Data Mining and Predictive Analytics group project

Mane Sahakyan, Anna Gaplanyan, Nare Sedrakyan

08, May, 2021

Contents

Overview

The usage of data mining in education, especially in the online learning aspect, has become a necessity. The database gained from the elearning environment(Moodle system) needs to be processed, and we need data mining tools in order to get insights into the data. Our project aims to represent the whole phases of processing the e-learning dataset and putting into use data mining techniques such as visualization, clustering, and classification for the Moodle log records. These techniques will help predict students' academic performance, group students based on their similar features that will be formed by processing the data. The results of our project can be used by universities that use Moodle learning management system to create a new education system that will inform students that they are at risk of failing. Secondly, universities may predict the low-performing students and sign learning contracts with them. That contract allows students to attend more office hours, prioritize homework and reading, and show more improvement on exam performance. "These learning contracts are low cost, lof effort tool to increase student commitment, boost academic performance, and encourage self-direction (Frank & Scharff, 2013).

Literature review

In order to do deep level analyzes and exploring meaningful insights, we have done research by reading different articles regarding the same topic. **Visualization** During the project, we use the R programming language in order to create different graphical representations of the data. Additionally, there is a graphical interactive student monitoring and tracking system tool (GISMO) that extracts tracking data from Moodle and generates graphical representations that can be explored by course instructors to examine various aspects of distance classes (Cristobal Romero & Ventura, 2007). **Cluster Analysis** The clustering method aims to assort together the items that display similarities in certain aspects. In clustering analyses, distinct metrics are usually utilized to identify things that have similar attributes (Cristobal Romero & Ventura, 2007). The critical factor of clustering analysis is deciding how many separate clusters will be connected. In a data that is grouped into an accurate number of clusters, the items in the same cluster are expected to have more similarities with those in the same cluster and fewer similarities with those in other clusters (Ryan

Baker & Siemens, 2014). Since, in our case, it is an educational environment, we used the clustering method for grouping students based on their features such as the number of different days that the student logs in Moodle, the number of views of the status of a submission, views of quiz attempt, views of feedback, etc. We mainly have chosen the activities for modeling based on the number of students who did these actions. For **Clustering analysis**, we used the tool K -Means algorithm based on the specific conditions. **Classification analysis** The purpose of classification analysis is to predict the value of the target feature on a categorical level by using another group of features(AKCAPINAR & BAYAZIT, 2019, p. 410). In supervised learning, random forest (RF) is one of the statistical learning theories, and the approach is applied to make predictions with multiple decision trees and uses voting to obtain the final prediction results(Hung et al., 2020, p.4). For the project, we use **Random forest**, **Support Vector Machine**, and **Logistic regression** to predict the students who are low performers and the students who are high performers based on their features. Besides, we used **KNN regression** to make predictions of students' grades.

Research Methodology

For our project, we used Moodle data system of the American University of Armenia. From this dataset, we select only undergraduate quantitive courses data. After processing data (removing duplications, missing values and intersecting the user ids with students user ids, defining the number of different days that the student log in Moodle, and selecting number of views of submission status, number of times that the submission is updated, number of views of submission form, number of times that the submission is created, number of views of quiz attempts, number of views of quiz attempts' summaries, number of times that the quiz attempt is submitted, number of times that the quiz attempt is started, number of times that the quiz attempt is reviewed, number of views of grade user report, number of views of feedbacks, number of views of discussions, number of views of a course, number of times that the course activity completion is updated, number of times that the submission is submitted, and number of times that the file is uploaded) we formed the data for data mining analysis. In the **visualization part** we used a bar plot to demonstrate the frequencies of activities done by students. wE used **ggcorrplot** function in order to represent the correlation coefficients between grade and the remaining variables in our data. Additionally, **ggpairs** function helped us determine whether there is a linear relationship between students' grades and the rest columns of our data. For **Clustering Analysis**, we use the **K-means** algorithm as our data is unlabeled. The aim of using the K-means algorithm is to group students in different clusters based on their features. We used the **fviz_nbclust()** function from **factoextra** package in order to create **Elbow curve** and to be able to define the optimal value of **K**, which stands for the number of clusters. For **Classification analysis** we divided students into two groups, low performers(Score<=50) and high performers (Score > 50). The division process is carried out based on normalized student scores. Percentile rank normalization is used for normalization (AKCAPINAR & BAYAZIT, 2019, p. 410). we used **Logistic regression**, **Random Forest**, and **Support Vector Machine** methods to predict the students who are low performers and the high performers' students based on their features. Then we compare the accuracy of the methods and recommend the method with the highest accuracy. Since all the initial variables are numeric, we used **KNN regression** to be suitable for the numeric dataset. **KNN regression** was applied for the purpose of predicting the grades of the students.

