VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of computer Scienceand engineering

J-component

Programme : B.Tech

Course : Essentials of data analystics

Course code : CSE3506

Slot : G2

Title: Performance Analysis of Student

Team Members

20BCE1123 -Maddu Ganesh

20BCE1765-Vishak nair

20BRS1181-Vinay Kumar

Faculty: Dr. N M ELANGO

Abstract :

To design a system that can help to understand and perform the student performance analysis and machine learning algorithms. In order for that, it is necessary to collect the student details and their marks and store it.

The dataset chosen has collected all the student's details and stored it in a public repository. From the student performance dataset, we are performing supervised and clustering algorithms in R programming language and generating results. The student performance dataset prediction is done with respect to the student's marks. The dataset provides various information of the student which can be used for data analysing process.

1. Introduction :

The student performance is always important to know how efficient the student is performing in school. Here, the data approach is between two school in Portuguese. The dataset has 33 attributes where it contains all the student's detail and marks. The target attribute is G3 which has a strong correlation between the attributes G1 and G2.

The dataset contains the student's school, sex, age, address, family size, parent's cohabitation status, mother's education, father's education, mother's job, father's job, reason to choose the school, guardian, travel time, study time, G1, G2, G3 and many more attributes which can be useful to do the data analyzing,

For the dataset, we performed machine leaning algorithms supervised and clustering algorithms like decision tree, support vector machine and k-means clustering algorithms and infer graphical representation based on the outcome of the algorithms.

Motivation:

Each student performs uniquely in different marks and exams this can be used to analyze the student performance metrics. It is hard to determine the overall performance of the student. Hence, this analysis will be helpful to understand and implement machine learning algorithms. The students' details analysis can also be done to obtain various outputs based on the requirements.

Objective:

The main objective is to analysis and implement different machine learning algorithms in R programming language. To provide necessary outputs and graphical representation for the dataset based on the implemented algorithm. To analyze the different attributes with the marks for better analyzation of the dataset.

Problem statement:

To design a system that can help to understand and perform the student performance analysis and machine learning algorithms. In order for that, it is necessary to collect the student details and their marks and store it.

The dataset chosen have collected all the student's details and stored it in a public repository. We have chosen the dataset and providing the necessary outputs for the project.

Methodology:

The STUDENT PERFORMANCE DATASET contains numerous amounts of student's information so the predictions can be occurred to an accurate value.

R studio is used to run the machine learning algorithms in R language. RStudio is an integrated development environment for R, a programming language for statistical computing and graphics.

Here, the data exploration, data extraction, feature selection is already processed in the dataset and have been recorded in a arranged format. To apply the algorithm, the data is split into train and test data, we train the model using certain features and use it to predict the testing data. Then we calculate the performance of the system.

2.Literature Review:

1. Using Data Mining Techniques to Predict Student Performance to Support Decision Making in University Admission Systems

A data set of 2,039 students enrolled in a Computer Science and Information College of a Saudi public university from 2016 to 2019 was used to validate the problem. A model using the Linear Regression technique, which is used here for finding relationship between independent variables and a dependent variable which is used to determine the relationship between the three admission criteria as the independent variables, and the CGPA through student's first two semesters as the dependent variable. And to predict applicant's early academic performance before admitting them based on their pre-admission test scores four prediction models are used by applying four well-known data mining classification techniques, namely: Artificial Neural Network (ANN), Decision Tree, Support Vector Machine (SVM), and Naive Bayes.

Mengash, H.A. (2020). Using Data Mining Techniques to Predict Student Performance to Support Decision Making in University Admission Systems. IEEE Access, 8, pp.55462–55470. doi:10.1109/ access.2020.2981905.

2. Prediction System for Student Performance Using Data Mining Classification

The paper provides an illustration of the suggested system for predicting third-year student's fourth-year results based on their present and prior performance. Overall, a data mining technique of classification is presented to be used in the student performance analysis system to anticipate the performance of current students. C4.5, ID3, and the revised ID3 algorithm are compared while creating decision trees. Performance of the improved ID3 algorithm is superior than that of the conventional ID3 & C4.5 algorithm

Patil, R., Salunke, S., Kalbhor, M. and Lomte, R. (2018). Prediction System for Student Performance Using Data Mining Classification. 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA). doi:10.1109/iccubea.2018.8697770.

3. Naive Bayes Classification Model for the Student Performance Prediction

The student performance for the prediction analysis and applied feature extraction technique to calculate relationship between the attributes using data clustering techniques like KNN is applied and then applied Naïve Bayes classifier for the data classification and generating results for the prediction analysis which is compared with prediction of SVM algorithm. It is analysed that proposed model has high accuracy where the accuracy from SVM is about 87% and naïve bayes is about 92% also the execution time of naïve bayes classifier is compared with the SVM classifier which results that naïve bayes classifier has less execution time as compared to SVM classifier.

Tripathi, A., Yadav, S. and Rajan, R. (2019). Naive Bayes Classification Model for the Student Performance Prediction. 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT). doi:10.1109/icicict46008.2019.8993237.

4. Student Performance Prediction and Classification Using Machine Learning Algorithms

The three machine learning algorithms namely Back propagation, Support Vector Regression and Long-Short Term Memory have been used in order to predict student performances. In addition to these algorithms, Gradient Boosting Classifier is implemented in classification phase. By considering 40% of testing ratio of instances, BP, SVM and GBC achieved 80.91%, 79.38% and 74.04% of classification rates respectively both for test and training data. For 30% of testing ratio, higher results are obtained for BP, SVM and GBC as 87.78%, 83.20% and %82.44 respectively

Sekeroglu, B., Dimililer, K. and Tuncal, K. (2019). Student Performance Prediction and Classification Using Machine Learning Algorithms. Proceedings of the 2019 8th International Conference on Educational and Information Technology - ICEIT 2019. doi:10.1145/3318396.3318419.

3. Proposed Work:

3.1.Data Source

This data set is taken from kaggle which consists of student achievement in secondary education of two Portuguese schools. The data attributes include student grades, demographic, social and school related features and it was collected by using school reports and questionnaires.

3.2.Data Analytics Models

K - Means :

K Means Clustering in R Programming is an Unsupervised Non-linear algorithm that cluster data based on similarity or similar groups. It seeks to partition the observations into a pre-specified number of clusters. Segmentation of data takes place to assign each training example to a segment called a cluster. In the unsupervised algorithm, high reliance on raw data is given with large expenditure on manual review for review of relevance is given. It is used in a variety of fields like Banking, healthcare, retail, Media, etc.

