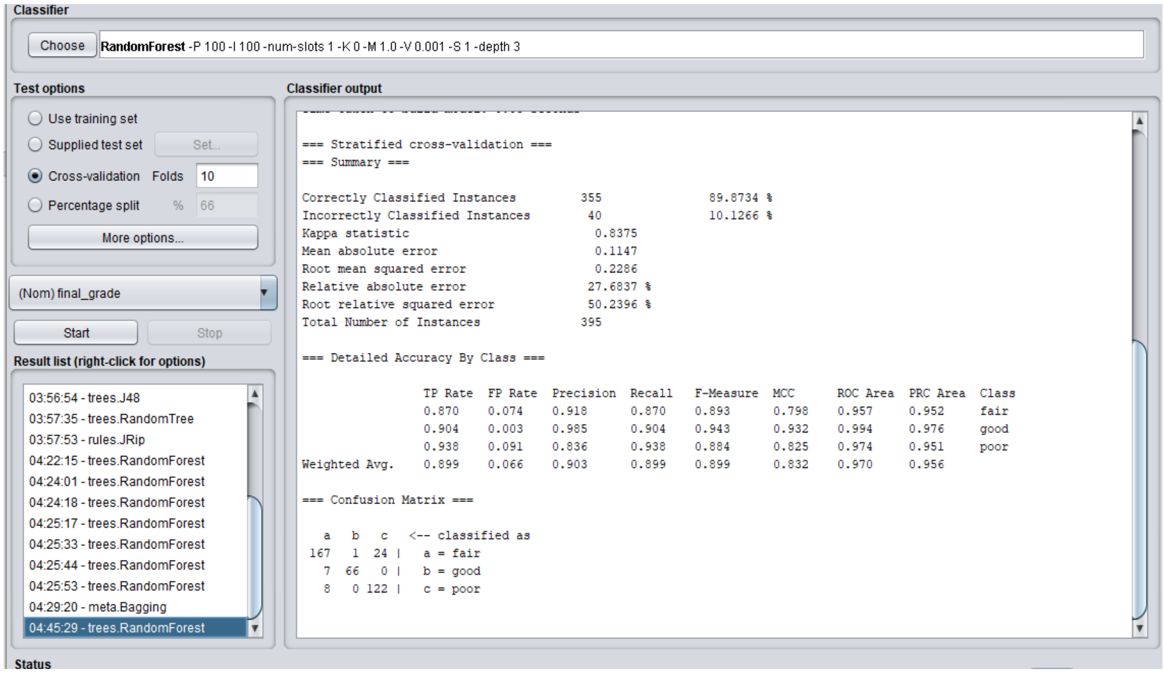
Result list (right-click for options)

08:25:10 - trees.RandomForest
08:27:21 - trees.RandomForest
08:28:00 - trees.RandomForest
08:28:11 - trees.RandomForest
08:31:30 - rules.JRip
08:51:12 - trees.RandomForest

Classifier output

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         350               88.6076 %
Incorrectly Classified Instances        45               11.3924 %
Kappa statistic                          0.8155
Mean absolute error                      0.2093
Root mean squared error                  0.2724
Relative absolute error                 50.496  %
Root relative squared error             59.8669 %
Total Number of Instances              395

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Cl
                 0.896    0.123    0.873      0.896    0.884      0.772    0.951     0.949     fa
                 0.904    0.006    0.971      0.904    0.936      0.923    0.992     0.969     go
                 0.862    0.068    0.862      0.862    0.862      0.794    0.972     0.938     po
Weighted Avg.    0.886    0.083    0.887      0.886    0.886      0.807    0.966     0.949