Analysis

Visualizations

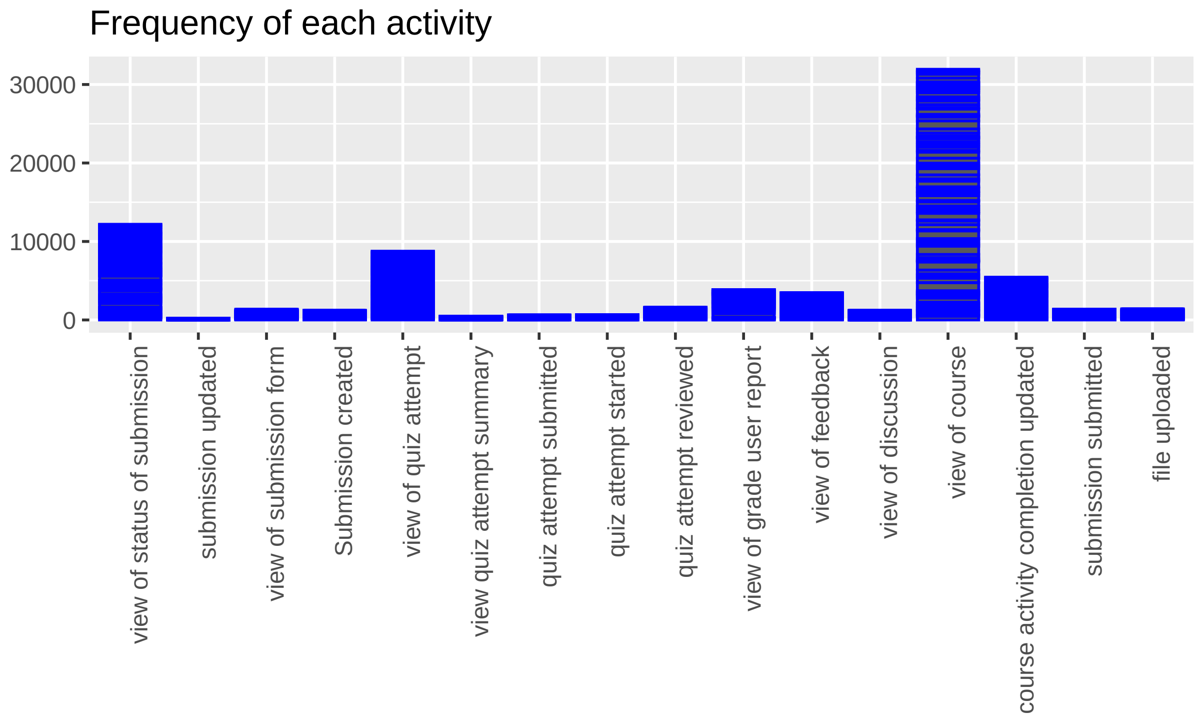
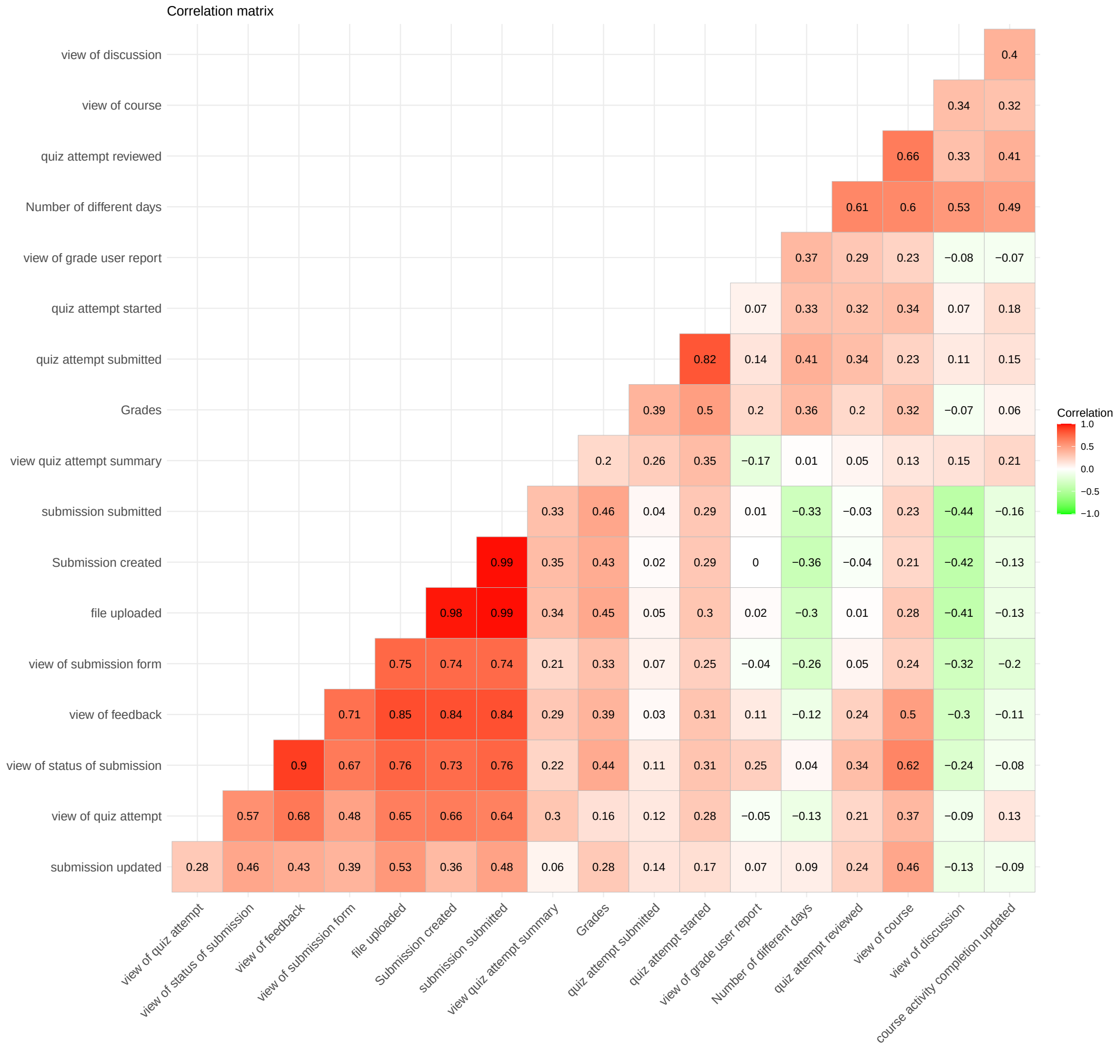


Figure 1: Bar plot

According to the bar plot, the three most frequent activities were the view of course, the view of status of the submission, and the view of the quiz attempt.

Correlation coefficient is a measure of the strength of the relationship between two variables. We create a correlation matrix to calculate and visualize correlation among grades, the various activities, and the number of different days that the student logged on Moodle.



	view of quiz attempt	view of status of submission	view of feedback	view of submission form	file uploaded	Submission created	submission submitted	view quiz attempt summary	Grades	quiz attempt submitted	quiz attempt started	view of grade user report	Number of different days	quiz attempt reviewed	view of course	view of discussion	course activity completion updated
view of discussion																	0.4
view of course																0.34	0.32
quiz attempt reviewed															0.66	0.33	0.41
Number of different days														0.61	0.6	0.53	0.49
view of grade user report													0.37	0.29	0.23	−0.08	−0.07
quiz attempt started												0.07	0.33	0.32	0.34	0.07	0.18
quiz attempt submitted											0.82	0.14	0.41	0.34	0.23	0.11	0.15
Grades										0.39	0.5	0.2	0.36	0.2	0.32	−0.07	0.06
view quiz attempt summary									0.2	0.26	0.35	−0.17	0.01	0.05	0.13	0.15	0.21
submission submitted								0.33	0.46	0.04	0.29	0.01	−0.33	−0.03	0.23	−0.44	−0.16
Submission created							0.99	0.35	0.43	0.02	0.29	0	−0.36	−0.04	0.21	−0.42	−0.13
file uploaded						0.98	0.99	0.34	0.45	0.05	0.3	0.02	−0.3	0.01	0.28	−0.41	−0.13
view of submission form					0.75	0.74	0.74	0.21	0.33	0.07	0.25	−0.04	−0.26	0.05	0.24	−0.32	−0.2
view of feedback				0.71	0.85	0.84	0.84	0.29	0.39	0.03	0.31	0.11	−0.12	0.24	0.5	−0.3	−0.11
view of status of submission			0.9	0.67	0.76	0.73	0.76	0.22	0.44	0.11	0.31	0.25	0.04	0.34	0.62	−0.24	−0.08
view of quiz attempt		0.57	0.68	0.48	0.65	0.66	0.64	0.3	0.16	0.12	0.28	−0.05	−0.13	0.21	0.37	−0.09	0.13
submission updated	0.28	0.46	0.43	0.39	0.53	0.36	0.48	0.06	0.28	0.14	0.17	0.07	0.09	0.24	0.46	−0.13	−0.09

Figure 2: Correlation matrix

As we can see from the plot, the correlation coefficient between grade and any other variable is not strong, as the correlation coefficients are less than 0.5. We use the **ggpairs** function in order to demonstrate the distribution of grades, number of times that submission is updated, number of times that a quiz attempt is viewed, number of times that status of submission is viewed, number of times that feedback is viewed, number of times that a submission form is viewed, number of times that a file is uploaded, number of times that submission is created, number of times that submission is submitted, and number of times that a quiz attempt summary is viewed. We choose these variables as there is a correlation between them and students' grades.