Steps :

- Randomly choose k groups in the feature plan
- Group observations by minimising the distance with the centroid. It results groups with n observations
- Shift initial centre to the mean of the coordinates of the n observations within group k
- Repeat Until no observations change group
- Group observation by minimising the distance.It results in k groups with n observations

K means uses Euclidean distance for calculation:

$$distance(x, y) = \sum_{i}^{n} (x_i - y_i)^2$$

K means is really simple to implement for our student dataset which is large in size. With the help of k means we can adapt to new examples and easily cluster/group the set of marks scored by students in exams. The main disadvantage being the manual selection of 'k' number of clusters and clustering of outliers which are not needed. And also clustering using k means is sensitive to scaling of dimensions of our dataset.

Decision Tree :

A Decision Tree is a Supervised Machine Learning algorithm which looks like an inverted tree, wherein each node represents a predictor variable (feature), the link between the nodes represents a Decision and each leaf node represents an outcome (response variable).

Steps:

☑ Select the feature (predictor variable) that best classifies the data set into the desired classes and assign that feature to the root node.

☑ Step 2: Traverse down from the root node, whilst making relevant decisions at each internal node such that each internal node best classifies the data.

☑ Step 3: Route back to step 1 and repeat until you assign a class to the input data.

Two measures are used to decide the best attribute:

1. Entropy
2. Information Gain

Entropy measures the impurity or uncertainty present in the data. It is used to decide how a Decision Tree can split the data.

$$Entropy = -\sum p(x) \log p(x)$$

Information Gain (IG) is the most significant measure used to build a Decision Tree. It indicates how much "information" a particular feature/variable gives us about the final outcome.

*Information Gain = entropy(parent) – [weighted average] * entropy(children)*

With this Decision tree algorithm we were able to find that for females whose parents are together provide education support to their children most likely than for those whose parents who are living apart, and for male regardless of their parents status they get educational support probably around 50%. But the accuracy of the algorithm is only around 55% .

Support Vector Machine:

Support Vector Machine, or SVM, is a popular Supervised Learning algorithm that is used for both classification and regression problems. However, it is primarily used in Machine Learning for Classification problems. The SVM algorithm's goal is to find the best line or decision boundary for categorising n-dimensional space so that we can easily place new data points in the correct category in the future. A hyperplane is the best decision boundary. SVM selects the extreme points/vectors that aid in the creation of the hyperplane. These extreme cases are referred to as support vectors, and the algorithm is known as the Support Vector Machine.

Pros:

☑ SVM classifiers excel in high-dimensional space and have high accuracy. Because they only use a portion of the training data, SVM classifiers require less memory.

☑ When there is a large gap between classes, SVM performs reasonably well.

☑ SVM works best in high-dimensional spaces.

☑ SVM is useful when the number of dimensions exceeds the number of samples.

☑ SVM makes good use of memory.

Cons:

O SVM requires a lengthy training period, making it unsuitable for large datasets.

O Another disadvantage of SVM classifiers is their inability to handle overlapping classes.

O The SVM algorithm does not work well with large data sets.

O SVM does not perform as well when the data set contains more noise, such as overlapping target classes.

O When the number of features for each data point exceeds the number of training data samples, the SVM performs poorly.

Pseudo code:

Inputs: Determine the various training and test data.

Outputs: Determine the calculated accuracy .

Select the optimal value of cost and gamma for SVM.

While do

Implement SVM train step for each data point.

Implement SVM classify for testing data points.

End while Return

accuracy

$$H: w^T(x) + b = 0$$

Here: b = Intercept and bias term of the hyperplane equation

In D dimensional space, the hyperplane would always be D -1 operator.

Linear Regression:

A linear regression is a statistical model that analyses the relationship between a response variable (often called y) and one or more variables and their interactions (often called x or explanatory variables).

The mathematical formula of the linear regression can be written as y = b0 + b1*x + e, where:

b0 and b1 are known as the regression beta coefficients or parameters: b0 is the intercept of the regression line; that is the predicted value when x = 0. b1 is the slope of the regression line.

e is the error term (also known as the residual errors), the part of y that can be explained by the regression model

Pros

- Linear regression performs exceptionally well for linearly separable data
- Easier to implement, interpret and efficient to train
- It handles overfitting pretty well using dimensionally reduction techniques, regularization, and cross-validation
- One more advantage is the extrapolation beyond a specific data set

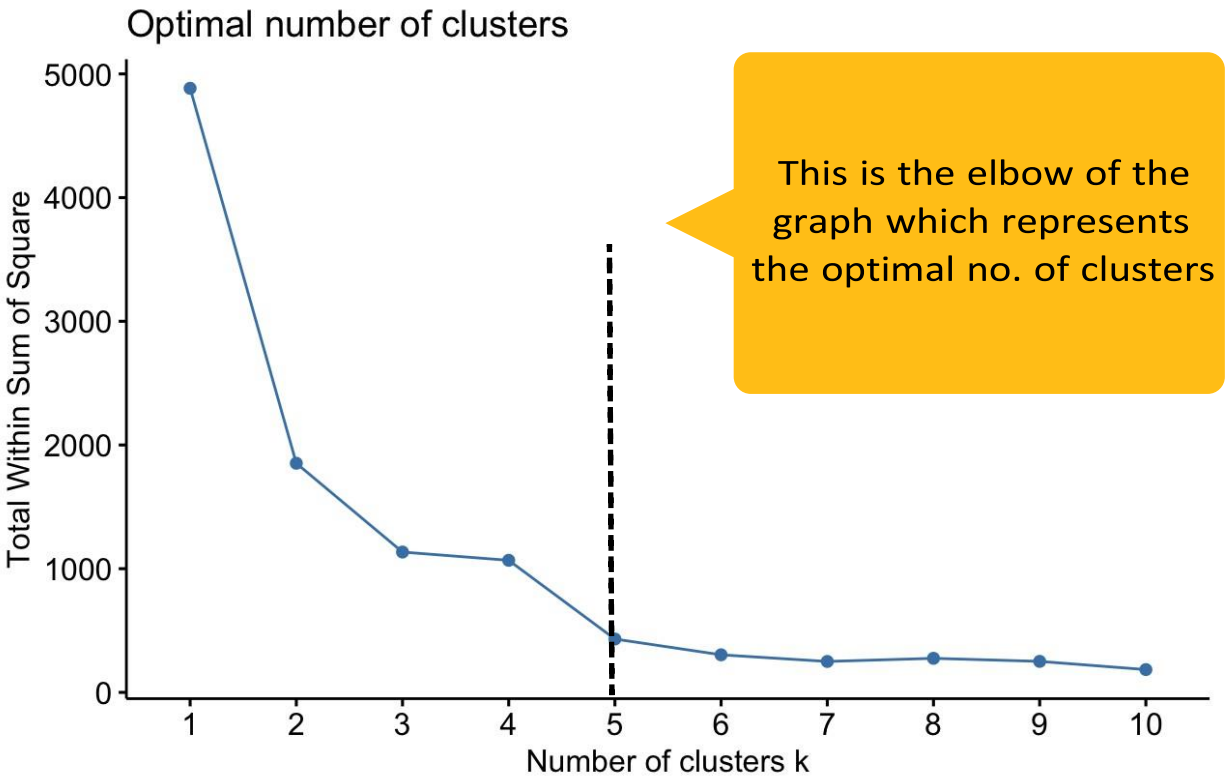
Cons

- The assumption of linearity between dependent and independent variables
- It is often quite prone to noise and overfitting
- Linear regression is quite sensitive to outliers
- It is prone to multicollinearity

