Report of mini project 2

Nazmus Sakib

Introduction

The aim of the project is to utilize two different supervised learning methods to make predictions about the final grades of students in an online course. Dataset was gathered from a fully online, nine-week machine learning course delivered through the Moodle online learning management system. The dataset comprised anonymized data pertaining to 107 enrolled students. This data included students' scores from 3 mini projects, 3 quizzes, 3 peer reviews, and their final overall grade, as well as their activity logs within the course. Below is the link to google colab:
https://colab.research.google.com/drive/1NZ-6PU4aDvggi0Acb-tyii--yWiC9Z7B?usp=sharing

Data processing

The dataset was clean and all the fields were numerical except the ID field but it was not relevant.
To select the relevant features I used **correlation matrix**.

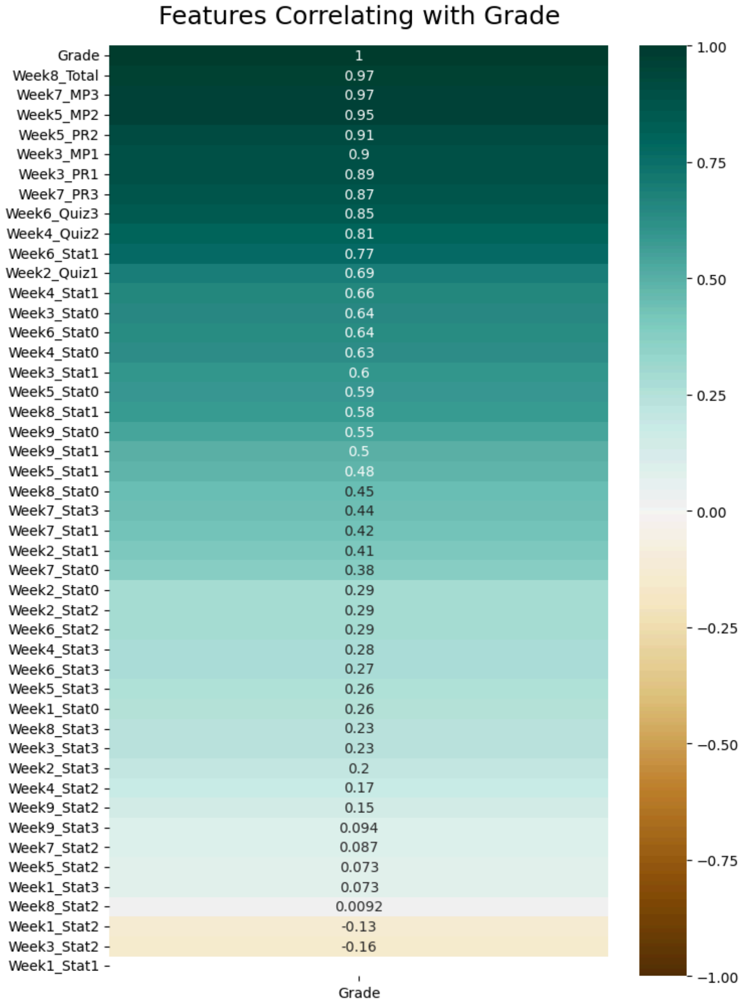

Fig: Feature correlation with Grade

The correlation coefficient has values between -1 to 1 where,

1. A value closer to 0 implies weaker correlation (exact 0 implying no correlation)
2. A value closer to 1 implies stronger positive correlation
3. A value closer to -1 implies stronger negative correlation

Here our target (dependent variable) is grade and from the above figure we find out strong and weak correlation with independent variable and set threshold. I chose the features that have a correlation coefficient higher than 0.7 and the pair plot below contains the ones that met my condition.