=== Confusion Matrix ===

   a   b   c   <-- classified as
 172   2  18 |   a = fair
   7  66   0 |   b = good
  18   0 112 |   c = poor
```

Figure 3.1: Random Forest with 32 attributes



Classifier

Choose **RandomForest** -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1 -depth 3

Test options

○ Use training set
○ Supplied test set Set...
● Cross-validation Folds 10
○ Percentage split % 66
More options...

(Nom) final_grade

Start Stop

Result list (right-click for options)

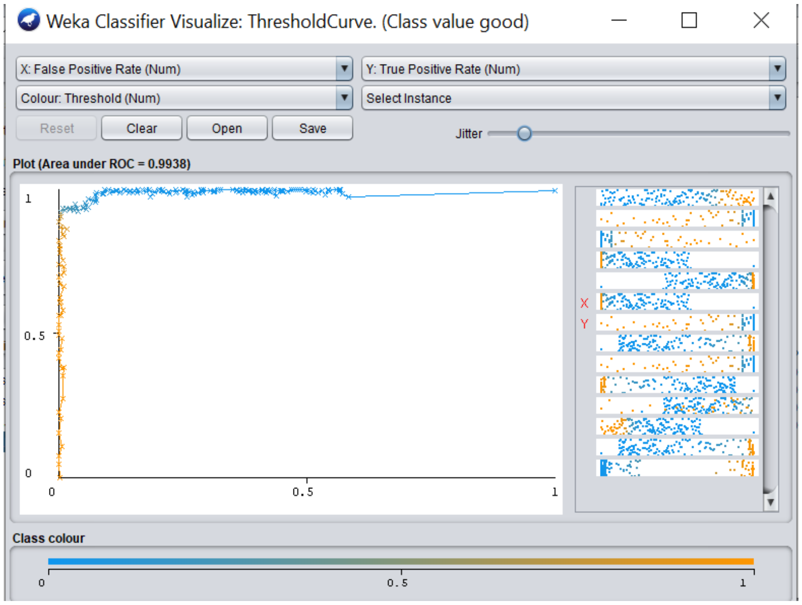
03:56:54 - trees.J48
03:57:35 - trees.RandomTree
03:57:53 - rules.JRip
04:22:15 - trees.RandomForest
04:24:01 - trees.RandomForest
04:24:18 - trees.RandomForest
04:25:17 - trees.RandomForest
04:25:33 - trees.RandomForest
04:25:44 - trees.RandomForest
04:25:53 - trees.RandomForest
04:29:20 - meta.Bagging
04:45:29 - trees.RandomForest

Classifier output

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         355               89.8734 %
Incorrectly Classified Instances        40               10.1266 %
Kappa statistic                          0.8375
Mean absolute error                      0.1147
Root mean squared error                  0.2286
Relative absolute error                 27.6837 %
Root relative squared error             50.2396 %
Total Number of Instances              395

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.870    0.074    0.918      0.870    0.893      0.798    0.957     0.952     fair
                 0.904    0.003    0.985      0.904    0.943      0.932    0.994     0.976     good
                 0.938    0.091    0.836      0.938    0.884      0.825    0.974     0.951     poor
Weighted Avg.    0.899    0.066    0.903      0.899    0.899      0.832    0.970     0.956

=== Confusion Matrix ===

   a   b   c   <-- classified as
 167   1  24 |   a = fair
   7  66   0 |   b = good
   8   0 122 |   c = poor
```

Status

Figure 3.2: Random Forest with 5 random attributes



Weka Classifier Visualize: ThresholdCurve. (Class value good)

X: False Positive Rate (Num)
Y: True Positive Rate (Num)
Colour: Threshold (Num)
Select Instance

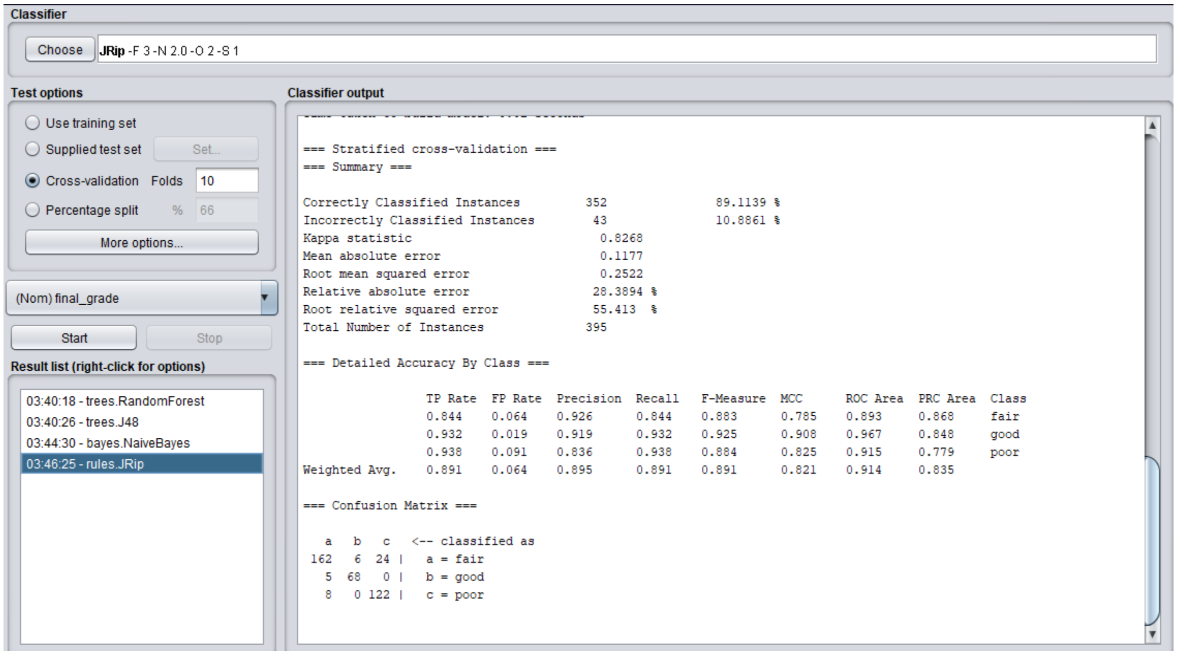
Plot (Area under ROC = 0.9938)

