Assignment - 6

[Association Rule Mining and Application]

Rahul Midha
B00766975

Niravsinh Jadeja
B00789139

1. Application Scenario

This application helps in deciding whether the students will go on for higher education or not, which is based on various factors such as Parent's cohabitation status, Mother's job and Father's Job etc. The application saves us from doing a lot of manual calculations. We used WEKA[1] and R[3] to select out dataset and to generate the rules. The Dataset has 13 columns which are as follows:-

1. school- It has 2 fields as school names(GP stands for Gabriel Pereira and MS stands for Mousinho da Silveira)
2. sex- Student's Gender
3. Pstatus- It denoted Parent's cohabitation status(A- living apart, T- living together)
4. Mjob- Mother's job
5. Fjob- Father's job
6. reason- denotes what is the reason to choose this school(nominal- close to home, school reputation, course preference or other)
7. guardian- student's guardian(options: mother, father or other)
8. schoolsup- extra educational support(options: yes or no)
9. famsup- family educational support(options- yes or no)
10. paid- extra paid classes within the course subject(options: yes or no)
11. activities- any extra-curricular activities(options: yes or no)
12. nursery- if student has attended the nursery school(options: yes or no)
13. higher- if students is interested in taking higher education(options: yes or no)

2. Data Preprocess

We have taken dataset from the Machine Learning Repository (Link) which is named as "Student Performance Data Set", we converted this dataset into CSV format to make it compatible with the WEKA and R. We removed the columns containing numerical data manually so that operations can be performed in WEKA. Also, cleaning the data helps in better results and less access time when we actually run our application.

school	sex	Pstatus	Mjob	Fjob	reason	guardian	schoolsu	famsup	paid	activities	nursery	higher
GP	F	A	at_home	teacher	course	mother	yes	no	no	no	yes	yes
GP	F	T	at_home	other	course	father	no	yes	no	no	no	yes
GP	F	T	at_home	other	other	mother	yes	no	no	no	yes	yes
GP	F	T	health	services	home	mother	no	yes	no	yes	yes	yes
GP	F	T	other	other	home	father	no	yes	no	no	yes	yes
GP	M	T	services	other	reputatio	mother	no	yes	no	yes	yes	yes
GP	M	T	other	other	home	mother	no	no	no	no	yes	yes
GP	F	A	other	teacher	home	mother	yes	yes	no	no	yes	yes
GP	M	A	services	other	home	mother	no	yes	no	no	yes	yes
GP	M	T	other	other	home	mother	no	yes	no	yes	yes	yes
GP	F	T	teacher	health	reputatio	mother	no	yes	no	no	yes	yes
GP	F	T	services	other	reputatio	father	no	yes	no	yes	yes	yes
GP	M	T	health	services	course	father	no	yes	no	yes	yes	yes
GP	M	T	teacher	other	course	mother	no	yes	no	no	yes	yes
GP	M	A	other	other	home	other	no	yes	no	no	yes	yes
GP	F	T	health	other	home	mother	no	yes	no	no	yes	yes
GP	F	T	services	services	reputatio	mother	no	yes	no	yes	yes	yes
GP	F	T	other	other	reputatio	mother	yes	yes	no	yes	yes	yes
GP	M	T	services	services	course	mother	no	yes	yes	yes	yes	yes
GP	M	T	health	other	home	father	no	no	no	yes	yes	yes
GP	M	T	teacher	other	reputatio	mother	no	no	no	no	yes	yes
GP	M	T	health	health	other	father	no	yes	yes	no	yes	yes
GP	M	T	teacher	other	course	mother	no	no	no	yes	yes	yes
GP	M	T	other	other	reputatio	mother	no	yes	no	yes	yes	yes

Figure 1. Student Performance Data Set

3. Classification DM Method

Classification is one of data mining function which assigns items in class. The ultimate aim is to predict target class precisely for each item in dataset. For the classifying data, first we provide training data set to classifier and then we provide test data to test. For the training phase, a classification algorithm finds association between the values of the predictors and the values of target. Here, technique underlying for finding patterns may varied based on different algorithms. These patterns derived from training process can be applied to different data set in which class values are unknown. Further, they can be verified by comparing predicted values to known target values in test data.