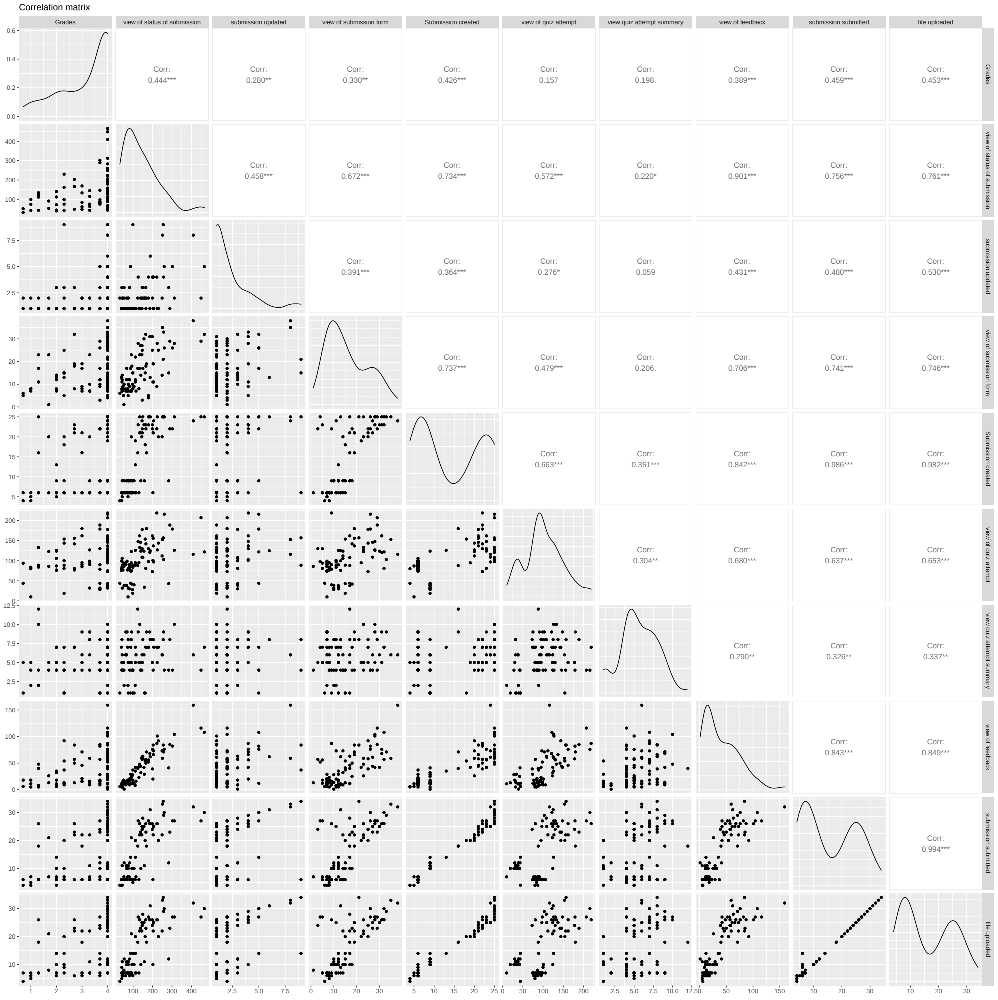


Figure 3: Correlation matrix

According to the **Correlation matrix** there is no linear relationship among students' grades and number of times that submission is updated, number of times that a quiz attempt is viewed, number of times that status of submission is viewed, number of times that feedback is viewed, number of times that a submission form is viewed, number of times that a file is uploaded, number of times that submission is created, number of times that submission is submitted, and number of times that a quiz attempt summary is viewed.

K-Means Algorithm

K-Means algorithm is an unsupervised cluster algorithm used to minimize cluster performance index, square error, and error criterion(Li, Wu, 2012). In the K-means algorithm, we take into consideration **Total sum of squares**, **Between groups sum of squares**, **Within group sum of squares**. **k** is the number of pre-defined clusters. In order to find the optimal value of **k**, we use the **Elbow method**. We use the **summary** function for figuring out whether there is a need for normalization.

```
##      nuDays       status_of_subm_viewed submission_updated sub_form_view
##  Min.   : 43.00   Min.   : 32.0         Min.   :1.000      Min.   : 1.00
##  1st Qu.: 66.25   1st Qu.: 75.0         1st Qu.:1.000      1st Qu.: 8.00
##  Median : 76.00   Median :119.5         Median :2.000      Median :13.00
##  Mean   : 77.21   Mean   :141.7         Mean   :2.256      Mean   :15.78
##  3rd Qu.: 87.75   3rd Qu.:187.5         3rd Qu.:3.000      3rd Qu.:23.00
##  Max.   :118.00   Max.   :466.0         Max.   :9.000      Max.   :38.00
##  submission_created quiz_attempt_view quiz_attempt_sum_view quiz_attempt_sub
##  Min.   : 4.00      Min.   : 10.0     Min.   : 1.000        Min.   : 2.000
##  1st Qu.: 6.00      1st Qu.: 77.5     1st Qu.: 4.000        1st Qu.: 7.000
##  Median : 9.00      Median : 96.0     Median : 5.000        Median : 8.000
##  Mean   :14.19      Mean   :101.9     Mean   : 5.616        Mean   : 7.372
##  3rd Qu.:22.00      3rd Qu.:127.0     3rd Qu.: 7.000        3rd Qu.: 8.000
##  Max.   :25.00      Max.   :219.0     Max.   :12.000        Max.   :10.000
##  quiz_attempt_star quiz_attempt_rev grade_user_rep_viewed feedback_viewed
##  Min.   : 2.000    Min.   : 1.00    Min.   :  1.00        Min.   :  1.00
##  1st Qu.: 8.000    1st Qu.: 9.25    1st Qu.:  9.25        1st Qu.: 13.00
##  Median : 8.000    Median :14.00    Median : 21.50        Median : 30.50
##  Mean   : 7.733    Mean   :18.69    Mean   : 44.88        Mean   : 40.27
##  3rd Qu.: 8.000    3rd Qu.:24.00    3rd Qu.: 49.75        3rd Qu.: 60.50
##  Max.   :11.000    Max.   :66.00    Max.   :445.00        Max.   :159.00
##   disc_viewed      course_view     course_act_comp_updat submission_submitted
##  Min.   :  1.00   Min.   :  99.0   Min.   :  5.00        Min.   : 4.00
##  1st Qu.:  3.00   1st Qu.: 189.2   1st Qu.: 24.00        1st Qu.: 6.00
##  Median :  4.50   Median : 304.5   Median : 43.00        Median :12.00
##  Mean   : 14.26   Mean   : 371.4   Mean   : 63.33        Mean   :15.91
##  3rd Qu.: 15.75   3rd Qu.: 479.2   3rd Qu.: 80.00        3rd Qu.:25.00
##  Max.   :102.00   Max.   :1055.0   Max.   :261.00        Max.   :34.00
##  file_uploaded
##  Min.   : 4.0
##  1st Qu.: 7.0
##  Median :13.0
##  Mean   :16.3
##  3rd Qu.:25.0
##  Max.   :34.0
```