4.Results and Discussions

K - Means:

```
install.packages("factoextra")
library(factoextra)
fviz_nbclust(dfx,kmeans,method = "wss",k.max = 10) #For finding the correct number
                                                   #of clusters required for kmeans
```




```
dfx = data.frame(g1 = df_csv$G1)
km.res=kmeans(dfx, 5, iter.max = 10, nstart = 1)
```

```
K-means clustering with 5 clusters of sizes 92, 160, 225, 81, 91

Cluster means:
         g1
1  7.108696
2  9.593750
3 12.951111
4 15.975309
5 11.000000

Clustering vector:
  [1] 1 2 3 3 5 3 3 2 4 3 3 2 3 3 3 4 3 3 1 3 3 5 3 2 2 2 5 5 3 3 2 4 3 3 3 5 3 3 5 3 5 2 3 2 2 2 3
 [48] 4 5 3 3 4 2 3 3 3 4 4 3 4 4 2 3 3 3 4 5 2 5 4 3 5 3 3 5 5 3 3 2 3 5 2 3 3 3 3 3 4 3 2 2 3 3 3
 [95] 5 3 2 3 3 3 2 4 3 5 4 2 2 3 3 4 3 5 1 4 2 4 4 3 3 3 3 3 3 3 3 2 2 2 3 3 2 2 5 5 3 4 2 3 3 3 2
[142] 3 5 3 3 2 2 3 1 2 2 4 3 3 2 2 3 5 5 3 5 2 3 5 2 3 5 3 1 2 5 3 2 2 1 1 1 2 1 1 2 4 3 4 2 4 5 3
[189] 3 2 3 5 2 5 5 5 4 3 3 5 2 3 3 2 3 3 2 5 3 3 3 2 3 5 3 5 3 3 3 1 5 3 3 5 3 3 3 3 3 3 5 2 3 3 5
[236] 3 3 2 4 1 4 2 2 4 3 3 3 3 2 3 4 2 3 2 2 1 1 2 3 5 3 3 2 1 5 4 4 4 3 3 3 5 3 5 2 3 4 5 2 1 2 5
[283] 1 2 1 3 3 2 3 3 2 5 5 5 5 5 4 2 2 2 4 2 2 4 5 1 4 5 2 2 1 5 4 3 4 3 4 3 3 5 3 3 2 2 2 5 3 4 3
[330] 3 3 3 4 3 3 4 4 4 4 3 3 3 3 4 4 3 3 1 4 4 2 3 2 2 3 5 4 3 3 4 5 5 3 5 3 4 3 5 1 2 1 2 3 5 4 3
[377] 3 3 4 5 3 4 5 5 5 2 4 4 2 2 3 5 3 3 3 3 4 2 5 3 4 3 5 3 3 2 2 2 2 3 3 3 3 2 5 2 4 3 3 1 3 1 2
[424] 2 3 2 2 4 1 2 2 2 1 3 2 2 1 3 2 3 1 1 3 1 2 1 5 1 4 5 4 2 2 2 2 2 3 2 3 2 5 3 3 3 5 2 2 3 3 3
[471] 3 3 3 2 2 3 2 3 1 2 2 2 2 2 1 5 2 1 1 1 2 1 1 2 1 3 3 2 3 4 1 3 2 3 3 5 5 5 2 4 4 5 1 1 1 2 3
[518] 3 1 1 1 1 1 1 1 4 3 2 2 5 1 2 2 3 5 2 3 3 5 2 2 5 2 2 3 1 3 2 5 4 3 3 2 5 2 3 2 1 1 3 2 2 5 1
[565] 5 2 2 1 1 1 1 1 1 1 1 2 2 2 1 2 1 1 1 1 2 1 1 1 1 1 1 3 5 3 4 4 4 2 2 3 3 1 5 1 2 1 4 5 1 5 1
[612] 3 2 2 3 4 3 4 4 3 4 3 1 4 1 2 1 2 2 1 4 2 1 3 4 1 4 1 3 1 1 3 1 1 2 4 5 2 2

Within cluster sum of squares by cluster:
[1] 136.91304  38.59375 152.46222  87.95062   0.00000
 (between_SS / total_SS =  91.5 %)
> aggregate(dfx, by=list(cluster=km.res$cluster), mean)
  cluster        g1
1       1  7.108696
2       2  9.593750
3       3 12.951111
4       4 15.975309
5       5 11.000000
```

Here we have used K means for the G1 Attribute/Marks of the student in G1 exam.

For abstraction let's consider only the first 10 rows.