Here, in this assignment we have covered several algorithms for the classification which are as follows:

1. Decision Tree
    a. C 4.5 Decision Tree (J48 Decision Tree)
    b. C 5.0 Decision Tree
2. Naive Bayes

Explanation:-

1. Decision Tree: it is a flowchart-like structure in which each decision tree is a flowchart-like structure in which each internal node represents a "test" on an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the

test, and each leaf node represents a class label (decision taken after computing all attributes). The paths from root to leaf represent classification rules.

1.a. C4.5 Decision Tree (J48 Decision Tree): it is an algorithm which is used to generate a decision tree. It was developed by Ross Quinlan and is an extension of his earlier “ID3 algorithm”. It builds decision tree from a set of training data in the same way as an ID3 algorithm, with the help of information entropy concept.

J48 is an open source Java implementation of the C4.5 algorithm in the Weka data mining tool. It has certain improvements from an ID3 algorithm such as handling continuous and discrete data and pruning trees after creation.

1.b. C5.0 Decision Tree

It is an extension of a C4.5 algorithm, developed by Ross Quinlan. The improvements in C5.0 from C4.5 are speed, less error for unknown cases.

```r
# Starting time
startTime <- proc.time()

# load the package
library(C50)

# load data
X2 <- read.csv("321.csv", header = TRUE);

# omitting null values
X2 = na.omit(X2)

# fit model
fit <- C5.0(higher~., data=X2)

# summarize the fit
print(fit)

# make predictions
predictions <- predict(fit, X2)

# summarize accuracy
table_val <- table(predictions, X2$higher)

# displaying table
table_val

# finding accuracy and displaying it
y <- "%"
paste(c((sum(diag(table_val))/sum(X) * 100 ) , y), collapse = " ")

# write the confusion matrix
write.table(table_val,"result.txt",sep="\t",row.names=FALSE)

plot(predictions, main="C5.0 Plot", ylab = "Numbers", xlab="Target Column Values")

# time consumed
proc.time() - startTime
```

Figure 2 - C5.0 implementation in R

2. Naive Bayes

Naive Bayes is an assortment of classification algorithms supported Thomas Bayes theorem. it's a family of algorithms that {every one} share common principle that is every feature classified is freelance of the worth of any feature.

We have implemented a C4.5 algorithm, Naive Bayes algorithm in Weka and C5.0 and Naive Bayes in R Studio.

Confusion Matrix: It is a matrix of actual versus predicted value, based on which we are able to calculate the accuracy of our model.

```
# Starting time
startTime <- proc.time()

# importing library
library(e1071)

# reading csv file
X2 <- read.csv("321.csv", header = TRUE);

# omitting null values
X2 = na.omit(X2)

# naive bayes model
m <- naiveBayes(higher~ ., data = X2)

# predicting output
NB_Predictions=predict(m,X2)

# predicted table
X <- table(NB_Predictions,X2$higher)

# displaying table
X

# finding accuracy and displaying it
y <- "%"
paste(c((sum(diag(X))/sum(X) * 100 ) , y), collapse = " ")

# write the confusion matrix
write.table(X,"result.txt",sep="\t",row.names=FALSE)

plot(NB_Predictions, main="Naive Bayes Plot", ylab = "Numbers", xlab="Target Column Values")

# time consumed
proc.time() - startTime
```