Figure 3.3: ROC curve for RandomForest(class value good)

4. JRIP CLASSIFICATION ALGORITHM:

- JRip is based on association rules with reduced error pruning (REP), a very common and effective technique found in decision tree algorithms.
- When we experimented with all 32 attributes, JRip Classification Algorithm showed 89.1139% accuracy.

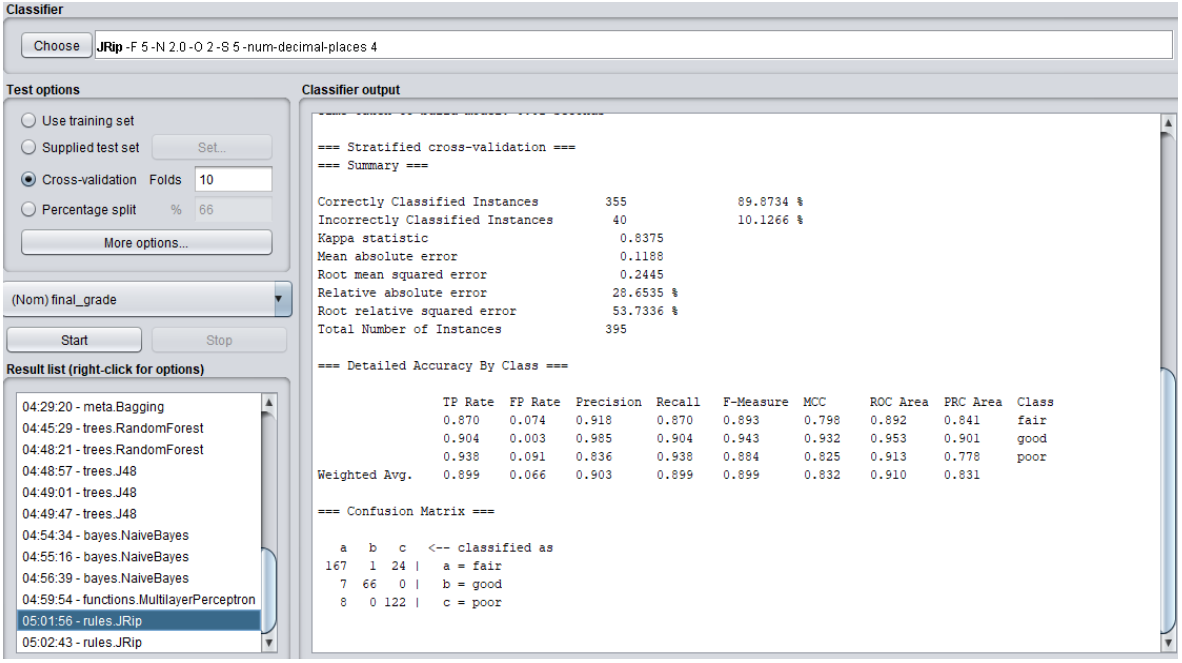
- When randomly five attributes selected and set folds to 5, seed value is set to 5, numDecimalPlaces value is set to 4, its accuracy increased to 89.8734% which is slight increase in the accuracy i.e. 0.7595% improve the prediction accuracy.