The range of the variables varies on a large scale. Hence, we normalize the range of the variables by using **Z-score normalization**. **Z-score** represents the distance of the point from the mean measured in standard deviation. The **mean** is equal to zero, and **standard deviation** is equal to one. The normalization is done in order to make the algorithm independent from the random variable unit.

We use **fviz_nbclust()** function from **factoextra** package in order to create **Elbow curve**.

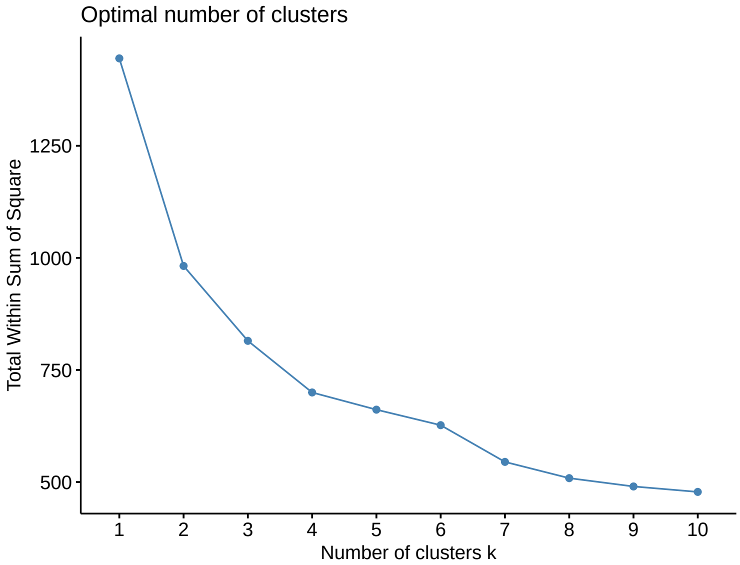


Figure 4: Elbow curve

According to the graph, we can state that by increasing the number of k, the total within sum of squares decreases. However, we need to define the optimal number of k for the k-means algorithm. This is the number for which the decrease in **WSS** will be tiny as we increase the number of k. Therefore, in this case, the optimal value for **k** is **three**.

The evaluation of the model is done using **internal measures** such as **Silhouette coefficient**, **Dunn index**, **Connectivity**.

```
##
## Clustering Methods:
##  kmeans
##
## Cluster sizes:
##  3
##
## Validation Measures:
##                            3
##
## kmeans Connectivity  23.8933
##        Dunn           0.1491
##        Silhouette     0.2417
##
## Optimal Scores:
##
##              Score   Method Clusters
## Connectivity 23.8933 kmeans 3
## Dunn          0.1491 kmeans 3
## Silhouette    0.2417 kmeans 3
```

The **Silhouette coefficient** is **0.2417**, and we can conclude that the clustering is not so good. The **Dunn index** is **0.1491**. This means that **min.separation** is lower and **max.diameter** is higher. As the **min.separation** is lower, between cluster distance is lower. As the **max. diameter** is higher, the within cluster distances are higher. **Connectivity** is **23.8933**, which means that not all the nearest neighbors are in the same cluster. So, the clustering is not so good. We also check what percent of the total variance in the data can be explained by the clusters dividing Between Groups Sum of Squares by Total Sum of Squares.

43.6 percent of the total variance in the data can be explained by the clusters dividing Between Groups Sum of Squares by Total Sum of Squares.

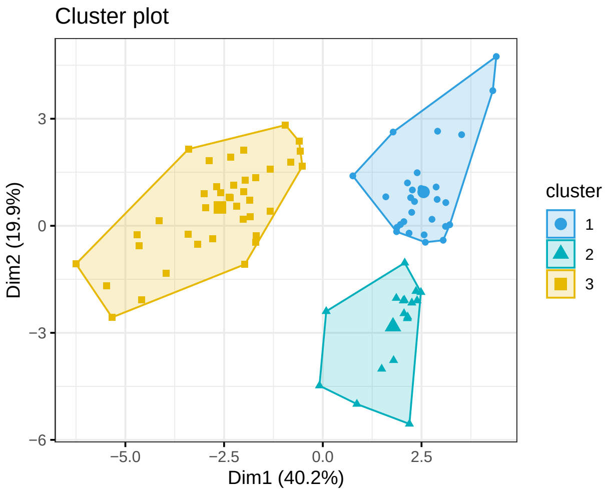


Figure 5: Cluster plot

As we can see in cluster number 1, the spread of the points is less than the spread of cluster number 2 and 3. This means that within group sum of squares for cluster number one is less than for cluster number 2 and 3.

Random forest

RF is an ensemble learning method used for classification and regression. Developed by Breiman (2001), the method combines Breiman's bagging sampling approach (1996a) and the random selection of features, introduced independently by Ho (1995; 1998) and Amit and Geman (1997), in order to construct a collection of decision trees with a controlled variation. Using bagging, each decision tree in the ensemble is constructed using a sample with replacement from the training data (Fawagreh et al., 2014, p. 604). We used the **Random forest** algorithm to predict the students who are low performers and the high performers. In the model, our dependent variable is grade, and independent variables are the number of times that a status of the submission is viewed, number of times that submission is created, number of times a quiz is submitted, number of times a grade user report is viewed, number of times a course is viewed, number of times a file is uploaded, number of times that submission is updated, number of times that a quiz attempt is viewed, number of times that a quiz attempt is started, number of times that feedback is viewed, number of times that a course activity completion is updated, number of different days that a student logs in to Moodle, number of times that a submission form is viewed, number of times that a quiz attempt summary is viewed, number of times that a quiz attempt is reviewed, number of times that a discussion is viewed, and number of times that a submission has been submitted. We divided our data set into training and testing sets by using 80/20 proportion. We did ten-fold cross-validation.

```
## Random Forest
##
## 86 samples
## 17 predictors
##  2 classes: 'high_performer', 'low_performer'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 78, 78, 77, 77, 78, 77, ...
## Resampling results across tuning parameters:
##
##   mtry  ROC        Sens   Spec
##    2    0.8745833  0.880  0.7583333
##    9    0.8683333  0.855  0.7333333
##   17    0.8475000  0.855  0.6750000
##
## ROC was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 2.
```

Mtry shows how many variables are selected at random out of the set of independent variables to split the given node. The value of mtry is holding constant while growing the forest. In other words, the mtry hyperparameter will be the same for the whole process of generating the random forest. For every split in every tree, different random subsets of variables are used.