Data analysis

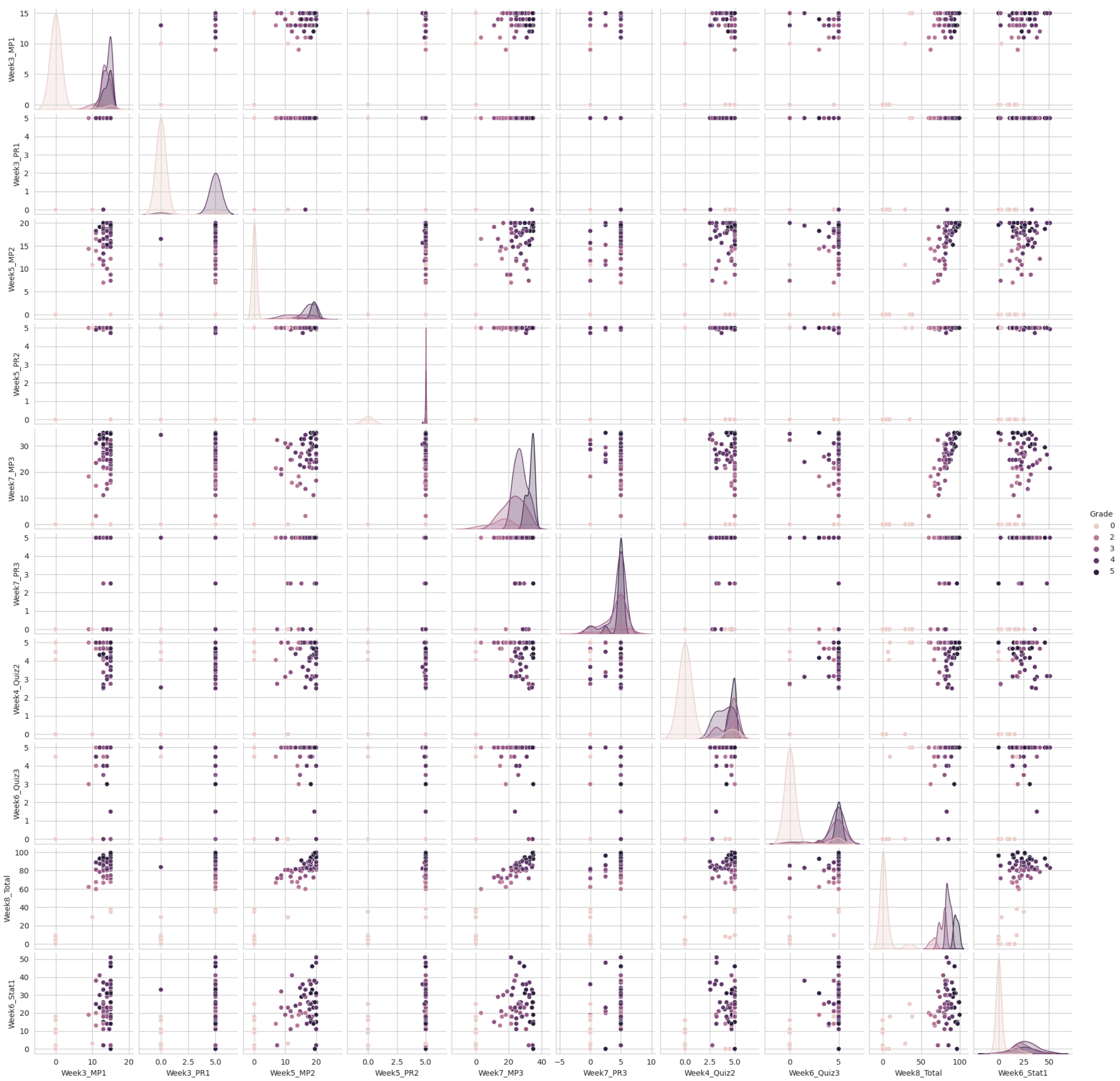

Fig: Pair plot of selected features

Among the multitude of methods available for Exploratory Data Analysis (EDA), a highly effective initial tool is the pairs plot, which is also known as a scatterplot matrix. This type of plot offers the advantage of simultaneously visualizing the distribution of individual variables and the associations between two variables. The diagonal histogram provides insight into the distribution of a single variable, while the scatter plots in both the upper and lower triangles reveal the connection, if any, between two variables.

In our dataset,

- Week3_MP1 shows a clear distinction in the distribution of grade 0.
- The distribution for Week8_Total shows a good degree of separation, but there is also some overlap among the data points.
- All of the distributions seem to have a good amount of overlap.

Data Split

We partitioned the dataset into training and testing subsets, allocating 30% of the data to the test set. We have also separated the features (X) from the target variable 'Grade' (y). Both training and testing sets consist of a set of features (X) and their corresponding target values(y). Our model learns patterns and relationships between features and targets using the training set. Whereas, we use the testing set to assess how well the model generalizes to new, unseen data.