JRip -F 3 -N 2.0 -O 2 -S 1

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         352               89.1139 %
Incorrectly Classified Instances        43               10.8861 %
Kappa statistic                          0.8268
Mean absolute error                      0.1177
Root mean squared error                  0.2522
Relative absolute error                 28.3894 %
Root relative squared error             55.413  %
Total Number of Instances              395

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.844    0.064    0.926      0.844    0.883      0.785    0.893     0.868     fair
                 0.932    0.019    0.919      0.932    0.925      0.908    0.967     0.848     good
                 0.938    0.091    0.836      0.938    0.884      0.825    0.915     0.779     poor
Weighted Avg.    0.891    0.064    0.895      0.891    0.891      0.821    0.914     0.835

=== Confusion Matrix ===

   a   b   c   <-- classified as
 162   6  24 |   a = fair
   5  68   0 |   b = good
   8   0 122 |   c = poor
```

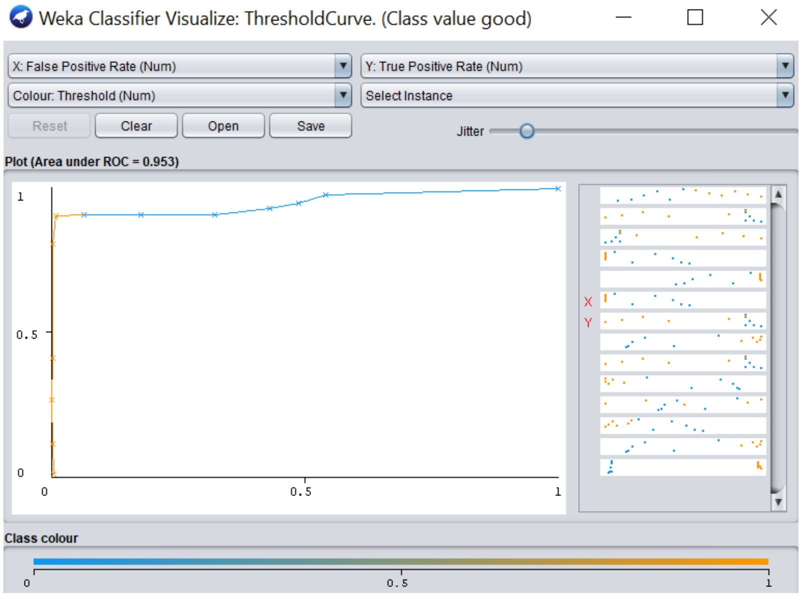
Figure4.1: JRip with 32 attributes



JRip -F 5 -N 2.0 -O 2 -S 5 -num-decimal-places 4

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         355               89.8734 %
Incorrectly Classified Instances        40               10.1266 %
Kappa statistic                          0.8375
Mean absolute error                      0.1188
Root mean squared error                  0.2445
Relative absolute error                 28.6535 %
Root relative squared error             53.7336 %
Total Number of Instances              395

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.870    0.074    0.918      0.870    0.893      0.798    0.892     0.841     fair
                 0.904    0.003    0.985      0.904    0.943      0.932    0.953     0.901     good
                 0.938    0.091    0.836      0.938    0.884      0.825    0.913     0.778     poor
Weighted Avg.    0.899    0.066    0.903      0.899    0.899      0.832    0.910     0.831

=== Confusion Matrix ===

   a   b   c   <-- classified as
 167   1  24 |   a = fair
   7  66   0 |   b = good
   8   0 122 |   c = poor
```

Figure 4.2: JRip with random 5 attributes



X: False Positive Rate (Num)

Y: True Positive Rate (Num)