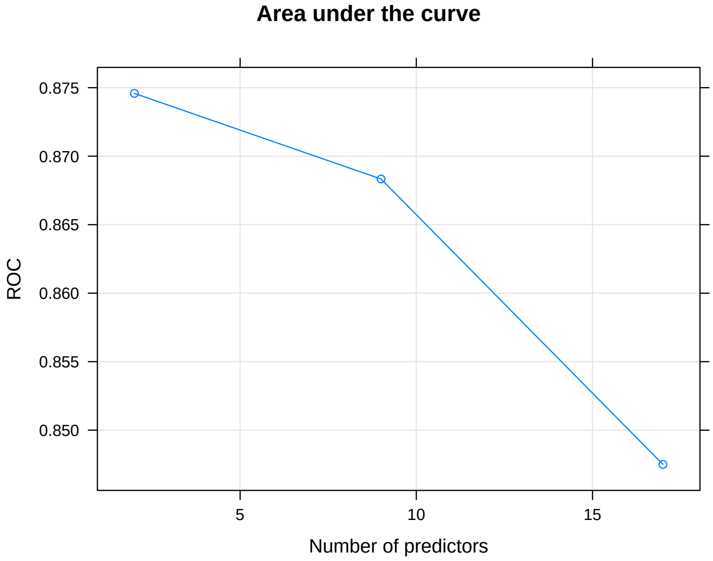


Figure 6: Area under the curve

According to the graph, we can state that the optimal value of mtry is equal to two.

We train a Random Forest model on the training data with 50 trees using the *randomForest* package.

The **Out of Bag score** is **24.29** percent. This means that the model has ***75.71** percent out of sample accuracy for the training set.

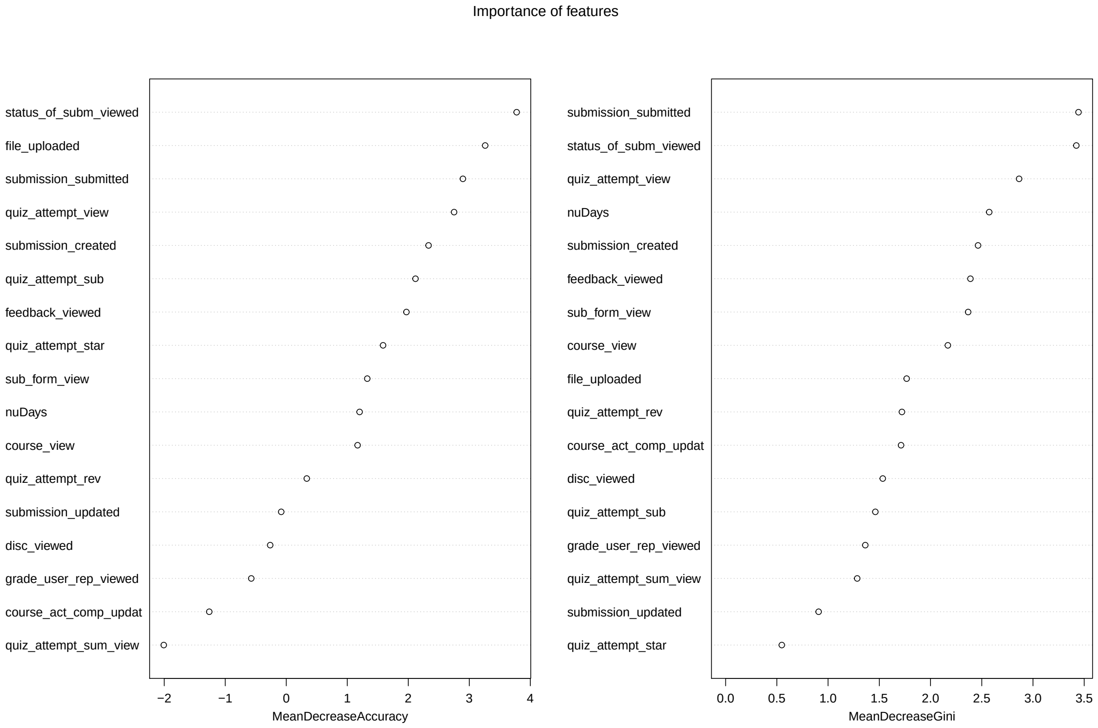


Figure 7: Importance of features

The **Mean Decrease Accuracy** plot expresses how much accuracy of the model losses by excluding each variable. The more the accuracy suffers, the more important the variable is for the successful classification. The higher the variable scores here, the more important it is for the model. The top 3 predictor variables for Mean Decrease Accuracy are the number of views of the status of a submission, the number of times the file is uploaded, and the number of times the submission is submitted. The **Mean Decrease in Gini coefficient** is a measure of how much each variable contributes to the homogeneity of the nodes and leaves in the resulting random forest. The higher the value of the mean decrease Gini score, the higher the importance of the variable in the model. The top 3 predictor variables for Mean Decrease in Gini coefficient are the number of times the submission is submitted, the number of views of the status of submission, and the number of views of the quiz attempt.

For evaluation of the model performance, we created a **Receiver Operator Characteristic(ROC) curve**.

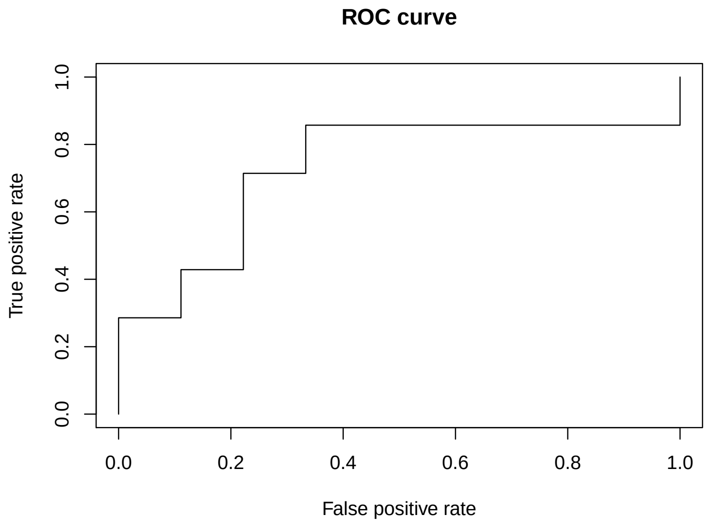


Figure 8: ROC curve

The curve is close to the **top-left corner** which indicates a better performance.