Model Training

Subsequently, We ran two distinct supervised models:

- KNeighborsClassifier
- RandomForestClassifier

Performance Evaluation

We evaluated their performance using metrics, including the confusion matrix. The confusion matrix provides a comprehensive snapshot of a model's performance, breaking down its predictions into four categories: true positives, true negatives, false positives, and false negatives. This information can be used to assess the accuracy, precision, recall, and F1-score of the models, shedding light on their effectiveness in predicting students' final grades.

RandomForestClassifier performed better than K-Nearest Neighbors

The better performance of the RandomForestClassifier, which achieved an accuracy of 93.94%, compared to the K-Nearest Neighbors (KNN) classifier with an accuracy of 84.85%, indicates that the RandomForest model was more successful in making correct predictions on our dataset.

The characteristics of your dataset play a significant role in the choice of an algorithm. RandomForest, being a tree-based method, can handle a variety of data distributions and doesn't rely on strict assumptions about data. In our dataset we don't have a very clear pattern and also contain overlap. KNN is sensitive to data distribution. Since our data doesn't exhibit clear, consistent patterns, KNN can not perform very well.

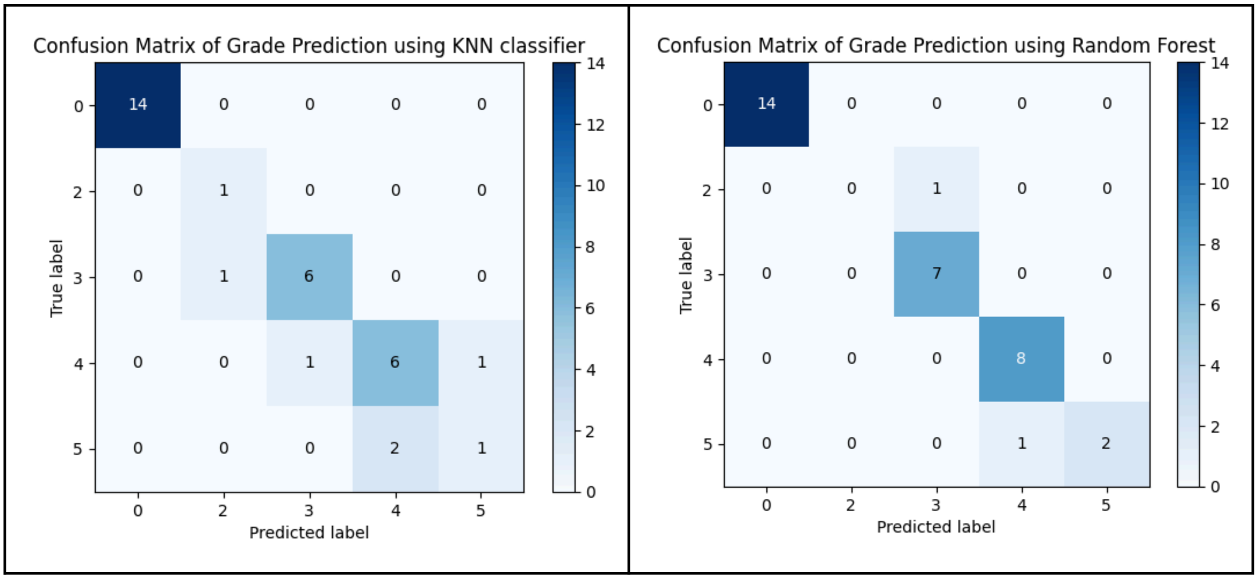

Fig: Confusion Matrix for KNN and Random forest classifier

For the Random Forest model, the results are as follows:

Precision: This metric measures the accuracy of positive predictions. The model achieved perfect precision (1.00) for class "0," indicating that when it predicted this class, it was correct every time. However, for classes "2" and "5," precision is notably lower, indicating that the model had difficulty correctly identifying these classes.

Recall: Recall quantifies the model's ability to capture all positive instances. The model demonstrated strong recall for classes "0," "3," and "4," suggesting it could effectively identify most of the relevant cases for these classes. However, recall is lower for class "2" and class "5."

F1-score: The F1-score is the harmonic mean of precision and recall. It provides a balance between the two metrics. The F1-scores for classes "0," "3," and "4" are relatively high, but they are considerably lower for classes "2" and "5."

Accuracy: The overall accuracy of the model is **93.94%**. This figure represents the proportion of correctly predicted instances out of the total.

Confusion Matrix: The confusion matrix shows the model's performance for each class. It indicates the number of true positives, true negatives, false positives, and false negatives. For class "0," the model performed perfectly, while for classes "2" and "5," it struggled to make accurate predictions.