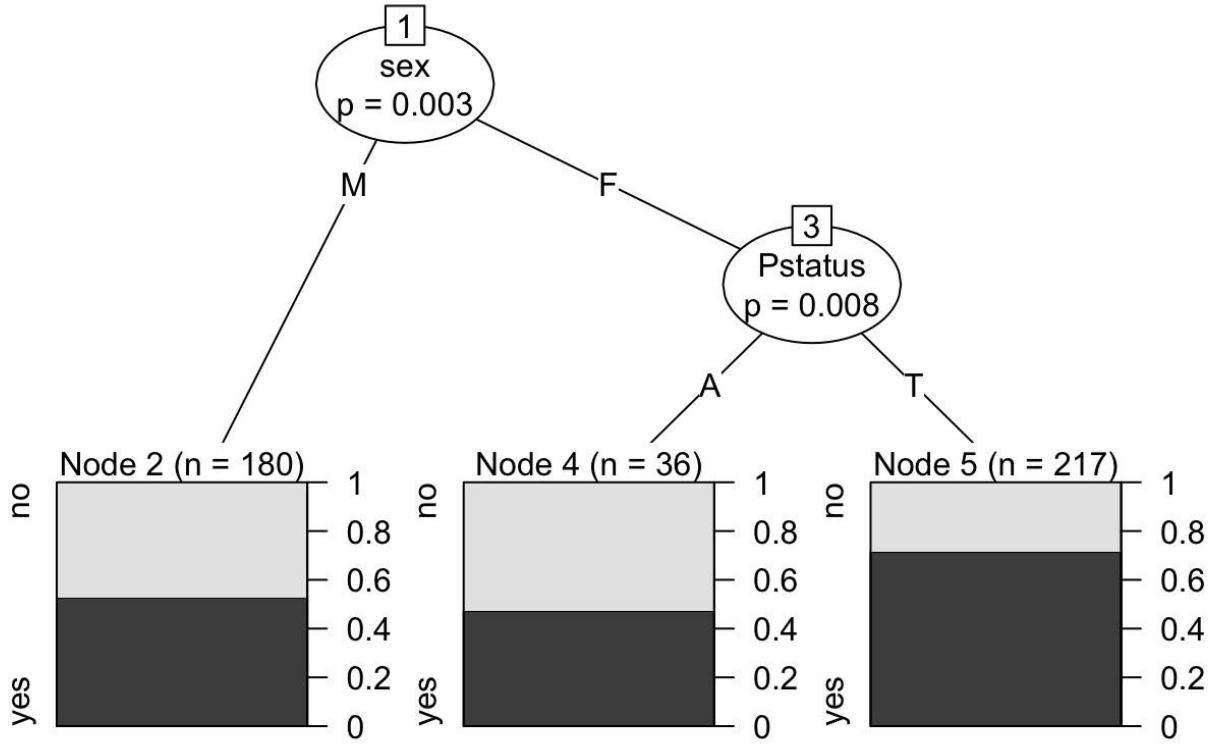
```
K-means clustering with 5 clusters of sizes 2, 2, 4, 1, 1

Cluster means:
     g1
1 14.50
2  9.50
3 11.75
4 13.00
5  0.00

Clustering vector:
 [1] 5 2 3 1 3 3 4 2 1 3

Within cluster sum of squares by cluster:
[1] 0.50 0.50 0.75 0.00 0.00
 (between_SS / total_SS =  98.9 %)
```

	g1	cluster
1	0	5
2	9	2
3	12	3
4	14	1
5	11	3
6	12	3
7	13	4
8	10	2
9	15	1
10	12	3

Decision tree:

```
install.packages('party')
library(datasets)
library(caTools)
library(party)
library(dplyr)
library(magrittr)
```

```
> dfx2 = df_csv[,c(2,6,17)]
> head(dfx2)
  sex Pstatus famsup
1   F       A     no
2   F       T    yes
3   F       T     no
4   F       T    yes
5   F       T    yes
6   M       T    yes
> dfx2$famsup = factor(dfx2$famsup)
> dfx2$Pstatus = factor(dfx2$Pstatus)
> dfx2$sex = factor(dfx2$sex)
> sample_data = sample.split(dfx2, SplitRatio = 0.8)
> train_data <- subset(dfx2, sample_data == TRUE)
> test_data <- subset(dfx2, sample_data == FALSE)
>
> class(dfx2$sex)
[1] "factor"
> model<- ctree(famsup ~ sex + Pstatus, train_data)
> plot(model)
```



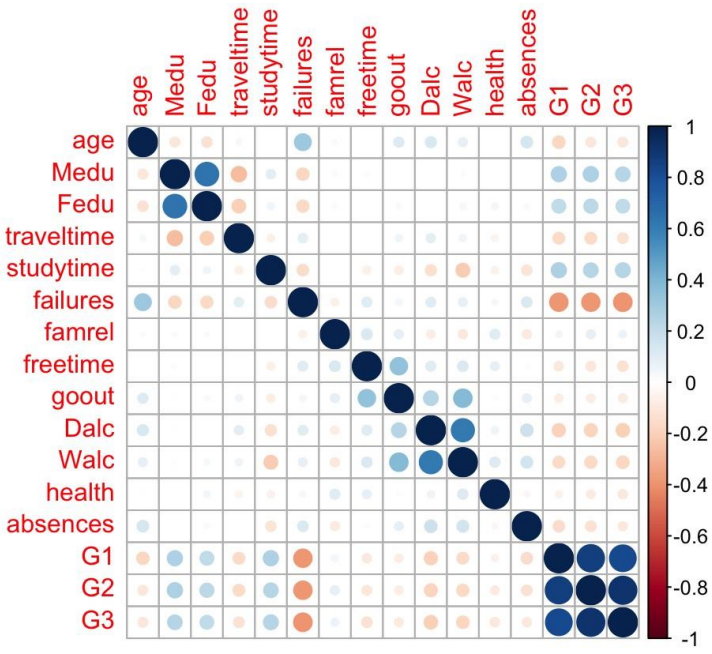
```
install.packages('caret')
library(caret)
predict_model<-predict(ctree(famsup ~ sex + Pstatus, train_data), test_data)
```

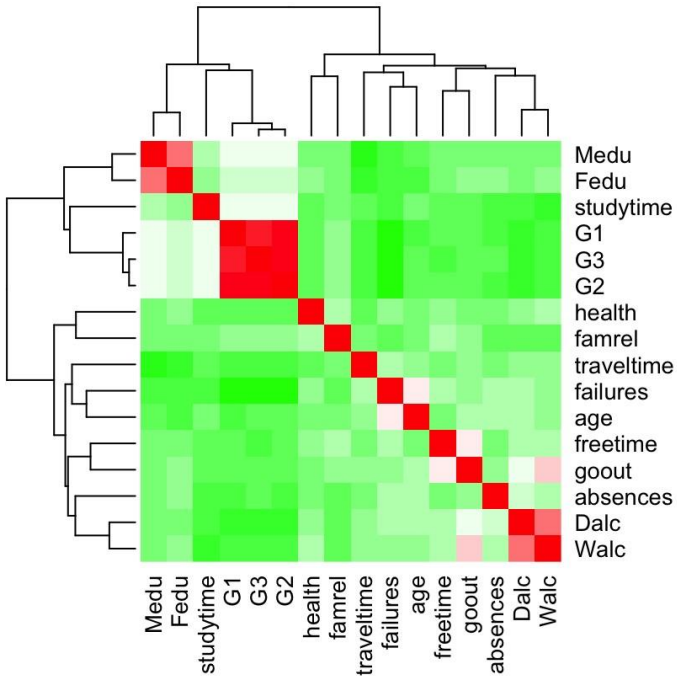
```
> m_at <- table(test_data$famsup, predict_model)
> m_at
     predict_model
       no yes
  no    3  82
  yes  15 116
> ac_Test = sum(diag(m_at)) / sum(m_at)
> print(paste('Accuracy for test is found to be', ac_Test))
[1] "Accuracy for test is found to be 0.550925925925926"
```