The area under the curve is a single number between 0 and 1. The closer it is to 1, the better is going to be the model. In this case, AUC is 0.7301587, which means that there is a 73.02% chance that the model will distinguish between classes.

```
## Confusion Matrix and Statistics
##
##                 Reference
## Prediction        high_performer low_performer
##   high_performer              7             3
##   low_performer               2             4
##
##                Accuracy : 0.6875
##                  95% CI : (0.4134, 0.8898)
##     No Information Rate : 0.5625
##     P-Value [Acc > NIR] : 0.2269
##
##                   Kappa : 0.3548
##
##  Mcnemar's Test P-Value : 1.0000
##
##             Sensitivity : 0.5714
##             Specificity : 0.7778
##          Pos Pred Value : 0.6667
##          Neg Pred Value : 0.7000
##              Prevalence : 0.4375
##          Detection Rate : 0.2500
##    Detection Prevalence : 0.3750
##       Balanced Accuracy : 0.6746
##
##        'Positive' Class : low_performer
##
```

The **accuracy** of the model is equal to 0.6875. It shows that 68.75 percent of the predictions of the model are correct. This means that this model is quite a good model. The **sensitivity** of the model shows what percentage of actual low performer students are actually predicted to be low performers. It is equal to 0.5714, and it means that 57.14 percent of actual low performer students are actually predicted to be low performers. The **specificity** of the model shows what percentage of actual high performer students are actually predicted to be high performers. It is equal to 0.7778, and it means that 77.78 percent of actual high performer students are actually predicted to be high performers. **Positive Predictive Value** shows how many of the predicted low performer students are actually low performers. It shows what percentage of the predicted low performer students are actually low performers. It is equal to 0.6667, and it means that 66.67 percent of the predicted low performer students are actually low performers. **Negative Predictive Value** shows what percentage of the predicted high performer students are actually high performers. It is equal to 0.7000, which means that 70 percent of the predicted high performer students are actually high performers.

Logistic regression

Generally, logistic regression is well suited for describing and testing hypotheses about relationships between a categorical outcome variable and one or more categorical or continuous predictor variables (Peng, 2002, p.4). The model predicts the probability that the given case will belong to one of the classes, and based on that probability, the class label is assigned to the given case. In our project, students are divided into two groups, low performers(Score<=50) and high performers (Score > 50). The division process is carried out based on normalized student scores. Percentile rank normalization is used for normalization (AKCAPINAR & BAYAZIT, 2019, p. 410). We use **Logistic regression** analysis to predict the students who are low performers and the high performers. In the model, our dependent variable is grade, and independent variables are the number of times that a status of the submission is viewed, number of times that submission is created, number of times a quiz is submitted, number of times a grade user report is viewed, number of times a course is viewed, number of times a file is uploaded, number of times that submission is updated, number of times that a quiz attempt is viewed, number of times that a quiz attempt is started, number of times that feedback is viewed, number of times that a course activity completion is updated, number of different days that a student logs in to Moodle, number of times that a submission form is viewed, number of times that a quiz attempt summary is viewed, number of times that a quiz attempt is reviewed, number of times that a discussion is viewed, and number of times that a submission has been submitted. We divide our data set into training and testing sets by using 80/20 proportion.

```
##
## Call:
## glm(formula = Grades ~ `view of status of submission` + `Submission created` +
##     `quiz attempt submitted` + `view of grade user report` +
##     `view of course` + `file uploaded` + `submission updated` +
##     `view of quiz attempt` + `quiz attempt started` + `view of feedback` +
##     `course activity completion updated` + `Number of different days` +
##     `view of submission form` + `view quiz attempt summary` +
##     `quiz attempt reviewed` + `view of discussion` + `submission submitted`,
##     family = "binomial", data = log_train, method = "brglm_fit")
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.56737  -0.58019  -0.04224   0.38781   1.41729
##
## Coefficients:
##                                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)                          24.886748   9.653413   2.578  0.00994 **
## `view of status of submission`       -0.020852   0.016155  -1.291  0.19680
## `Submission created`                  0.403785   1.652952   0.244  0.80701
## `quiz attempt submitted`             -0.964435   0.697314  -1.383  0.16664
## `view of grade user report`           0.008084   0.011851   0.682  0.49518
## `view of course`                      0.005265   0.004930   1.068  0.28552
## `file uploaded`                      -0.677977   1.759873  -0.385  0.70006
## `submission updated`                  0.602446   1.616864   0.373  0.70945
## `view of quiz attempt`                0.024915   0.013837   1.801  0.07177 .
## `quiz attempt started`               -0.326609   1.067360  -0.306  0.75961
## `view of feedback`                    0.061821   0.046320   1.335  0.18199
## `course activity completion updated`  0.011330   0.009598   1.180  0.23781
## `Number of different days`           -0.205779   0.072042  -2.856  0.00428 **
## `view of submission form`             0.038557   0.084901   0.454  0.64973
## `view quiz attempt summary`           0.305245   0.223628   1.365  0.17226
## `quiz attempt reviewed`               0.014461   0.041829   0.346  0.72955
## `view of discussion`                  0.018290   0.027799   0.658  0.51058
## `submission submitted`               -0.234277   0.489719  -0.478  0.63237
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 96.125  on 69  degrees of freedom
## Residual deviance: 28.791  on 52  degrees of freedom
## AIC: 64.791
##
## Number of Fisher Scoring iterations: 37
```

According to the model results, it can be stated that the number of different days that students log in Moodle is statistically significant as the p-value is less than alpha (0.05). The number of different days is significantly and negatively associated with the grade. By adding the number of different days that students log in Moodle by one day, the log odds of high performers will decrease by -0.205779, holding other variables constant. The odds ratio for the number of different days that students log in to Moodle is 0.814013. This means that if we increase the number of different days that students log in Moodle by one day, the odds of being a high performer will decrease by 18.6 percent, holding other variables constant. We assess the **model's predictive power** based on a **Confusion Matrix**. For cutting the predicted probabilities, we define the **threshold** as

0.5 to get the class assignments. This means we classify every student with a predicted probability from the model greater than 0.5 as a **low performer**.

```
## Confusion Matrix and Statistics
##
##                 Reference
## Prediction       low_performer high_performer
##   low_performer              2              9
##   high_performer             5              0
##
##                Accuracy : 0.125
##                  95% CI : (0.0155, 0.3835)
##     No Information Rate : 0.5625
##     P-Value [Acc > NIR] : 1.0000
##
##                   Kappa : -0.6716
##
##  Mcnemar's Test P-Value : 0.4227
##
##             Sensitivity : 0.2857
##             Specificity : 0.0000
##          Pos Pred Value : 0.1818
##          Neg Pred Value : 0.0000
##              Prevalence : 0.4375
##          Detection Rate : 0.1250
##    Detection Prevalence : 0.6875
##       Balanced Accuracy : 0.1429
##
##        'Positive' Class : low_performer
##
```

The **accuracy** of the model is 0.125. It shows that 12.5 percent of the predictions of the model are correct. This means that this model is worse than random guessing. This means that there is no linear relationship between the dependent variable(grade) and independent variables. The **sensitivity** of the model shows what percentage of actual low performer students are actually predicted to be low performers. It is equal to 0.2857, and it means that 28.57 percent of actual low performer students are actually predicted to be low performers. The **specificity** of the model shows what percentage of actual high performer students are actually predicted to be high performers. It is equal to 0.0000, and it means that 0 percent of actual high performer students are actually predicted to be high performers. **Positive Predictive Value** shows how many of the predicted low performer students are actually low performers. It shows what percentage of the predicted low performer students are actually low performers. It is equal to 0.1818, and it means that 18.18 percent of the predicted low performer students are actually low performers. **Negative Predictive Value** shows what percentage of the predicted high performer students are actually high performers. It is equal to 0.0000, which means that 0 percent of the predicted high performer students are actually high performers.