In summary, the Random Forest model demonstrates strong performance for some classes, particularly class "0," but faces challenges in correctly predicting classes "2" and "5." The overall accuracy of **93.94%** suggests that it is a promising model, but further analysis and potential adjustments may be needed to improve predictions for specific classes.

For the K-Nearest Neighbors (KNN) model, the performance metrics are as follows:

Precision: The model achieved perfect precision (1.00) for class "0," indicating that when it predicted this class, it was correct every time. However, for classes "2", “3”, "4", and "5" precision is lower, indicating that the model had some difficulty correctly identifying these classes.
Recall: The model demonstrated strong recall for classes "0" and "2," suggesting it could effectively identify most of the relevant cases for these classes. However, recall is lower for classes "3" and "5."
F1-score: The F1-score is the harmonic mean of precision and recall. It provides a balance between the two metrics. The F1-scores for class "0" and class "3" are relatively high, but they are lower for classes "2," "4," and "5."
Accuracy: The overall accuracy of the KNN model is 84.85%. This figure represents the proportion of correctly predicted instances out of the total.
Confusion Matrix: For class "0," the model performed perfectly, while for other classes, there are varying degrees of correct and incorrect predictions.

In summary, the KNN model demonstrates strong performance for certain classes, particularly class "0," but faces challenges in correctly predicting classes "3," "4," and "5." The overall accuracy of **84.85%** suggests that it is a promising model, but there is room for further improvement, especially in correctly classifying some of the other classes.

Important features

In scikit-learn, tree models have a property called .feature_importances_ that becomes available once the model is trained. This property stores the scores representing the importance of each feature in the dataset. I used this property to obtain the feature importance values.

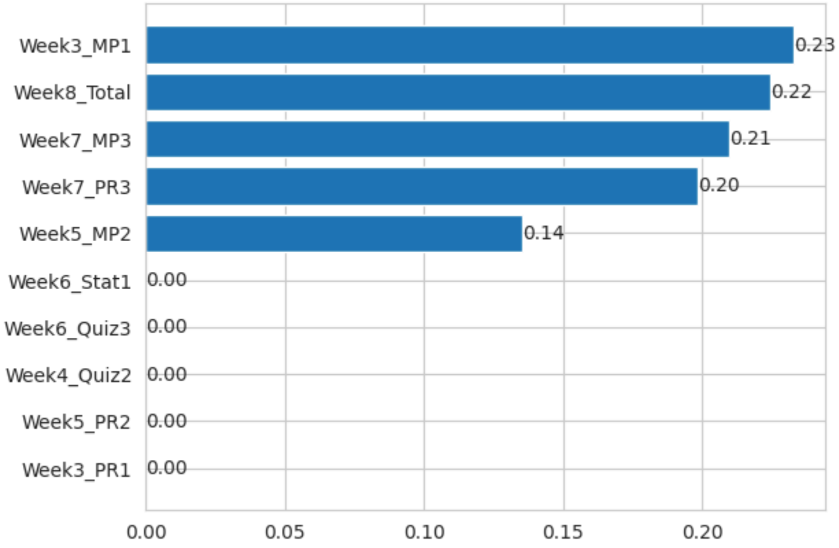

Fig: feature_importances_ values of Random Forest Model

From the feature_importances_ list we can see that,

- **Week3_MP1** is the most important one. This feature has an importance value of approximately 0.2327, indicating that it is the most influential feature among the ones that have been used.
- Next one is **Week8_Total** which has an importance value of approximately 0.2243, making it the second most important feature in the list.
- **Week7_MP3** has a value of approximately 0.2096, making it the third most important feature in the list.

Conclusion

In summary, our report evaluated two machine learning models, Random Forest and K-Nearest Neighbors (KNN), for predicting students' final grades in an online course. The Random Forest model demonstrated good performance but faced challenges with specific classes, achieving an accuracy of 93.94%. The KNN model excelled in some classes but struggled with others, resulting in an accuracy of 84.85%. Feature importance analysis highlighted critical variables for prediction. Further refinements are necessary to improve model performance, especially for specific classes.

The scientific bottlenecks we encountered revolved around the selection of relevant features and the identification of a model that aligns effectively with our dataset. Evaluating model performance posed a significant challenge, and representing data visually to highlight specific patterns also presented scientific obstacles.