Mini Project II – Students Performance

1) Introduction

This project's goal was to use supervised learning techniques to estimate students final marks in an online machine learning course. Grades from exams, quizzes, peer reviews, and course diaries were included in the data, which was collected from 107 enrolled students. In order to provide additional support and improve student performance, instructor must identify those students who are having difficulty in the course and those who are having trouble quitting.

The main indicators of student's achievement in a particular course are their grades and involvement in class activities, as stated in the project description.

We may use the grades and course activity data received from the students to train machine learning algorithms and forecast overall student performance. This will give the underperforming students a chance to perform better and assist in reducing the number of course dropouts. Additionally, it would aid teachers and student tutors to identify weak performing students and help them.

The methods, procedures, and outcomes of this predictive analysis are described in this study. Both the Random Forest Classifier and the Naive Bayes Classifier along with Sklearn models are used in this project to make predictions.
The projects tools includes: -

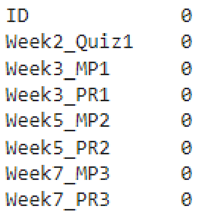
a) Numpy
b) Pandas
c) Matplotlib
d) Sklearn and so on

2) Data Overview

The "MP2_data.csv" CSV file has been made available to us. The available dataset contained anonymized information relating to 107 enrolled students. The data included students grades (from 3 mini projects, 3 quizzes, 3 peer reviews, and the final overall grade) as well as the course logs. The deadline for the three mini projects fell within weeks 3, 5, and 8 of the course, whereas the deadline for the quizzes fell within weeks 2, 4, and 8.
Status0: course / lectures / content related, Status1: assignment related, Status2: grade related and Status3: forum related.

3) Data Processing

We must preprocess the data before using it to train and make predictions from models. First, the dataset was verified for null values; none were present.
For e.g.;



```
ID              0
Week2_Quiz1     0
Week3_MP1       0
Week3_PR1       0
Week5_MP2       0
Week5_PR2       0
Week7_MP3       0
Week7_PR3       0
```

Figure 3.1 table with null values status

Every available column in the provided data did not contain any null values, as seen in the figure above. Then, the varied values were examined as well. Since "Week1_Stat1" was discovered to have a NaN value, this column was deleted. After checking for balanced dataset, due to imbalance of data, 4 column was created to sum up all other column into this based on their types.

4) Data Analysis

In order to train a good model, features must be identified that will provide the model with the most relevant data. Significant features should be discovered and the number of features used should be reduced in order to reduce the noise brought on by unneeded features and improve the performance of the model. The qualities with the strongest correlation to the expected values should be chosen. Additionally, the link between stats 0, 1, 2, and 3 and grade was examined.

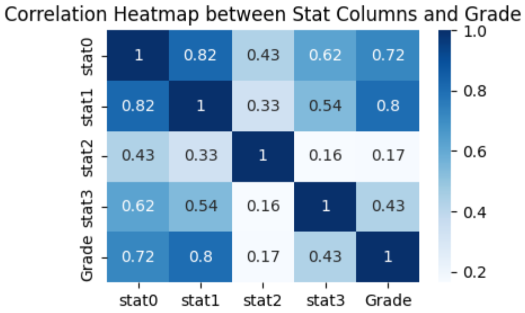


Figure 4.1 Correlation between stat column and Grade

It is clear from the above correlation matrix between Grade and Stat column that Stat0 (course related) and Stat1 (assignment related) have the strongest correlation (0.82), and that Stat3 and Stat2 have the lowest correlation (0.16).

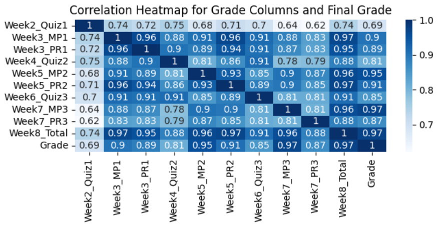


Figure 4.2 Correlation between grade column and final grade

It is evident from the following figure that the majority of the grade columns have a strong association to grades. The Week8_Total and Grade have the highest correlation (0.97) while the Week2_Quiz1 and Grade have the lowest correlation (0.69).

Both the Random Forest Classifier and the Naive Bayes Classifier were used to train the filtered features column. The Random Classifier's features importance function was used to obtain the final features for the training model with a threshold of (0.07). The final features chosen column are thus: - Week8_Total,Week7_MP3,Week5_MP2,Week3_MP1.

