Introduction

The aim of the project is developing a machine learning model that can predict student grades with given Moodle data. The data is collected from nine weeklong online ML course which is hosted on the online learning management system Moodle. The dataset is high dimensional with 48 features and has a small number of samples when comparing of number of columns. There are 107 students, and the grade scale is between 0 to 5. Hence, the model needs to be developed by supervised learning algorithms with a categorical prediction.

The data includes 9 grades and 36 logs that represents the number of activities done by each student in the Moodle website such as content page viewing, submission update or creating a discussion. All the data is numerical, the grades are floating numbers and logs are integers.

Data Processing

Every entry has a unique id which there is no need to store in dataset for a machine learning model. Hence, the id column has been dropped for the first step. Secondly, the data has been analyzed and checked if there is any null value and it has been observed that there is no null value in any of the columns. However, the data still needs to be checked for if there is a column with all values are the same. Because, if all values are the same it does not provide any information for a ML model. When it has been checked, it is observed that all values are the same in Week1_Stat1 column. Therefore, there is no need to have Week1_Stat1 feature in the dataset then it has been dropped. There are two target values in the datasets, one is Week8_Total and the other is Grade. Since the grade is calculated by Week8_Total, it should not be included as a feature. Otherwise, the model can predict the student's grade by only checking Week8_Total feature. Hence, Week8_Total is dropped and only grade is used for target.

Machine Learning Models Training & Testing

The data has been divided into two as train and test with respectively 75 and 25 percentages. Moreover, features were stored as X and labels were stored as Y. The first model has been developed is a Random Forest model. Trained data overfits the model and it achieves 100% accuracy for training dataset. However, it achieves 0.63 accuracy for the test dataset. The

second method is k-NN. It uses 3 neighbors to predict the classes and achieves 0.775 and 0.593 accuracies respectively for the training and testing datasets. Since both of the models cannot perform very well, a third model has been developed by using Decision Tree. It achieves %100 accuracy for training and 0.778 for testing dataset. All of the models compared by confusion matrices in Figure I.

Predicted Grades / Actual Grades	0	3	4	5
0	10	0	0	0
2	0	2	1	0
3	0	2	1	0
4	0	1	5	1
5	0	3	1	0

Predicted Grades / Actual Grades	0	2	3	4
0	10	0	0	0
2	0	1	2	0
3	0	0	3	0
4	1	1	3	2
5	1	1	2	0

Predicted Grades / Actual Grades	0	3	4	5
0	10	0	0	0
2	1	2	0	0
3	0	3	0	0
4	0	1	5	1
5	0	1	0	3

Figure I: Confusion matrices of Random Forest, kNN and Decision Tree

k-NN is effective when there is a large number of training samples but in this problem, there is not much data. On the other hand, Random Forest can handle high dimensional spaces very well and can have good results even if the number of samples is not high. The reason why Decision Tree is the best model for this problem is that there are too many features and some of them are much important than the others. Since random forest selecting the features randomly, it uses the features with has very low importance. On the other hand, the decision tree chooses only the most important features and fits the data better and as a result has a higher accuracy than random forest. Figure II shows that how the decision tree is formed by using some of the most important features.