Support Vector Machine:

```
> stud_por <- read.csv("/Users/aktshaykumar/Documents/Foundations of Data Analytics/data_csv.csv",header = T)
> stud_por_df <- data.frame(stud_por)
> mydata.cor = cor(stud_por_df[, unlist(lapply(stud_por_df, is.numeric))])
```

```
> corrplot(mydata.cor)
> palette = colorRampPalette(c("green", "white", "red")) (20)
> heatmap(x = mydata.cor, col = palette, symm = TRUE)
```



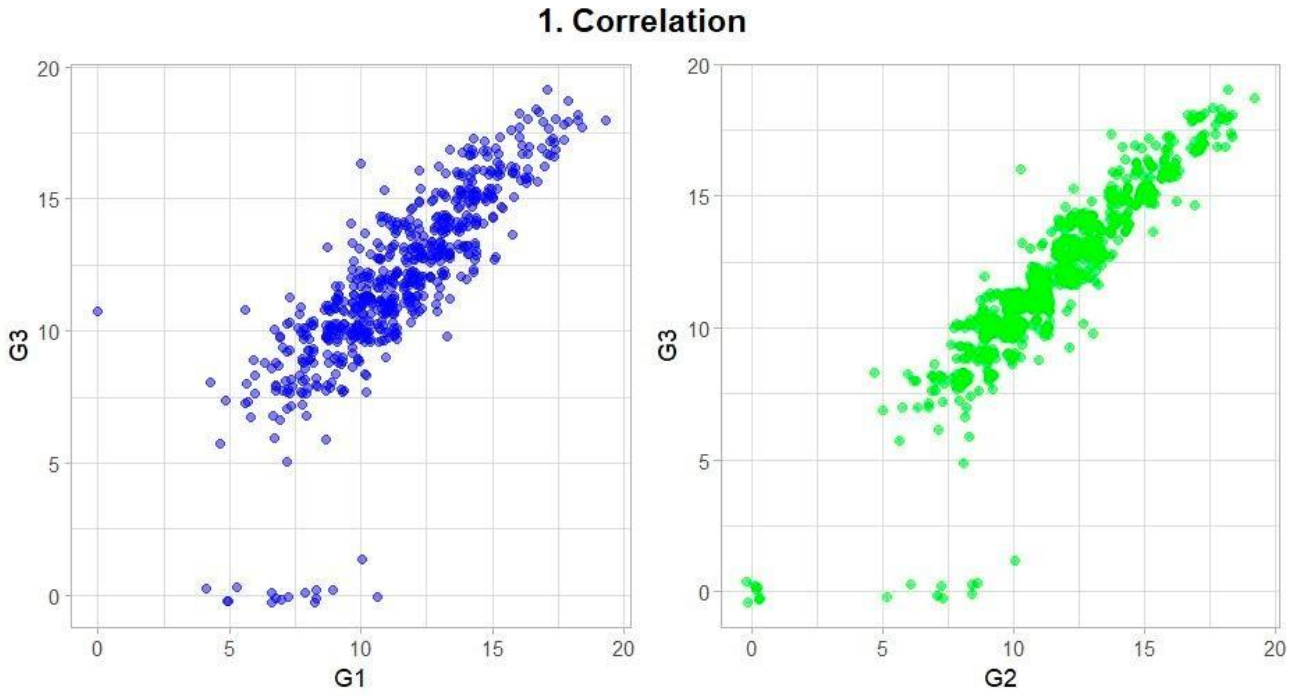


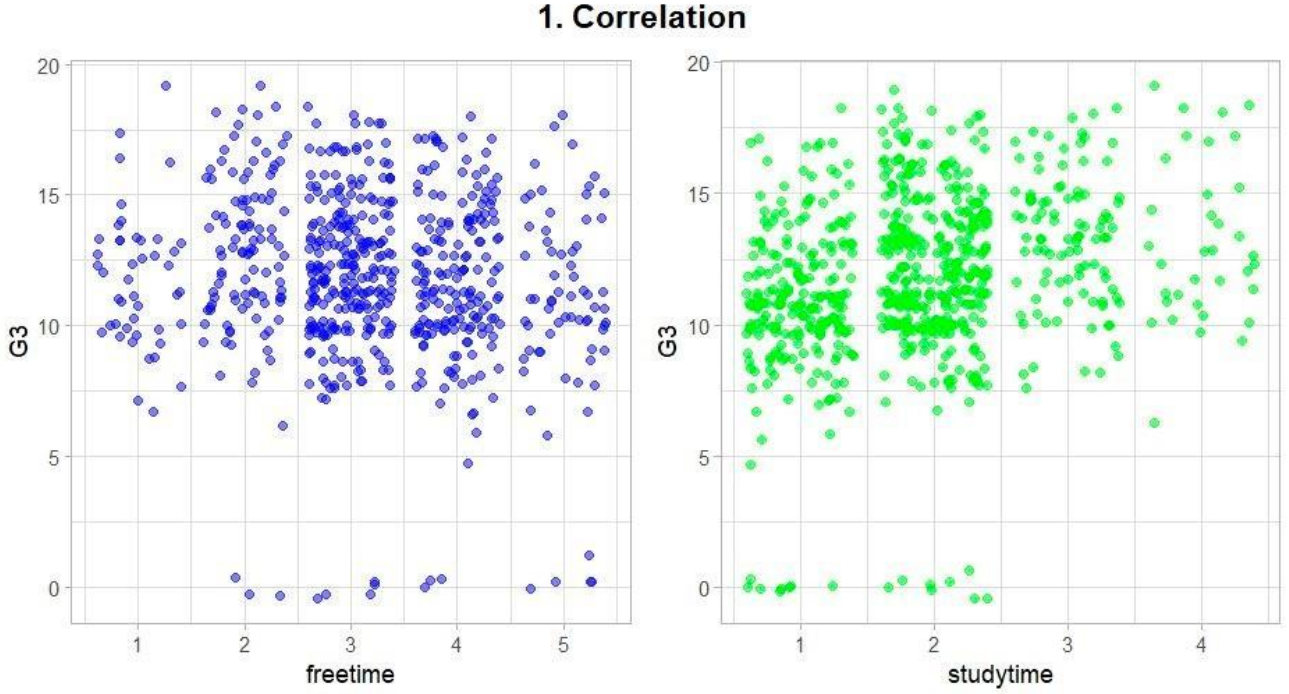

```
> # Equivalent to using geom_bar(stat = "bin")
> ggplot(stud_por_df, aes(x = G3)) +
+   geom_bar()
> # Set seed and create assignment
> set.seed(123)
> assignment <- sample(1:3, size = nrow(stud_por_df), prob = c(70,15,15), replace = TRUE)
> # Create a train, validation and tests from the original data frame
> stud_por_train <- stud_por_df[assignment == 1, ]    # subset grade to training indices only
> stud_por_valid <- stud_por_df[assignment == 2, ]  # subset grade to validation indices only
> stud_por_test <- stud_por_df[assignment == 3, ]
> regressor = svm(formula = G3~ ., data = stud_por_train, cost = 1)
> regressor

Call:
svm(formula = G3 ~ ., data = stud_por_train, cost = 1)


Parameters:
   SVM-Type:  eps-regression
 SVM-Kernel:  radial
       cost:  1
      gamma:  0.024
    epsilon:  0.1


Number of Support Vectors:  313

> pred <- predict(object = regressor,   # model object
+                 newdata = stud_por_test)
> pred
   2   26   34   37   53   58   65   67   68   69   71   73   84   97  115  134  137  138  145  150
11.3 11.1 12.9 14.4  9.7 15.0 12.1 11.0 10.0 10.4 12.6 11.2 12.7 10.7  9.6 12.0  9.8 11.3 12.0 10.6
 151  167  174  181  183  216  219  223  238  240  246  250  256  275  276  281  295  296  300  301
 9.8  9.9  8.2  9.9 12.4 12.0 13.4 12.7  9.7  8.7 13.9 12.6  8.3  9.8 11.2 10.2 11.7 13.7 12.5 15.5
 303  304  313  316  317  324  333  334  360  366  377  382  384  393  394  407  412  443  446  447
11.5 13.5 14.9 14.4 16.6 11.0 18.3 14.3 15.0 15.0 13.8 15.4 11.2 13.8 14.8 13.1 14.9 12.5  9.7 11.5
 457  458  470  472  474  480  482  490  494  500  515  518  520  534  536  538  541  543  545  549
13.9 10.4 13.1 11.5  8.7  9.5  9.9  8.7 10.0 15.7  7.6 13.9  8.1 15.3 10.8 13.1 11.0 11.4 11.5 10.2
 563  568  576  582  606  608  616  618  623  631  632  634  638  639  642  647
12.8  4.2 11.2  8.1  4.2 12.1 14.2 17.5  9.8 16.2 11.8 14.2  7.4 15.9 16.8 12.0
```