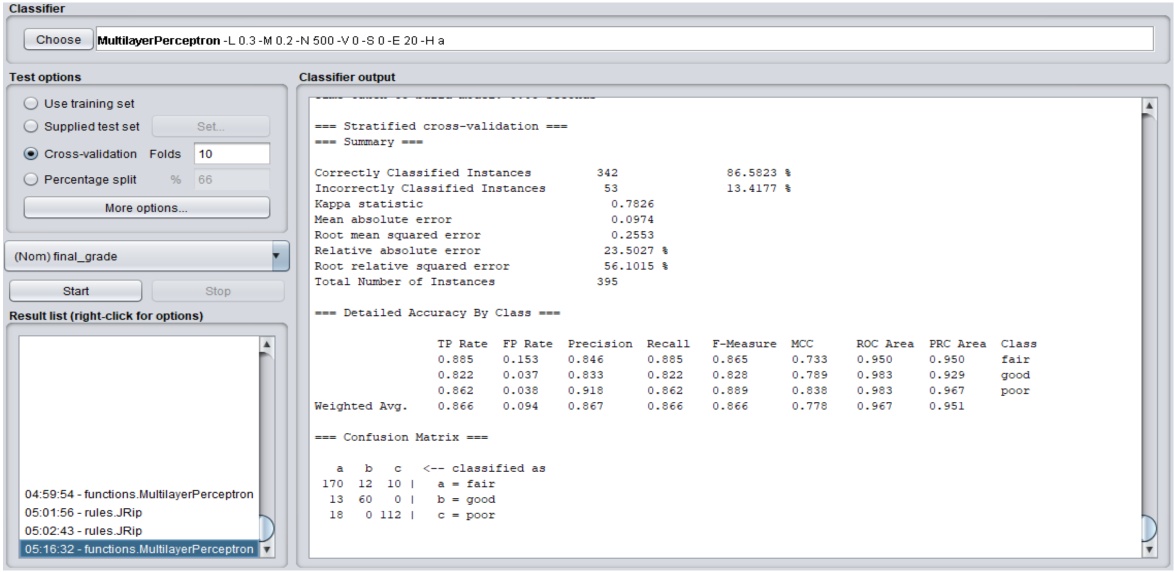
Colour: Threshold (Num)

Plot (Area under ROC = 0.953)

Figure 4.3: ROC curve for JRip(class value good)

5. MULTILAYER PERCEPTRON :

- Multilayer Perceptron is a feed forward artificial neural network model trained with the standard back propagation algorithm that maps sets of input data onto a collection of acceptable output.[3]
- It showed accuracy with all 32 attributes is 86.5823%.
- When randomly five attributes were selected it showed the increased accuracy of 88.1013 %.



MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a

○ Use training set
○ Supplied test set
● Cross-validation Folds 10
○ Percentage split % 66