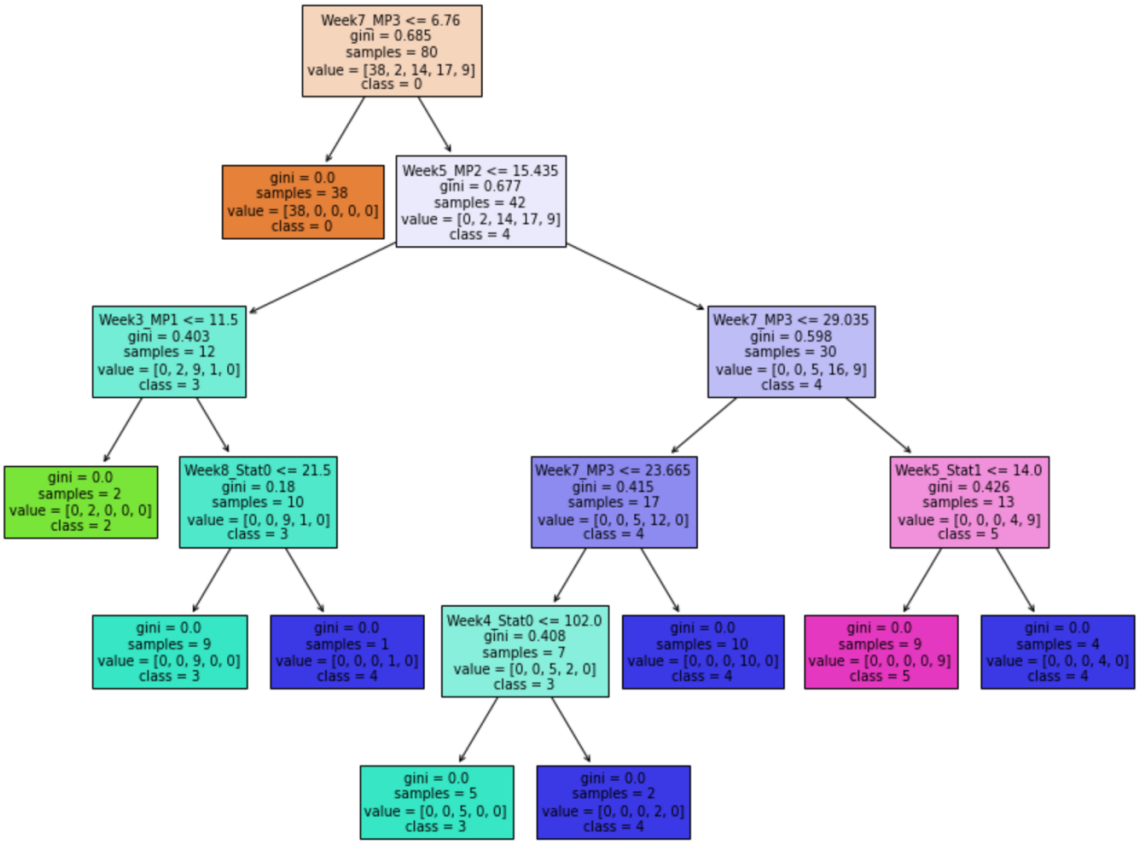

Figure II: Visualization of the decision three model

Random Forest could be generated any number of trees, to decided how many trees should be used the model has been developed again and again with different number of trees. It is seen that the best result is with 15 trees and using less then 10 trees could cause a low accuracy.

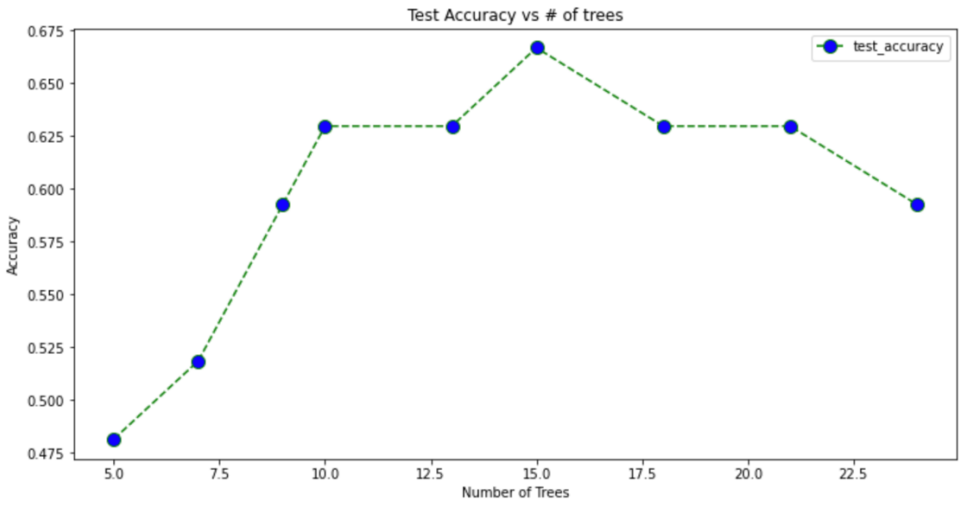

Figure III: Test Accuracy vs Number of Trees

Another study has done for k-NN algorithm and for its number of neighbors. Using 1 or 3 neighbors leads to around 0.6 accuracy and using 5 or more is leading around 0.52 accuracy.