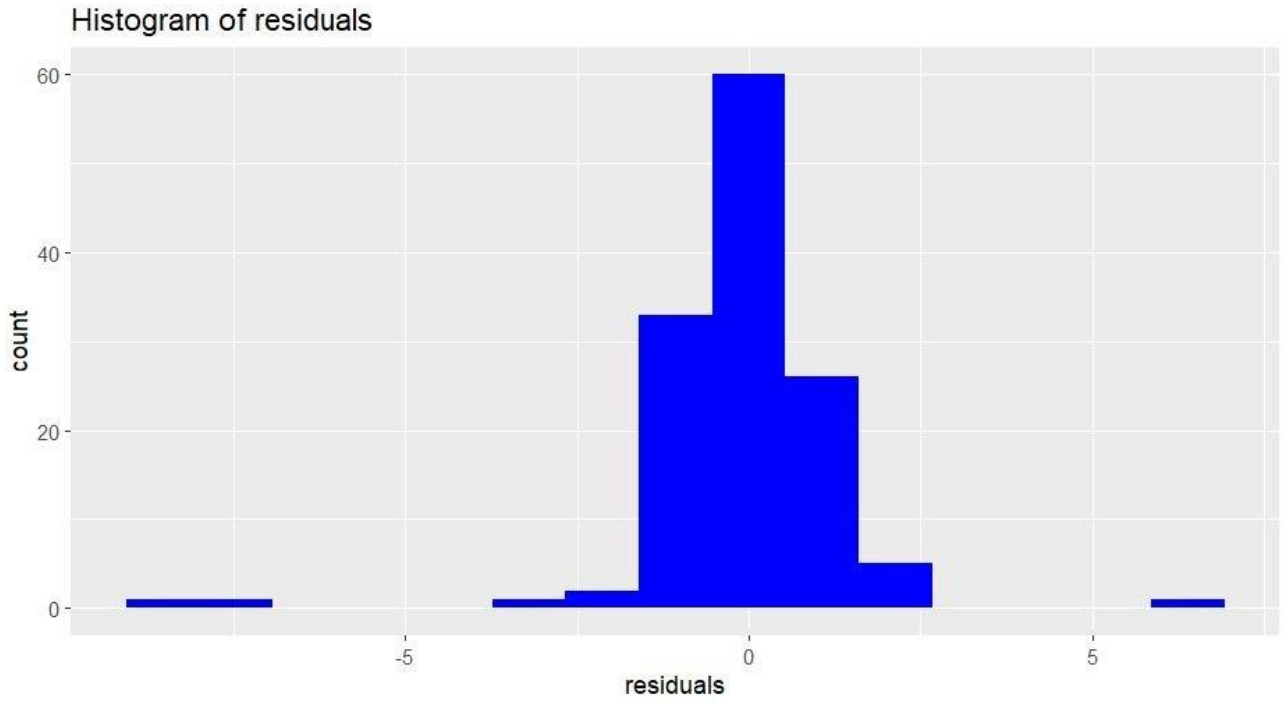
```
> # Compute the RMSE
> rmse(actual = stud_por_test$G3,
+      predicted = pred)
[1] 1.5
> # Compute the MAE
> mae(actual = stud_por_test$G3,
+     predicted = pred)
[1] 0.91
```