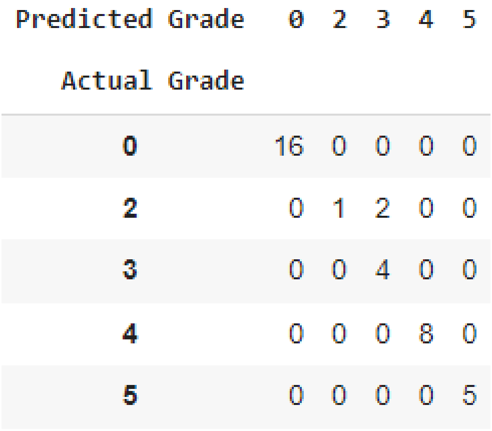
5) Model Training

I split the dataset into training and testing sets using the train_test_split function, utilizing 33% of the data for testing and the remaining data for training. According to the splitting, we get a total of 71 observations from 107 data for the training dataset and 36 observations for the testing dataset.

The Random Forest Classifier was used on the training dataset, and predictions were then made using the trained model. New predictions are made using the trained classifier on the test data (X_test[total_features]). Additionally computed and given is the estimated probability for the test data's first 15 observations. Using the training data and associated target values, a Gaussian Naive Bayes Classifier is trained. The trained classifier creates predictions based on the test data, and these predictions are then saved in 'clf_3_preds'. The probability ratings for the predictions are calculated using the predict_prob function, and they are then saved in the 'clf_3_prob' file. Following the length calculation, the prediction is reshaped and added to a dataframe with the column "Predictions."

6) Performance Evaluation

In the first approach (Random Forest Classifier), we create a confusion matrix by comparing the actual grades to the projected grades and using the crosstab() function.



Predicted Grade	0	2	3	4	5
Actual Grade					
0	16	0	0	0	0
2	0	1	2	0	0
3	0	0	4	0	0
4	0	0	0	8	0
5	0	0	0	0	5

Figure 6.1 Crosstab showing predicted and actual grades

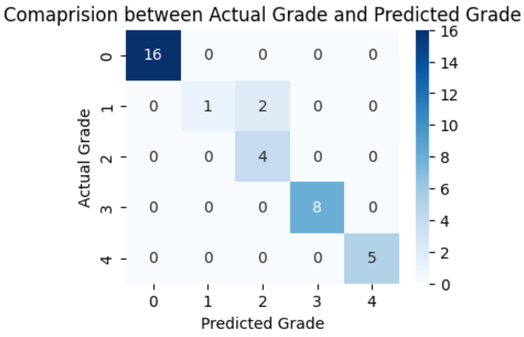
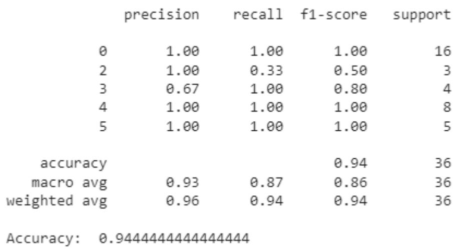


Figure 6.2 Comparision between Actual and Predicted Grade

In 16 cases when the actual grade was 0, the model accurately predicted it to be 0. In 0 cases, it mistakenly predicted it to be 0, and so on. The total number of right predictions (diagnoal elements) divided by the total number of forecasts yields the model's overall accuracy. The model's performance for each grade is broken out in depth in this confusion matrix, showing where it did well and where it predicted incorrectly. It is a useful tool for understanding the advantages and disadvantages of the classification model.

The accuracy score of the model is computed using accuracy score function.



```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        16
           2       1.00      0.33      0.50         3
           3       0.67      1.00      0.80         4
           4       1.00      1.00      1.00         8
           5       1.00      1.00      1.00         5

    accuracy                           0.94        36
   macro avg       0.93      0.87      0.86        36
weighted avg       0.96      0.94      0.94        36

Accuracy:  0.9444444444444444
```

Figure 6.3 Classification report on accuracy score

To assess the effectiveness of the

model, the classification report, which comprises precision, recall, F1-score, and support, is shown in the above figure. Using the training datasets and the random forest classifier, we were able to reach 94.4% accuracy in this case.

Predictions are transformed to a list and inserted as a new column with the name "Predictions" to the test DataFrame after the length of the predictions has been determined. Based on a certain column, the DataFrame is now combined with the original dataset and exported to a csv file called "rf_pred.csv".

