Student Performance Analysis

Introduction

The project, Student performance, aims to predict students' final grades in an online course on Machine Learning on Moodle platform. The project uses the two most popular supervised machine learning algorithms, namely, Decision Trees Classifier and Random Forest Classifier for predicting the grades. Through this project, we will also acknowledge how a prediction algorithm can be used for the identification of the most important features/attributes in our dataset.

Data Description:

The data is collected from a nine-week-long course on Machine learning from Moodle. The data includes information on 9-week course logs, grades of 3 quizzes, 3 mini-projects, and 3 peer reviews. Each course log has 4 statuses, each belonging to content-related, assignment related, grade related and forum-related.

Data Analysis:

On exploring the given dataset, we find the details of 107 anonymous student records, each having 48 attributes. Each record is uniquely identified by an ID and has 36 course logs (content, assignment, grade and forum related) for 9 weeks, 9 grades (Quizzes, Mini Project and Peer Review), Week8_total which represents the sum of these 9 grades and a Final Grade scaled from 0-5.

- ***Missing values***: No missing values were found in the dataset
- ***Feature Selection:***

  Since there are a lot of features in the dataset, some of the correlated features have been combined to reduce the dimensionality and some unimportant features are discarded.

  Looking at the description of the dataset, the column Week1_Stat1(assignment related) is completely 0 since the first week does not have any assignments. So, this column has been discarded.

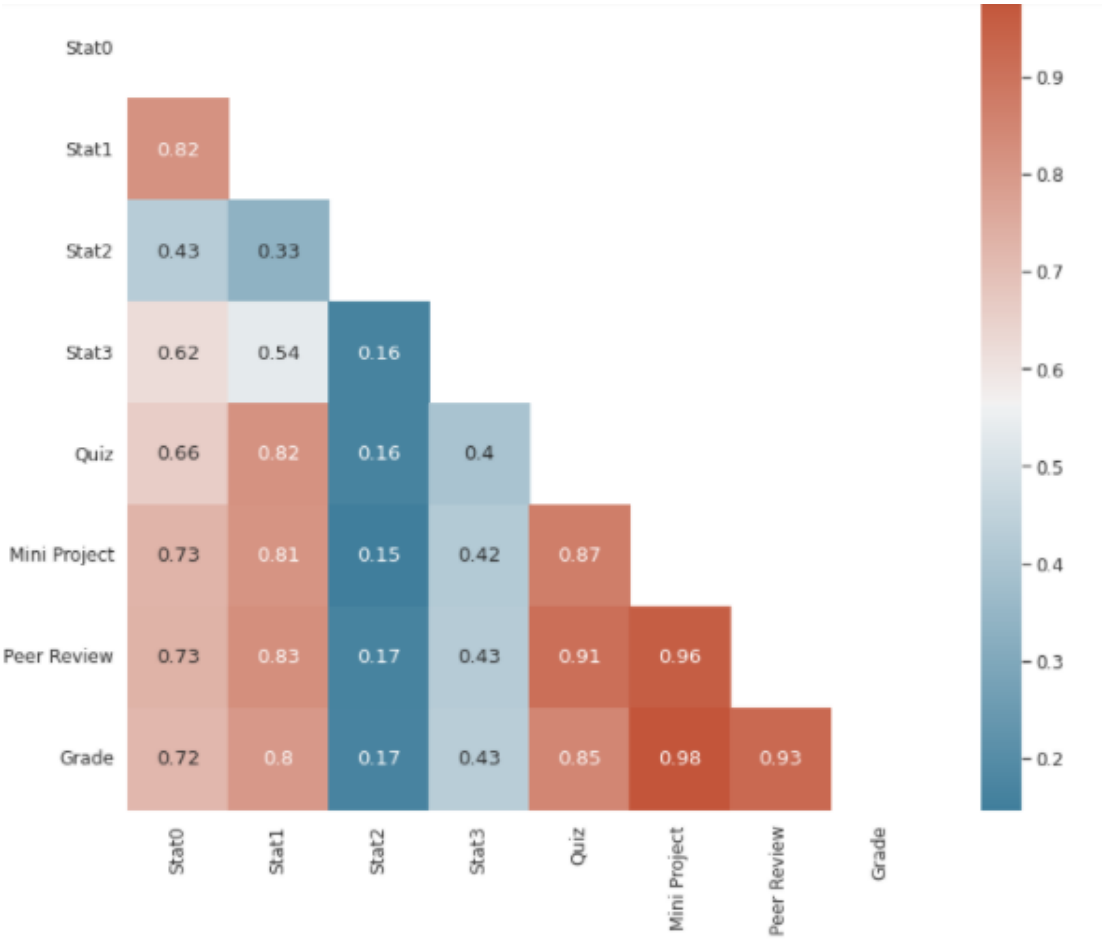
	Grade	Week1_Stat0	Week1_Stat1
count	107.000000	107.000000	107.0
mean	2.074766	6.785047	0.0
std	1.993863	7.157300	0.0
min	0.000000	0.000000	0.0
25%	0.000000	1.000000	0.0
50%	3.000000	4.000000	0.0
75%	4.000000	12.000000	0.0
max	5.000000	27.000000	0.0

Also, the column ‘Week8_Total’ has been removed because it is redundant to the target column ‘Grade’. The column ID can also be discarded since it is not useful for our model for prediction.
A new data frame has been created to store the required features for training our machine learning model. The columns of Status0, Status1, Status2, Status3, Quizzes, Mini Project and Peer review of each week have been aggregated and stored in the new dataset. So,

	Stat0	Stat1	Stat2	Stat3	Quiz	Mini Project	Peer Review	Grade
0	118	119	8	4	15.00	52.97	15.00	4
1	465	85	17	37	12.33	55.10	15.00	4
2	169	65	8	9	11.67	55.27	12.50	3
3	553	74	17	15	10.63	55.02	15.00	3
4	149	62	7	17	9.67	43.08	14.93	2

after filtering all the columns, we have 7 features for our model.