Figure 3- Naive Bayes implementation in R

4. Experiment Results

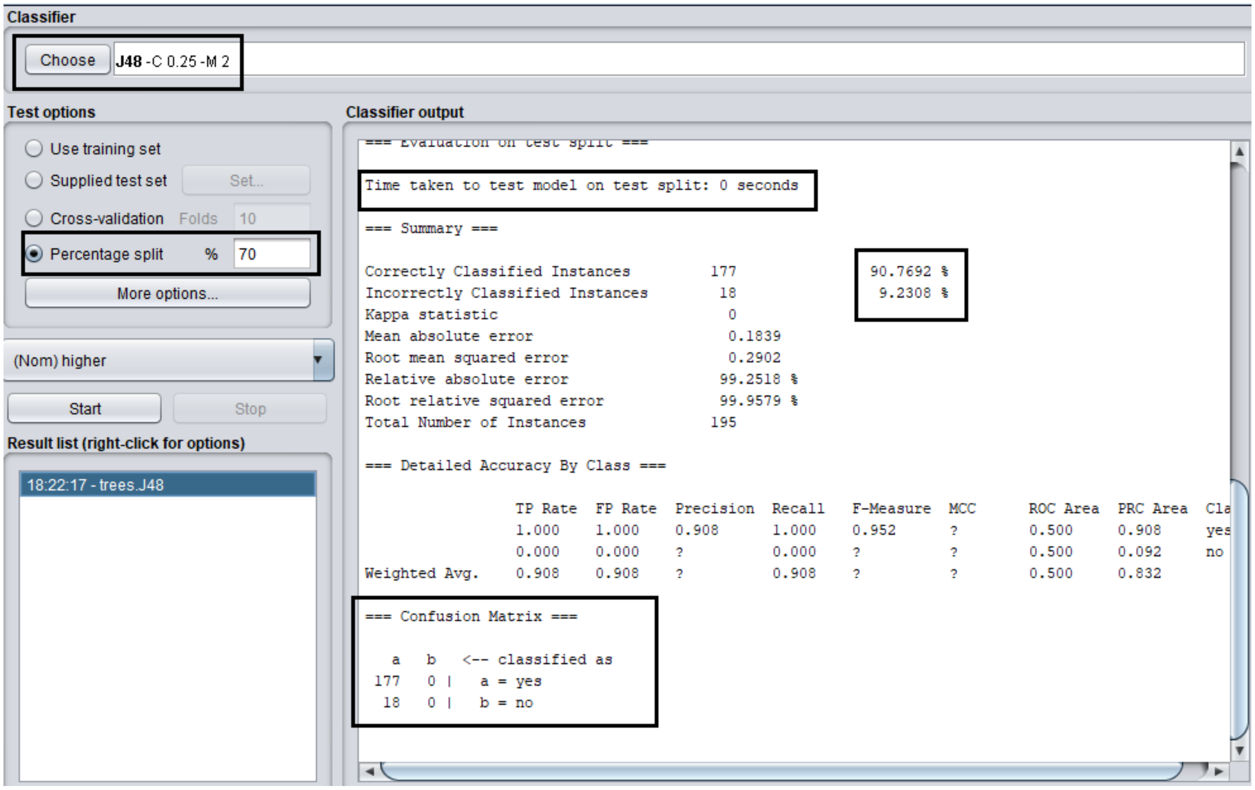

Figure 4 - J48 Result

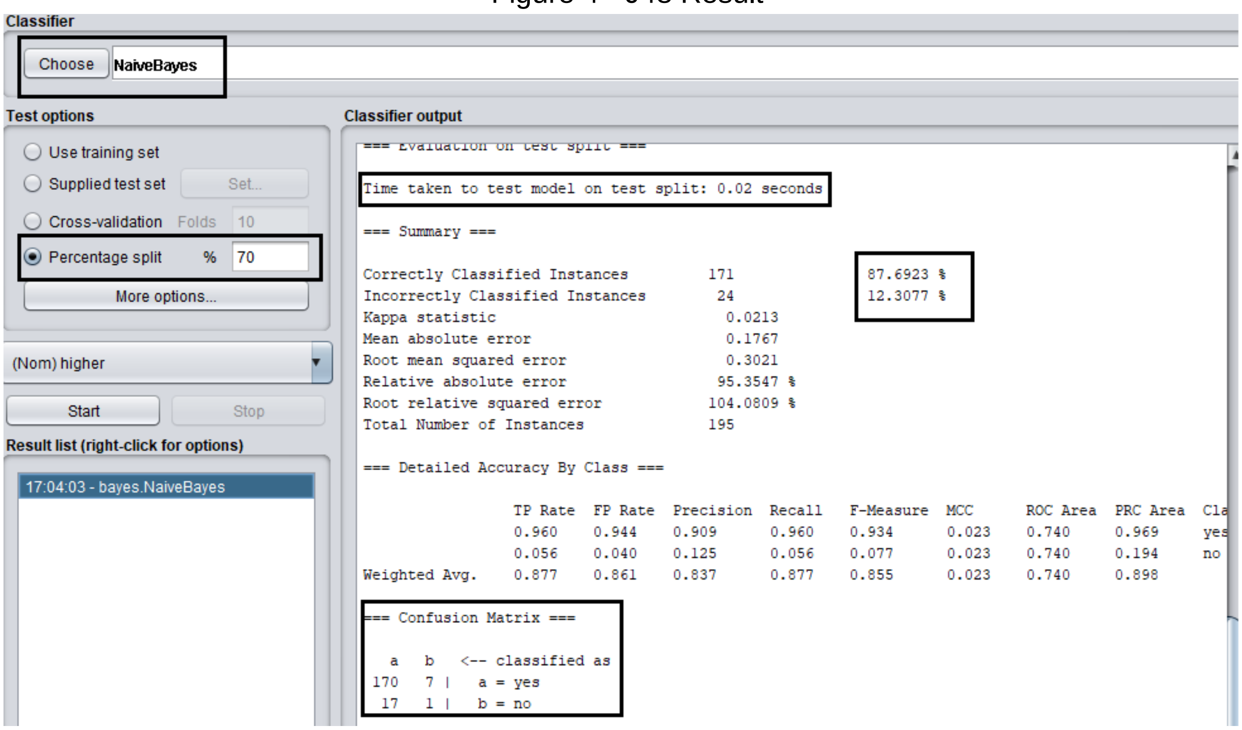

Figure 5- Naive Bayes Weka Result

```
> # Starting time
> startTime <- proc.time()
> # load the package
> library(C50)
> # load data
> X2 <- read.csv("321.csv", header = TRUE);
> # omitting null values
> X2 = na.omit(X2)
> # fit model
> fit <- C5.0(higher~., data=X2)
> # summarize the fit
> print(fit)

Call:
C5.0.formula(formula = higher ~ ., data = X2)

Classification Tree
Number of samples: 649
Number of predictors: 12

Tree size: 6

Non-standard options: attempt to group attributes

> # make predictions
> predictions <- predict(fit, X2)
> # summarize accuracy
> table_val <- table(predictions, X2$higher)
> # displaying table
> table_val
predictions  no yes
        no   10   3
        yes  59 577
> # finding accuracy and displaying it
> y <- "%"
> paste(c((sum(diag(table_val))/sum(X) * 100 ) , y), collapse = " ")
[1] "90.4468412942989 %"
> # write the confusion matrix
> write.table(table_val,"result.txt",sep="\t",row.names=FALSE)