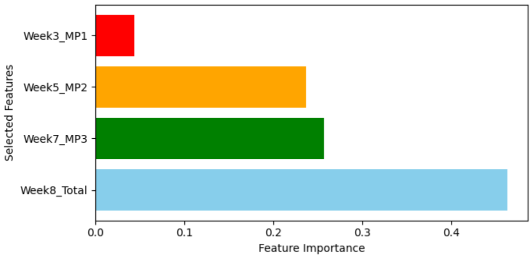
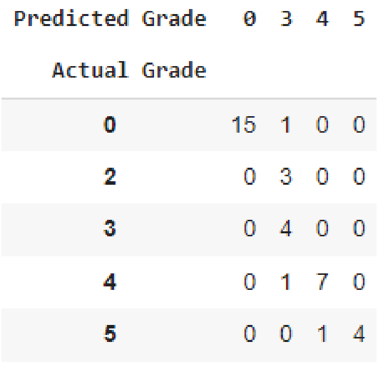


Figure 6.4 Graph showing selected features and feature Importance

The significance of a chosen feature in the machine learning model is shown in the horizontal bar chart using the "barch" function above. It provides a clear comparison of the importance of different features in the model, aiding in understanding which features have the most significant impact on the model's predictions.

The test features and predictions are concatenated column-wise in the Naive Bayes Classifier method. Based on particular columns, the generated DataFrame is combined with the initial dataset: Week 8 Total, Week 7 MP3, Week 5 MP2, and Week 3 MP1. The combined dataset's top 15 rows are presented to the console for visual review and stored as a csv file with the name "naive_prediction.csv."



Predicted Grade	0	3	4	5
Actual Grade				
0	15	1	0	0
2	0	3	0	0
3	0	4	0	0
4	0	1	7	0
5	0	0	1	4

Figure 6.5 Crosstab showing predicted and actual grade

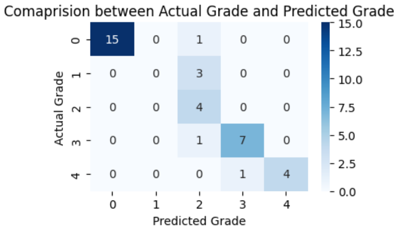


Figure 6.6 Comparision between Actual and Predicted Grade

We can evaluate the model's accuracy using our second technique, the Naive Bayes Classifier. Using the cross_cal_score function, the provided data is cross-validated. The mean accuracy found across all folds is stored in the 'naive_accuracy' variable. We have an accuracy rate of 88.9% using the Naive Bayes Classifier. According on the mean accuracy discovered using the cross-validation method, the Naive Bayes Classifier is predicted to perform well in the current environment.

7) Conclusion

When the model was implemented, there were certain bottlenecks. Unbalanced dataset, insufficient data, and data with many features were some of the problems identified.

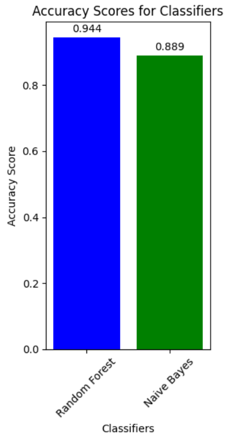


Figure 7.1 Bargraph showing comparison of accuracy

The accuracy ratings of the two classifiers are clearly displayed in the bar chart. Naive Bayes' accuracy is lower than Random Forest's. This comparison shows that Random Forest performs better than Naive Bayes in terms of accuracy on the tested dataset.

I was able to gain more knowledge about machine learning techniques, as well as training and testing facets, with this project. I learn about other available techniques while employing two different approaches.

The important features identified are: -

Week8_Total,

Week7_MP3,

Week5_MP2.

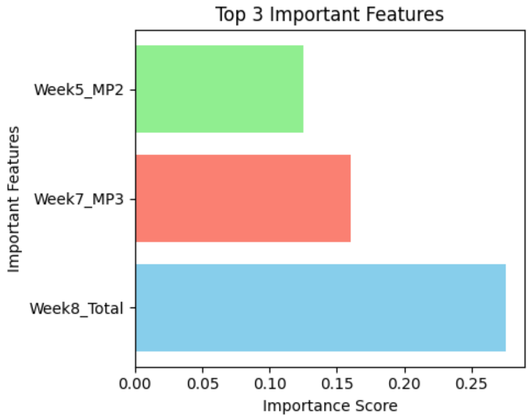


Figure 7.2 Bar graph showing three identified important features

From this, it can be inferred that mini projects and week 8_total have a significant impact on the student's final grade. As a result, it can be utilized as a gauge to help students' performance.