We create a **Receiver Operator Characteristic(ROC) curve** to represent the classifier's diagnostic ability.

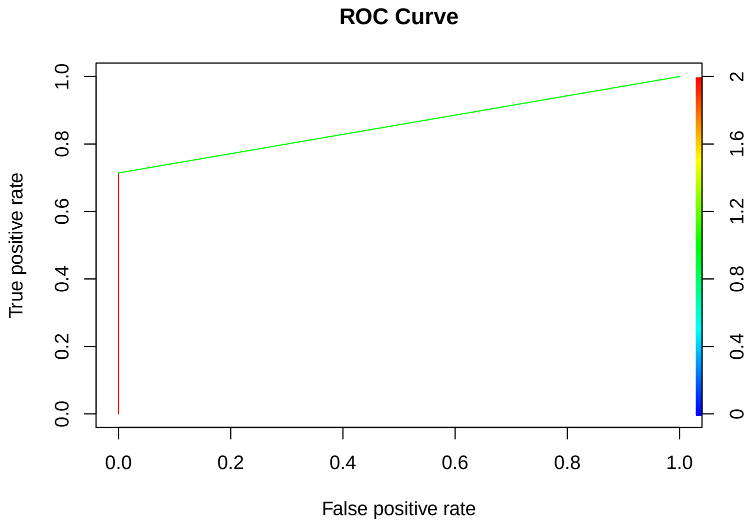


Figure 9: ROC curve

The curve is close to the **top-left corner** which indicates a better performance.

The area under the curve is a single number between 0 and 1. The closer it is to 1, the better is going to be the model. In this case, AUC is 0.8571, which means an 85.71% chance that the model will distinguish between classes.

Support Vector Machine

Support vector machine is a supervised machine learning classification algorithm that classifies data based on its features. It uses various dividing **hyperplanes** in order to divide the data into different classes. **Support vectors** are the points, which are very close to the **hyperplane**. Using the support vectors, we can select the best line to divide the data. **Margin** is the distance between the support vectors and the **hyperplane**. The best line has the greatest margin distance between the support vectors. **D+** is the shortest distance to the closest positive point. **D-** is the shortest distance to the closest negative point. The sum of **D+** and **D-** is called the **distance margin**. If the margin between the support vectors is not the maximum, then the data can be **misclassified**. Let define **R** as the number of dimensions of the data. **SVM** uses the kernel to convert $\mathbf{R}^{\mathbf{2}}$ dimension to $\mathbf{R}^{\mathbf{3}}$ dimension.

```
## Support Vector Machines with Linear Kernel
##
## 70 samples
## 17 predictors
##  2 classes: 'high_performer', 'low_performer'
##
## Pre-processing: centered (17), scaled (17)
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 63, 63, 63, 62, 63, 63, ...
## Resampling results across tuning parameters:
##
##   C           Accuracy   Kappa
##   0.00000000        NaN        NaN
##   0.05263158  0.7190476  0.4180956
##   0.10526316  0.7601190  0.5121641
##   0.15789474  0.8029762  0.6063367
##   0.21052632  0.8154762  0.6358239
##   0.26315789  0.8154762  0.6358239
##   0.31578947  0.8011905  0.6079253
##   0.36842105  0.8154762  0.6382586
##   0.42105263  0.8279762  0.6632586
##   0.47368421  0.8154762  0.6382586
##   0.52631579  0.8297619  0.6641048
##   0.57894737  0.8297619  0.6641048
##   0.63157895  0.8029762  0.6059032
##   0.68421053  0.8029762  0.6059032
##   0.73684211  0.8029762  0.5953857
##   0.78947368  0.7863095  0.5620524
##   0.84210526  0.7863095  0.5620524
##   0.89473684  0.7863095  0.5620524
##   0.94736842  0.7863095  0.5620524
##   1.00000000  0.7863095  0.5620524
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was C = 0.5263158.
```

The cost function controls training errors and margins. If the value of parameter **c** is **large**, then the **margin** will be **narrow**, allowing less misclassifications. If the value of **c** is **small** then the **margin** will be **large**, allowing more misclassifications.

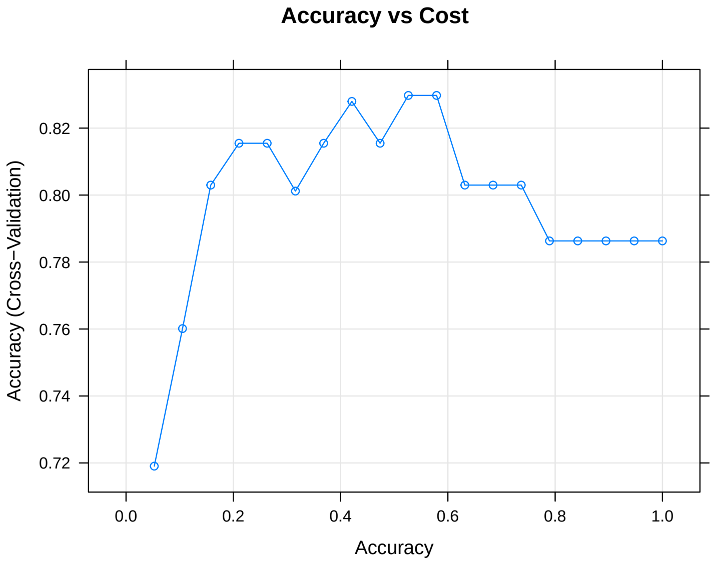


Figure 10: Accuracy vs Cost

The **optimal value** of **c** is equal to 0.5263158.