> plot(predictions, main="C5.0 Plot", ylab = "Numbers", xlab="Target Column Values")
> # time consumed
> proc.time() - startTime
   user  system elapsed
   0.11    0.05    3.19
```

Confusion Matrix

Accuracy

Time Consumed

Figure 6- Result of C5.0 algorithm implementation

```
> # Starting time
> startTime <- proc.time()
> # importing library
> library(e1071)
> # reading csv file
> X2 <- read.csv("321.csv", header = TRUE);
> # omitting null values
> X2 = na.omit(X2)
> # naive bayes model
> m <- naiveBayes(higher~ ., data = X2)
> # predicting output
> NB_Predictions=predict(m,X2)
> # predicted table
> X <- table(NB_Predictions,X2$higher)
> # displaying table
> X

NB_Predictions  no yes
            no  10   8
            yes 59 572
> # finding accuracy and displaying it
> y <- "%"
> paste(c((sum(diag(X))/sum(X) * 100 ) , y), collapse = " ")
[1] "89.6764252696456 %"
> # write the confusion matrix
> write.table(X,"result.txt",sep="\t",row.names=FALSE)
> plot(NB_Predictions, main="Naive Bayes Plot", ylab = "Numbers", xlab="Target Column Values")
> # time consumed
> proc.time() - startTime
   user  system elapsed
   0.29    0.05    2.31
```