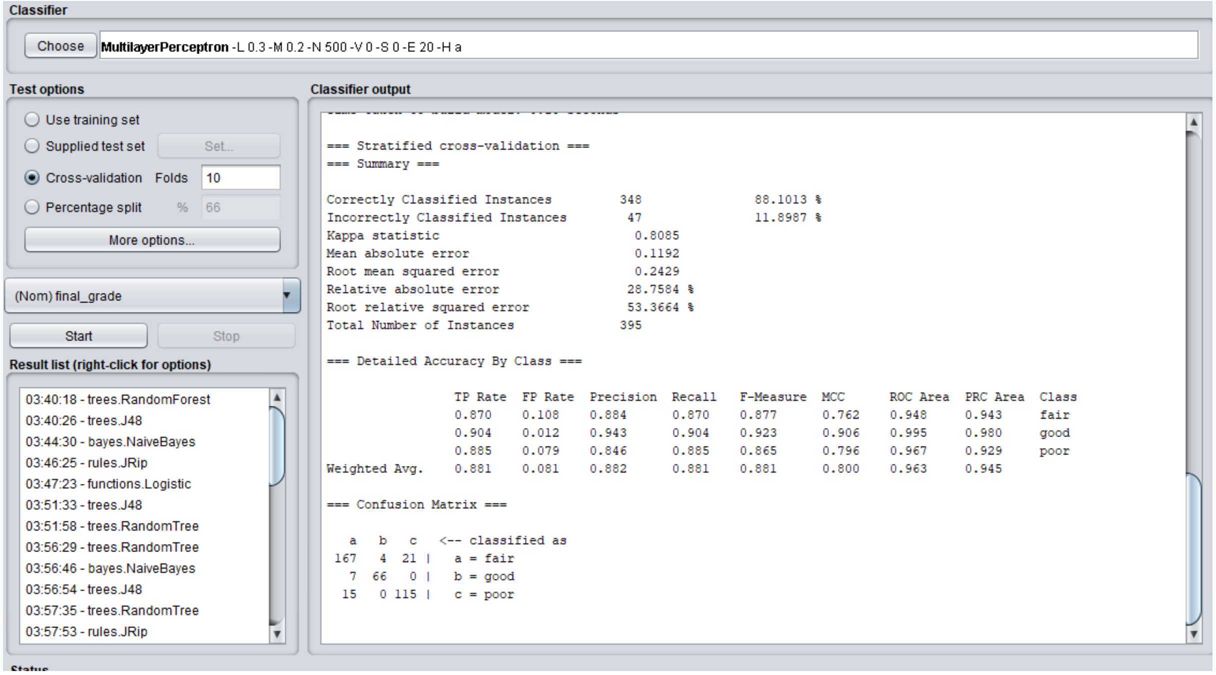
(Nom) final_grade

04:59:54 - functions.MultilayerPerceptron
05:01:56 - rules.JRip
05:02:43 - rules.JRip
05:16:32 - functions.MultilayerPerceptron

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         342               86.5823 %
Incorrectly Classified Instances        53               13.4177 %
Kappa statistic                          0.7826
Mean absolute error                      0.0974
Root mean squared error                  0.2553
Relative absolute error                 23.5027 %
Root relative squared error             56.1015 %
Total Number of Instances              395

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.885    0.153    0.846      0.885    0.865      0.733    0.950     0.950     fair
                 0.822    0.037    0.833      0.822    0.828      0.789    0.983     0.929     good
                 0.862    0.038    0.918      0.862    0.889      0.838    0.983     0.967     poor
Weighted Avg.    0.866    0.094    0.867      0.866    0.866      0.778    0.967     0.951

=== Confusion Matrix ===

   a   b   c   <-- classified as
 170  12  10 |   a = fair
  13  60   0 |   b = good
  18   0 112 |   c = poor
```

Figure 5.1: Multilayer Perceptron with 32 attributes

Nishi Gupta
Gu1978



Classifier: MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a

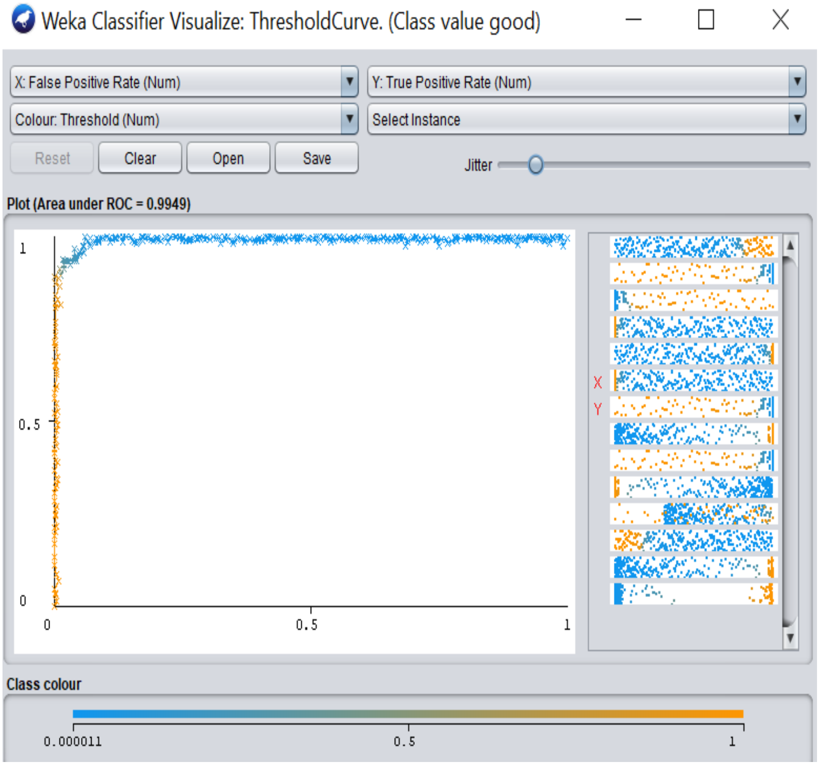
Test options: ● Cross-validation Folds 10 ○ Percentage split % 66

(Nom) final_grade