We evaluated the model performance by creating a **confusion matrix**.

```
## Confusion Matrix and Statistics
##
##
## prediction       high_performer low_performer
##   high_performer              7             2
##   low_performer               2             5
##
##                Accuracy : 0.75
##                  95% CI : (0.4762, 0.9273)
##     No Information Rate : 0.5625
##     P-Value [Acc > NIR] : 0.102
##
##                   Kappa : 0.4921
##
##  Mcnemar's Test P-Value : 1.000
##
##             Sensitivity : 0.7143
##             Specificity : 0.7778
##          Pos Pred Value : 0.7143
##          Neg Pred Value : 0.7778
##              Prevalence : 0.4375
##          Detection Rate : 0.3125
##    Detection Prevalence : 0.4375
##       Balanced Accuracy : 0.7460
##
##        'Positive' Class : low_performer
##
```

The **accuracy** of the model is equal to 0.75. It shows that 75 percent of the predictions of the model are correct. This means that this model is quite a good model. The **sensitivity** of the model shows what percentage of actual low performer students are actually predicted to be low performers. It is equal to 0.7143, and it means that 71.43 percent of actual low performer students are actually predicted to be low performers. The **specificity** of the model shows what percentage of actual high performer students are actually predicted to be high performers. It is equal to 0.7778, and it means that 77.78 percent of actual high performer students are actually predicted to be high performers. **Positive Predictive Value** shows how many of the predicted low performer students are actually low performers. It shows what percentage of the predicted low performer students are actually low performers. It is equal to 0.7143, and it means that 71.43 percent of the predicted low performer students are actually low performers. **Negative Predictive Value** shows what percentage of the predicted high performer students are actually high performers. It is equal to 0.7778, which means that 77.78 percent of the predicted high performer students are actually high performers.

KNN regression

KNN regression is a nonparametric regression technique. Nonparametric regression is an alternative approach to model complex interactions by deriving the functional form of models from the data itself (Goyal et al., 2013, p. 16). We used the number of times that a status of the submission is viewed, number of times that submission is created, number of times a quiz is submitted, number of times a grade user report is viewed, number of times a course is viewed, number of times a file is uploaded, number of times that submission is updated, number of times that a quiz attempt is viewed, number of times that a quiz attempt is started, number of times that feedback is viewed, number of times that a course activity completion is updated, number of different days that a student logs in to Moodle, number of times that a submission form is viewed, number of times that a quiz attempt summary is viewed, number of times that a quiz attempt is reviewed, number of times that a discussion is viewed, and number of times that a submission has been submitted variables to predict the grade by using kNN regression. The range of the variables varies on a large scale. Hence, we normalize the range of the variables by using **Z-score normalization**. **Z-score** represents the distance of the point from the mean measured in standard deviation. The **mean** is equal to zero, and **standard deviation** is equal to one. The normalization is done in order to make the algorithm independent from the random variable unit.

We divided our data set into **training** and **testing** sets with an 80-20 ratio.

For choosing the value of **k** in **KNN** algorithm for use **k= sqrt(n)** approach. Here, **n** is the number of observations in the **training** set.

The root mean squared error is equal to 0.70. The closer the root mean squared error is to zero, the more accurate the model is. Hence, we can conclude that the model is accurate.

Conclusion and Recommendations

In the scope of our project, we have done visualizations and found out that the three most frequent activities performed by students are view of a course, view of the status of a submission, and view of quiz attempt. The information obtained by these analyses will contribute to the researchers and instructors for clustering students by their engagement levels identifying low performers. The correlation matrix has been created, and it shows that the highest correlation coefficient exists between grade and submission submitted activity. However, no strong correlation exists.

In this work, we have shown how useful the application of data mining techniques in course management systems can be for online instructors. Hence, there is no linear relationship between them. A _single**K-means** algorithm was used for clustering, while three algorithms were used for classification. For **cluster analysis**, we used the K-means algorithm and found out that **43.6** percent of the total variance in the data can be explained by the clusters dividing Between Groups Sum of Squares by Total Sum of Squares. For **classification analysis**, we used **Logistic regression**, **Random Forest**, and **Support Vector Machine** methods to predict the students who are low performers and the high performers' students based on their features. The accuracy of **Logistic regression** is 0.125, of **Random Forest** is 0.6875, and of **Support Vector Machine** is 0.75. In our project, we have also used **KNN regression** and_ the root mean squared error** of the algorithm is equal to 0.70. The closer the root mean squared error is to zero, the more accurate the model is. Hence, we can conclude that the model is accurate. After doing all these analyses, we come up to the conclusion that **Support Vector Machine** algorithm suits to our data most since the accuracy of this algorithm is the highest.

References

AKCAPINAR, BAYAZIT. (2019). MoodleMiner: Data Mining Analysis Tool for Moodle Learning Management System (Vol. doi: 10.17051/ilkonline.2019.527645). Yeditepe University, Department of Computer Education & Instructional Technology.

Baker, R. (April 2012). Learning analytics and educational data mining: Towards communication and collaboration (Vol. DOI:10.1145/2330601.2330661). Pennsylvania : University of Pennsylvania .

Cristobal Romero * , Sebastian Ventura, Enrique Garcia . (2007). Data mining in course management systems: Moodle case study and tutorial . Spain: a Department of Computer Sciences and Numerical Analisys, University of Cordoba.

Dutt A, Ismail MA, Herawan T. A Systematic Review on Educational Data Mining. Vol. 5, IEEE Access. 2017. p. 15991?6005.

Hui-Chun Hung 1 , I-Fan Liu 2 , Che-Tien Liang 3 and Yu-Sheng Su 4. (28 January 2020). Applying Educational Data Mining to Explore Students? Learning Patterns in the Flipped Learning Approach for Coding Education. Taipei City 110: Graduate Institute of Network Learning Technolog.

Omar R, Md Tap AO, Abdullah ZS. Web usage mining: A review of recent works. 2014 5th Int Conf Inf Commun Technol Muslim World, ICT4M 2014. 2014.

Peng, J. (September 2002). An Introduction to Logistic Regression Analysis and Reporting (Vol. DOI: 10.1080/00220670209598786). (P. A. regression, Ed.) The Journal of Educational Research 96(1):3-14.

Rinkaj Goyala, * , Pravin Chandraa , Yogesh Singha. (2013). Suitability of KNN Regression in the Development of Interaction Based Software Fault Prediction Models. Delhi 110078: a University School of Information & Communication Technology, Guru Gobind Singh Indraprastha University. Timothy Frank1 and Lauren F.V. Scharff2. (October 2013). Learning contracts in undergraduate courses: Impacts on student behaviors and academic performance (Vols. Vol. 13, No. 4). Journal of the Scholarship of Teaching and Learning.

Romero C, Ventura S. Educational data mining: A review of the state of the art. Vol. 40, IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews. 2010. p. 601?18.

Youguo Li, Haiyan Wu . ((2012) 1104 ? 1109). A Clustering Method Based on K-Means Algorithm . 464000, China: Department of Computer Science Xinyang Agriculture College Xinyang, Henan.

Cristobal Romero * , Sebastian Ventura, Enrique Garcia . (2007). Data mining in course management systems: Moodle case study and tutorial . Spain: a Department of Computer Sciences and Numerical Analisys, University of Cordoba.

Hui-Chun Hung 1 , I-Fan Liu 2 , Che-Tien Liang 3 and Yu-Sheng Su 4. (28 January 2020). Applying Educational Data Mining to Explore Students? Learning Patterns in the Flipped Learning Approach for Coding Education. Taipei City 110: Graduate Institute of Network Learning Technolog.