Confusion Matrix

Accuracy

Time Consumed

Figure 7- Result of Naive Bayes algorithm implementation

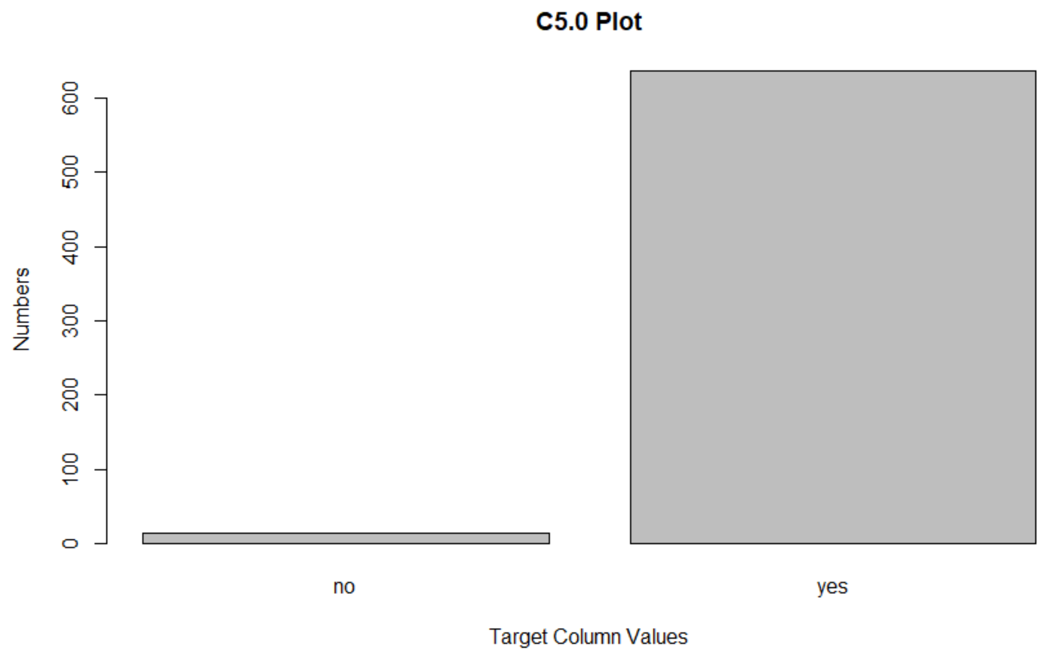

Figure 8- C5.0 Graph

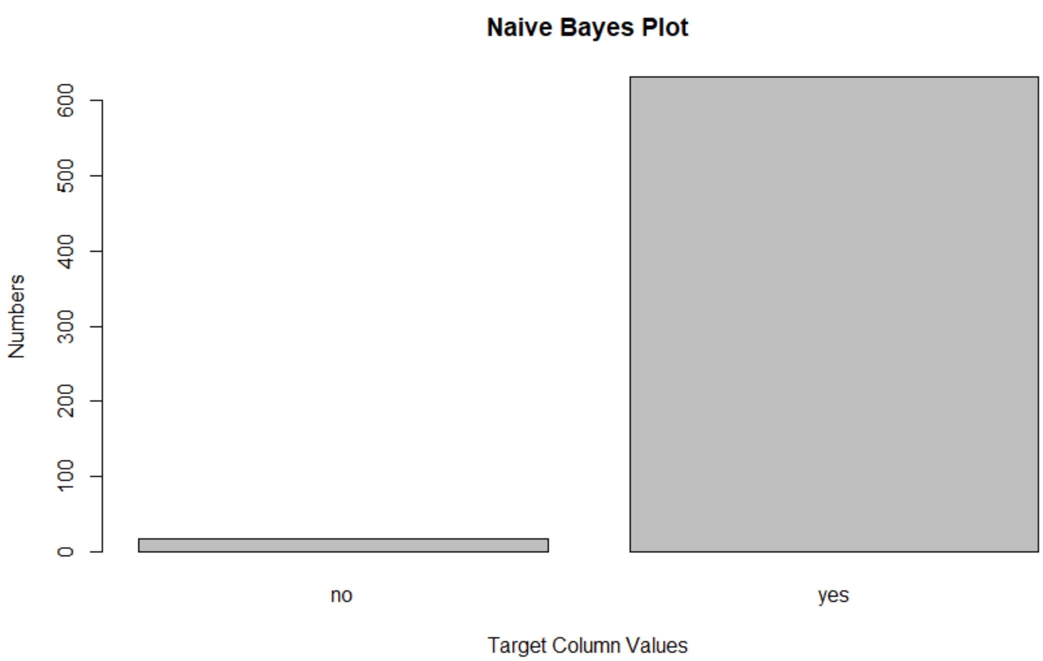

Figure 9- Naive Bayes Graph

Time Consumed (WEKA)

1. J48 Algorithm - ~0 Seconds
2. Naive Bayes Algorithm - 0.02 Seconds

Time Consumed (R Studio)
1. C5.0 Algorithm - 0.05 Seconds
2. Naive Bayes - 0.05 Seconds

Accuracy (WEKA)
1. J48 Algorithm - 90.76
2. Naive Bayes Algorithm - 87.69%

Accuracy (R Studio)
1. C5.0 Algorithm - 90.44 %
2. Naive Bayes - 89.67 %

If we look at the statistic mentioned above, we can say J48 execution is faster than Naive Bayes in WEKA and same for C5.0 and Naive Bayes in R Studio. In case of WEKA, J48 has higher accuracy (90.76%) and C5.0 has higher accuracy (90.44%) in case of R Studio. Based on above conclusion we can say that Decision Tree algorithms are much more reliable than the Naive Bayes based on time and accuracy.

5. Conclusion

In this assignment, we have created an Application through which we can find out if a student is going for higher education or not? WEKA, because of its simplicity, is very easy to use and we can load a dataset and generate results in minimal steps.

Same goes for R, the experience of using R was really good but one problem we faced in previous Assignment was repeated in this one too, that was to search for a relevant dataset. It took us a couple of hours to find a suitable dataset.