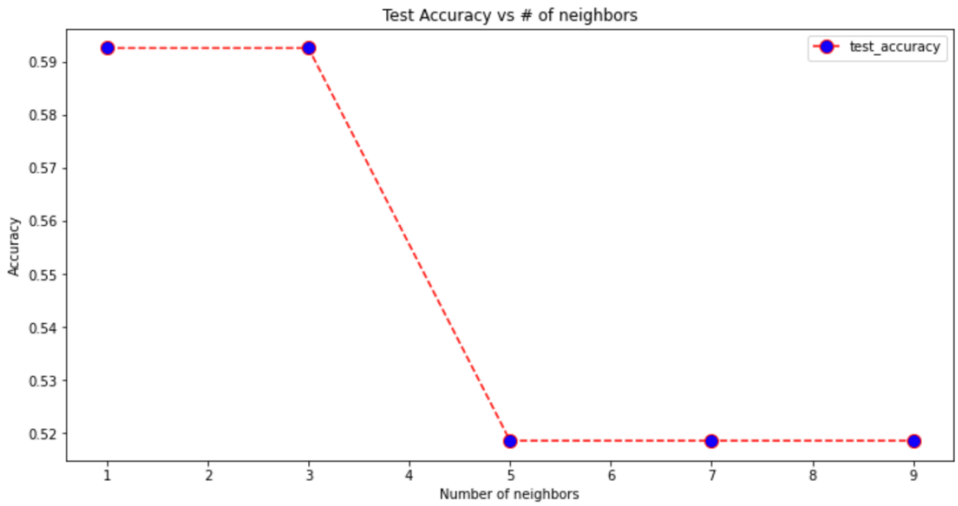

Figure IV: Test Accuracy vs Number of Neighbors

Importance of Features

The most important three features are Week7_MP3, Week5_MP2 and Week4_Stat0. It was expected since Mini Project 2 and Mini Project 3 are the assignments with the highest effect on total grade. 3rd most important feature which is Week4_Stat0 proves that students who reviewed courses and watch the lectures on week 4 has a positive correlation with Week5_MP2 grade which makes it also important feature. In figure V, the most important nine features are visualized.

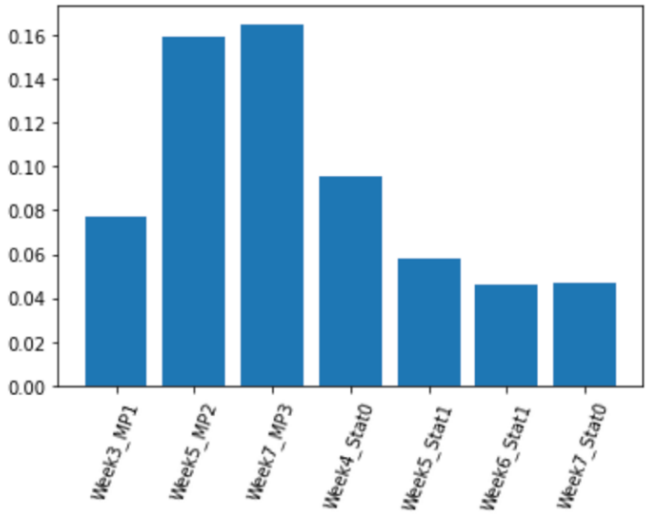

Figure V: Bar chart of features in terms of importance

Conclusion

The models are implemented successfully and the differences between various machine learning approaches are examined. It is shown that random forest is not always better than decision trees. If there are many features and some of the features has low importance than others, the trees created by those features could have a negative impact on the final model. Another study could be combining some of the features and decreasing the dimension by for example summing all log values for a week or getting only average log values of different types in each week. In conclusion, the problem is open for future studies and this study provides a detailed analysis and three different machine learning models that achieved various accuracies.