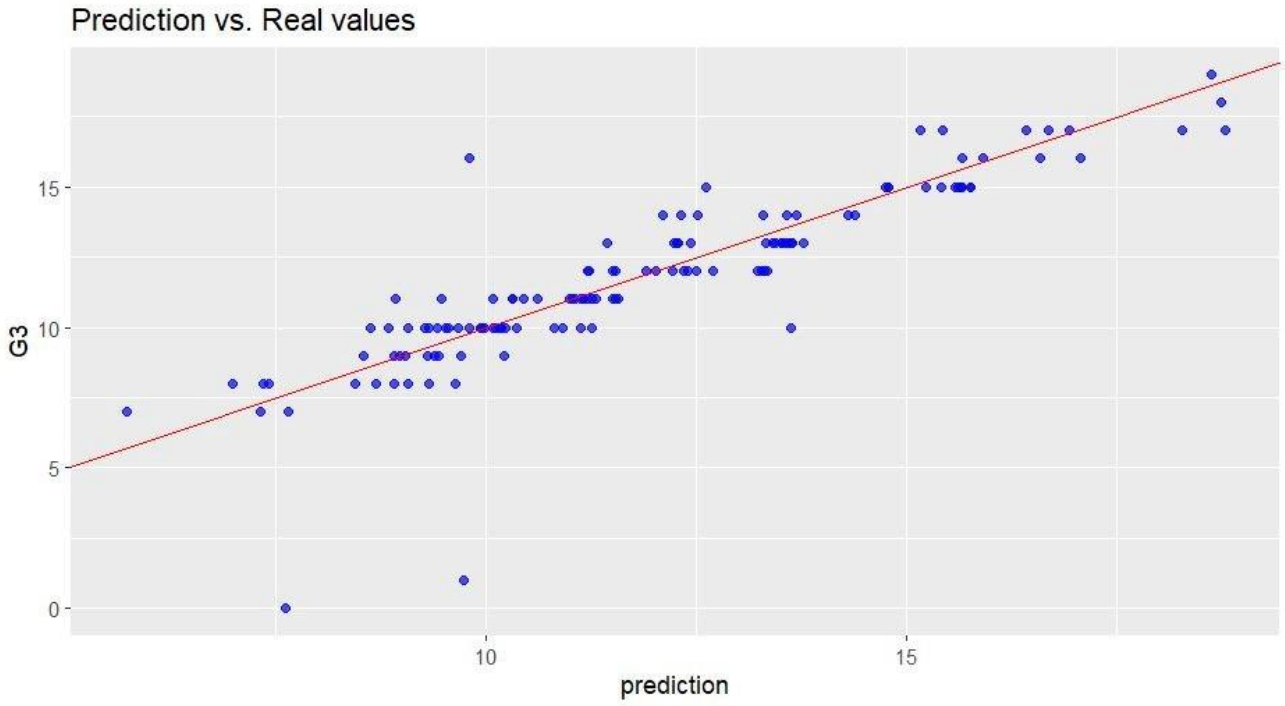
Linear Regression:

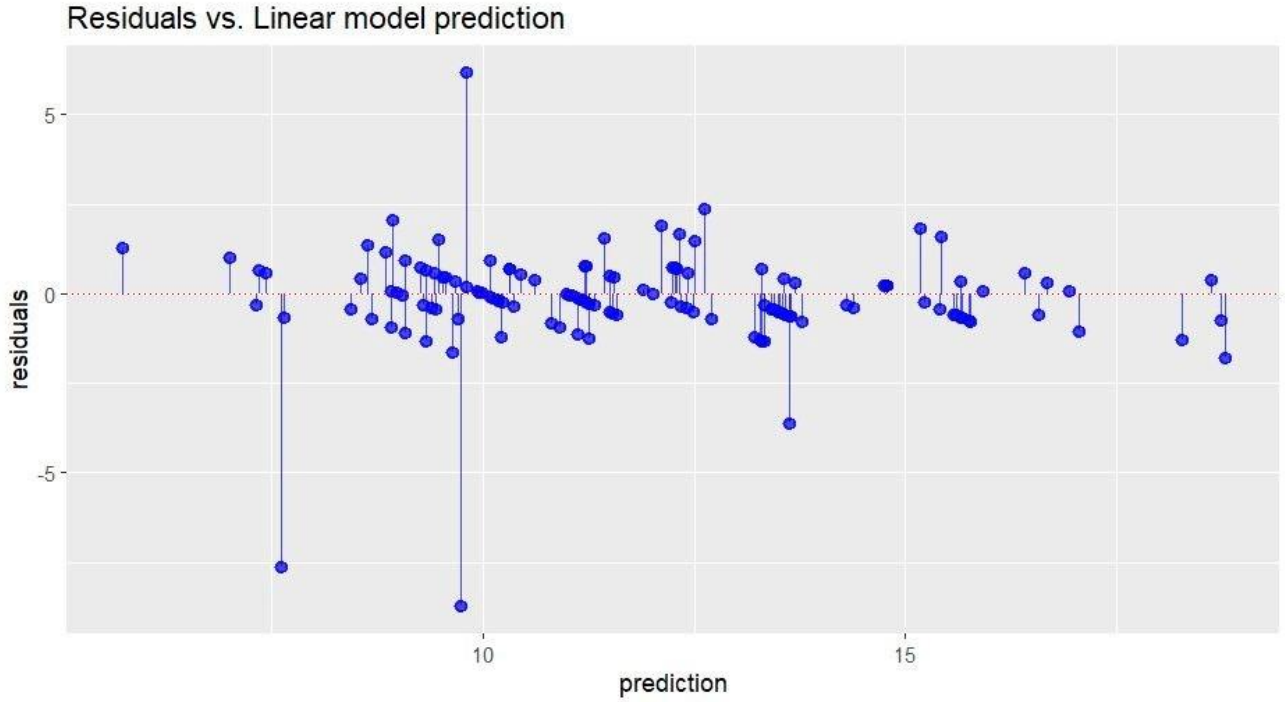
```
> formula_1 <- as.formula("G3 ~ age + Medu + Fedu  + traveltime + studytime + failures + fam
rel+ freetime+ goout + Dalc + Walc + health + absences + G1 +G2 ")
>
> model_1 <- lm(formula_1, data = Data_train)
> summary(model_1)

Call:
lm(formula = formula_1, data = Data_train)

Residuals:
    Min      1Q  Median      3Q     Max
-9.0517 -0.4531 -0.0455  0.6314  3.1343

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.190051   0.908338  -0.209 0.834354
age          0.034980   0.048122   0.727 0.467625
Medu        -0.077180   0.063587  -1.214 0.225402
Fedu         0.069236   0.065095   1.064 0.288012
traveltime   0.037258   0.075167   0.496 0.620344
studytime    0.075167   0.067692   1.110 0.267349
failures    -0.180473   0.097983  -1.842 0.066081 .
famrel      -0.056845   0.056038  -1.014 0.310885
freetime     0.001885   0.055826   0.034 0.973072
goout       -0.017311   0.052692  -0.329 0.742650
Dalc        -0.063588   0.076847  -0.827 0.408369
Walc        -0.035376   0.058002  -0.610 0.542197
health      -0.049737   0.038219  -1.301 0.193725
absences     0.027871   0.012351   2.257 0.024460 *
G1           0.134731   0.040289   3.344 0.000887 ***
G2           0.898240   0.036739  24.449  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.205 on 503 degrees of freedom
Multiple R-squared:  0.8693,    Adjusted R-squared:  0.8654
F-statistic: 223.1 on 15 and 503 DF,  p-value: < 2.2e-16
```