The experience of working with both the software was good and we highly recommend them for any future assignments. They are free to use, hence no validity restrictions. These softwares seem to be the best candidate for performing the data mining operations. Also running R script in R was a smooth experience and we did not face any issue in this part.

Dataset: Link

6. References

[1]W. Learning, "Weka – Graphical User Interference Way To Learn Machine Learning", *Analytics Vidhya*, 2018. [Online]. Available: https://www.analyticsvidhya.com/learning-paths-data-science-business-analytics-business-intelligence-big-data/weka-gui-learn-machine-learning/. [Accessed: 12- Apr- 2018].

[2]"R (programming language)", *En.wikipedia.org*, 2018. [Online]. Available: https://en.wikipedia.org/wiki/R_(programming_language). [Accessed: 12- Apr- 2018].

[3]"Classification", *Docs.oracle.com*, 2018. [Online]. Available: https://docs.oracle.com/cd/B28359_01/datamine.111/b28129/classify.htm#i1005746. [Accessed: 12- Apr- 2018].

[4]"C4.5 algorithm", *En.wikipedia.org*, 2018. [Online]. Available: https://en.wikipedia.org/wiki/C4.5_algorithm. [Accessed: 12- Apr- 2018].

[5]"Decision tree", *En.wikipedia.org*, 2018. [Online]. Available: https://en.wikipedia.org/wiki/Decision_tree. [Accessed: 12- Apr- 2018].

[6]D. performance, "Different decision tree algorithms with comparison of complexity or performance", *Stackoverflow.com*, 2018. [Online]. Available: https://stackoverflow.com/questions/9979461/different-decision-tree-algorithms-with-comparison-of-complexity-or-performance. [Accessed: 12- Apr- 2018].

[7]"UCI Machine Learning Repository: Student Performance Data Set", Archive.ics.uci.edu, 2018. [Online]. Available: http://archive.ics.uci.edu/ml/datasets/Student+Performance#. [Accessed: 12- Apr- 2018].