```
03:40:18 - trees.RandomForest
03:40:26 - trees.J48
03:44:30 - bayes.NaiveBayes
03:46:25 - rules.JRip
03:47:23 - functions.Logistic
03:51:33 - trees.J48
03:51:58 - trees.RandomTree
03:56:29 - trees.RandomTree
03:56:46 - bayes.NaiveBayes
03:56:54 - trees.J48
03:57:35 - trees.RandomTree
03:57:53 - rules.JRip
```

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         348               88.1013 %
Incorrectly Classified Instances        47               11.8987 %
Kappa statistic                          0.8085
Mean absolute error                      0.1192
Root mean squared error                  0.2429
Relative absolute error                 28.7584 %
Root relative squared error             53.3664 %
Total Number of Instances              395

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.870    0.108    0.884      0.870    0.877      0.762    0.948     0.943     fair
                 0.904    0.012    0.943      0.904    0.923      0.906    0.995     0.980     good
                 0.885    0.079    0.846      0.885    0.865      0.796    0.967     0.929     poor
Weighted Avg.    0.881    0.081    0.882      0.881    0.881      0.800    0.963     0.945

=== Confusion Matrix ===

   a   b   c   <-- classified as
 167   4  21 |   a = fair
   7  66   0 |   b = good
  15   0 115 |   c = poor
```

Figure 5.2: Multilayer Perceptron with 5 random attributes



Plot (Area under ROC = 0.9949)

Figure 5.3: ROC curve for Multilayer Perceptron(class value good)

ANALYSIS RESULTS AND COMPARISON:

After implementing all five models, I get different measures like accuracy, precision, recall, F1, TP rate, FP rate, Error rate, etc. and compared all model on the basis of these measures.

Measures	Decision Tree (J48)	Naïve Bayes	Random Forest	JRip Classifier	Multilayer Perceptron
Accuracy	89.8734 %,	85.3165%.	89.8734%	89.8734%	88.101%
TP Rate	0.899	0.830	0.899	0.899	0.881
FP Rate	0.066	0.112	0.066	0.066	0.081
Precision	0.903	0.830	0.903	0.903	0.882
Recall	0.899	0.830	0.899	0.899	0.881
F-measure	0.899	0.830	0.899	0.899	0.881
ROC Curve	0.910	0.954	0.910	0.910	0.963
Error Rate	10.126%	16.962%	10.126%	10.126%	11.898%

By comparing, we got the three models having same highest accuracy with 89.87%, which is J48, Random Forest and JRip for predicting student result and they perform equally well among all five classification models.

All models in the descending order of their accuracy :-

- J48 : 89.87%
- Random Forest : 89.87%
- JRip : 89.87%
- Multilayer Perceptron : 88.10%
- Naïve Bayes : 85.31%

CONCLUSION:

After implementing 5 different classification algorithms on our dataset. The obtained result shows that we can achieve high accuracy, provided that the first and second period grades are known. From the decision tree classification figure1.3, it is clear that G2 is the most significant factor as it appear on the top of decision tree for the prediction of G3 (final grades).

Overall, from our implementation of the models we can conclude that J48, Random Forest and JRip are performed equally well with more than 89% accuracy and surpass other classification algorithm.

CHALLENGES

1. After dataset preprocessing, the difficulty which faced is in converting CSV file to ARFF in WEKA.
2. Choosing perfect classification for a data mining problem like this is always a challenge as we have to perform lot of trials in selection of correct attribute which would lead to highest accuracy.
3. While using WEKA, we have chosen cross-validation for 10 folds. Having 10 folds means 90% of full data is used for training (10% for testing). However, the current model also shows good accuracy result. But if we can proceed implementation in R or python then we can check overfitting of data.

FUTURE ANALYSIS

To develop the same model using conventional programing language so that we can understand the data and handle the data manually. However, the current model also shows the good accuracy but if we can check implementation in R or python then we can check data is overfitted.

REFERENCE

I read the following papers and then implement my project. So, here are the reference:

1. P. Cortez and A. Silva. Using Data Mining to Predict Secondary School Student Performance. In A. Brito and J. Teixeira Eds., Proceedings of 5th FUture BUsiness TEChnology Conference (FUBUTEC 2008) pp. 5-12, Porto, Portugal, April, 2008, EUROSIS, ISBN.
2. Comparative Analysis of Models for Student Performance with Data Mining Tools A. K. Shrivas[1], Pragya Tiwari[2]
3. Prediction and Analysis of Student Performance by Data Mining in WEKA. www.rcciit.org/students_projects/projects/it/2018/GR4.pdf
4. Educational Data Mining: Student Performance Prediction in Academic Y. K. Salal, S. M. Abdullaev, Mukesh Kumar