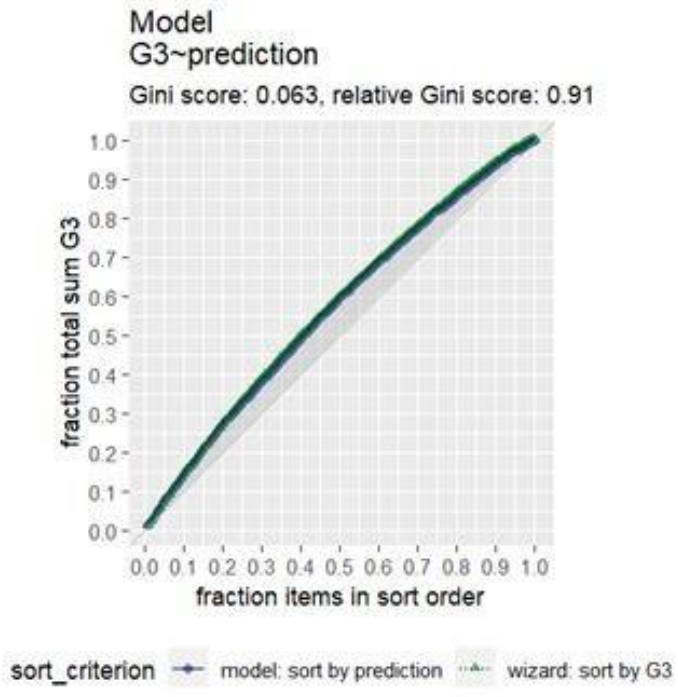

1. Correlation
G3
G1
G2
0
5
10
15
20


1. Correlation
G3
freetime
studytime
0
5
10
15
20
1
2
3
4


Histogram of residuals
60
40
20
0
count
-5
0
5
residuals


Prediction vs. Real values
15
10
5
0
G3
10
15
prediction


Residuals vs. Linear model prediction
residuals
5
0
-5
10
15
prediction


Model
G3~prediction
Gini score: 0.063, relative Gini score: 0.91
fraction total sum G3
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
fraction items in sort order
sort_criterion
model: sort by prediction
wizard: sort by G3

4. Conclusion

It focuses on analysing student academic growth using machine learning techniques. For analysis Support Vector Machine, Multiple Linear Regression, Decision tree and K means classifier are used. This process helps instructors easily determine student performance and plan better ways to improve their academic performance. In the future, additional features will be added to the dataset to improve accuracy. the target attribute G3 has a strong correlation with attributes G2 and G1. This occurs because G3 is the final year grade while G1 and G2 correspond to the 1st and 2ndperiod grades. It is more difficult to predict G3 without G2 and G1.

5. References

Xu, J., Moon, K.H. and van der Schaar, M. (2017). A Machine Learning Approach for Tracking and Predicting Student Performance in Degree Programs. IEEE Journal of Selected Topics in Signal Processing, 11(5), pp.742–753. doi:10.1109/jstsp.2017.2692560.

Conijn, R., Snijders, C., Kleingeld, A. and Matzat, U. (2017). Predicting Student Performance from LMS Data: A Comparison of 17 Blended Courses Using Moodle LMS. IEEE Transactions on Learning Technologies, 10(1), pp.17–29. doi:10.1109/tlt.2016.2616312.

Shahiri, A.M., Husain, W. and Rashid, N.A. (2015). A Review on Predicting Student's Performance Using Data Mining Techniques. Procedia Computer Science, 72, pp.414–422. doi:10.1016/j.procs.2015.12.157.

Mengash, H.A. (2020). Using Data Mining Techniques to Predict Student Performance to Support Decision Making in University Admission Systems. IEEE Access, 8, pp.55462–55470. doi:10.1109/access.2020.2981905.

Ahmed, A.M., Rizaner, A. and Ulusoy, A.H. (2016). Using data Mining to Predict Instructor Performance. Procedia Computer Science, [online] 102, pp.137–142. doi:10.1016/ j.procs.2016.09.380.

Kaunang, F. J., & Rotikan, R. (2018). Students' Academic Performance Prediction using Data Mining. In Paper presented at the 2018 third international conference on informatics and computing (ICIC).

Liao, S. N., Zingaro, D., Thai, K., Alvarado, C., Griswold, W. G., & Porter, L. (2019). A robust machine learning technique to predict low-performing students. ACM Transactions on Computing Education (TOCE), 19(3), 1–19.

Saa, A. A. (2016). Educational data mining & students' performance prediction. International Journal of Advanced Computer Science and Applications, 7(5), 212–220.

Chen, H. (2018). Predicting student performance using data from an Auto-grading system. University of Waterloo.

Costa-Mendes, R., Oliveira, T., Castelli, M., & Cruz-Jesus, F. (2020). A machine learning approximation of the 2015 Portuguese high school student grades: A hybrid approach. Education and Information Technologies, 1–21

Kabakchieva, D. 2013. Predicting Student Performance by Using Data Mining Methods for Classification. Cybernetics and Information Technologies. 13, 1 (2013), 61-72.

Agrawal, H. and Mavani, H. 2015. Student Performance Prediction using Machine Learning. International Journal of Engineering Research and. V4, 03 (2015).

Pandey, M. and Taruna, S. 2016. Towards the integration of multiple classifier pertaining to the Student's performance prediction. Perspectives in Science. 8, (2016), 364-366.

Wakelam, E., Jefferies, A., Davey, N. and Sun, Y. 2019. The potential for student performance prediction in small cohorts with minimal available attributes. British Journal of Educational Technology. 51, 2 (2019), 347-370.

Sandoval, A., Gonzalez, C., Alarcon, R., Pichara, K. and Montenegro, M. 2018. Centralized student performance prediction in large courses based on low-cost variables in an institutional context. The Internet and Higher Education. 37, (2018), 76-89.

Kotsiantis, S., Tselios, N., Filippidi, A. and Komis, V. 2013. Using learning analytics to identify successful learners in a blended learning course. International Journal of Technology Enhanced Learning. 5, 2 (2013), 133.

Yang, F. and Li, F. 2018. Study on student performance estimation, student progress analysis, and student potential prediction based on data mining. Computers & Education. 123,(2018), 97-108.

Tan, M. and Shao, P. 2015. Prediction of Student Dropout in E-Learning Program Through the Use of Machine Learning Method. International Journal of Emerging Technologies in Learning (iJET). 10, 1 (2015), 11.

Lykourentzou, I., Giannoukos, I., Nikolopoulos, V., Mpardis, G. and Loumos, V. 2009. Dropout prediction in e-learning courses through the combination of machine learning techniques. Computers & Education. 53, 3 (2009), 950-965.

Márquez-Vera, C., Cano, A., Romero, C., Noaman, A., Mousa Fardoun, H. and Ventura, S. 2015. Early dropout prediction using data mining: a case study with high school students. ExpertSystems. 33, 1 (2015), 107-124.