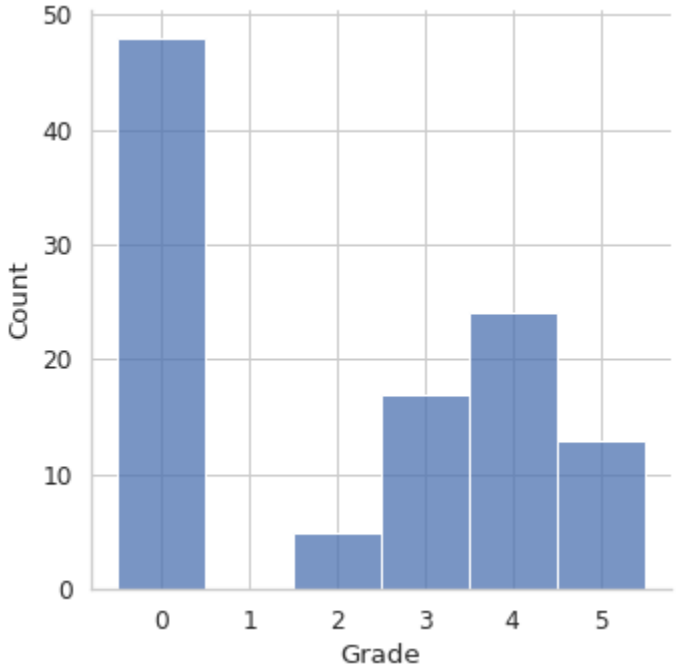
❖ ***Checking correlation:***



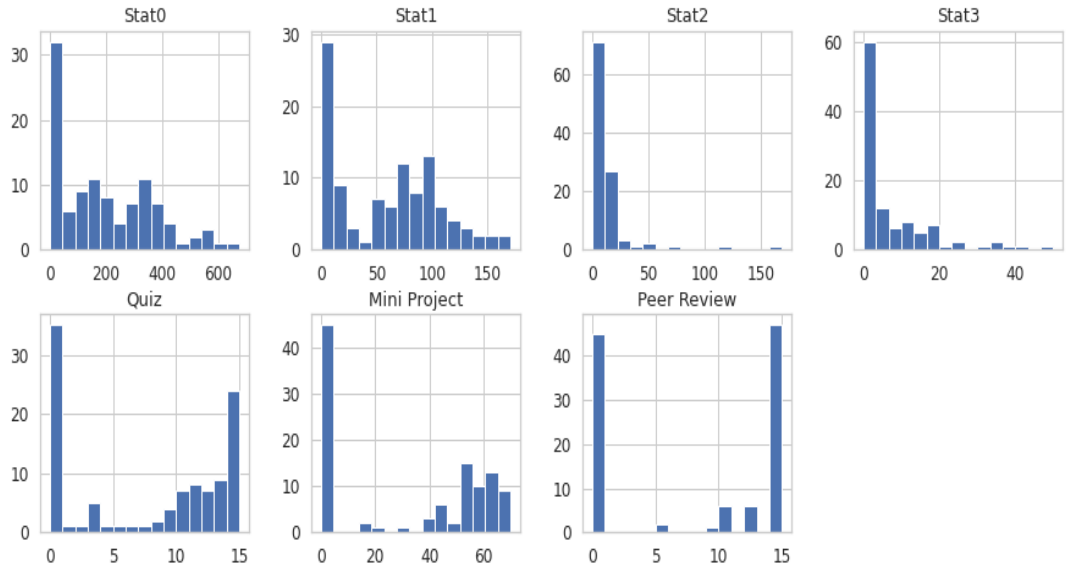
In the last line of the correlation heatmap, we can see that grade is strongly and positively correlated to Mini Project, Peer Review and Quiz. There is also a positive correlation with other features like the stats.

❖ ***Data Exploratory analysis:***

Distribution of Grades: The dataset only has grades of 0,2,3,4,5. There is no occurrence of grade 1 in our dataset. That means it is not possible to make predictions for grade 1.



Distribution of all features:



- ***Training & Test Dataset:***
  From sklearn.model_selection, train_test_split function has been used to split the 107 records of data into training and testing data with 70% and 30% records respectively.
- ***Standardizing data:***
  Before moving to train the model, feature scaling is done in order to bring all the independent variables of the dataset into the same scale to avoid any feature dominating the model. StandardScaler function is used for feature scaling which normalizes each feature with mean around 0 and variance around 1.

Applying Classification Algorithms:

1. ***Decision trees*** are one of the most popular techniques used for prediction. It is a supervised algorithm used for both classification and regression.

2. ***Random forest*** is a tree-based algorithm that learns from multiple decision trees to make decisions. In this algorithm, we simply combine multiple (random) decision trees to predict the final output.

Model Evaluation:

1. ***Decision trees classifier:***
   - accuracy_score() function is used for checking the accuracy of the trained model by matching the predicted values of the testing subset to the actual values of the testing subset.
   - The trained model gave an accuracy of 88%.
   - *Confusion matrix:*

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

Findings of the confusion matrix:

- Predictions were made on 5 classes: ‘0’, ‘2’, ‘3’, ‘4’, ‘5’
- The classifier has correctly predicted all the occurrence of grades ‘0’ and ‘2 ’15 times and 3 times respectively.
- The classifier has wrongly predicted grade ‘3’ as grade ‘4’ once, grade ‘4’ as grade ‘3’ twice and grade ‘5’ as grade ‘4’ once.

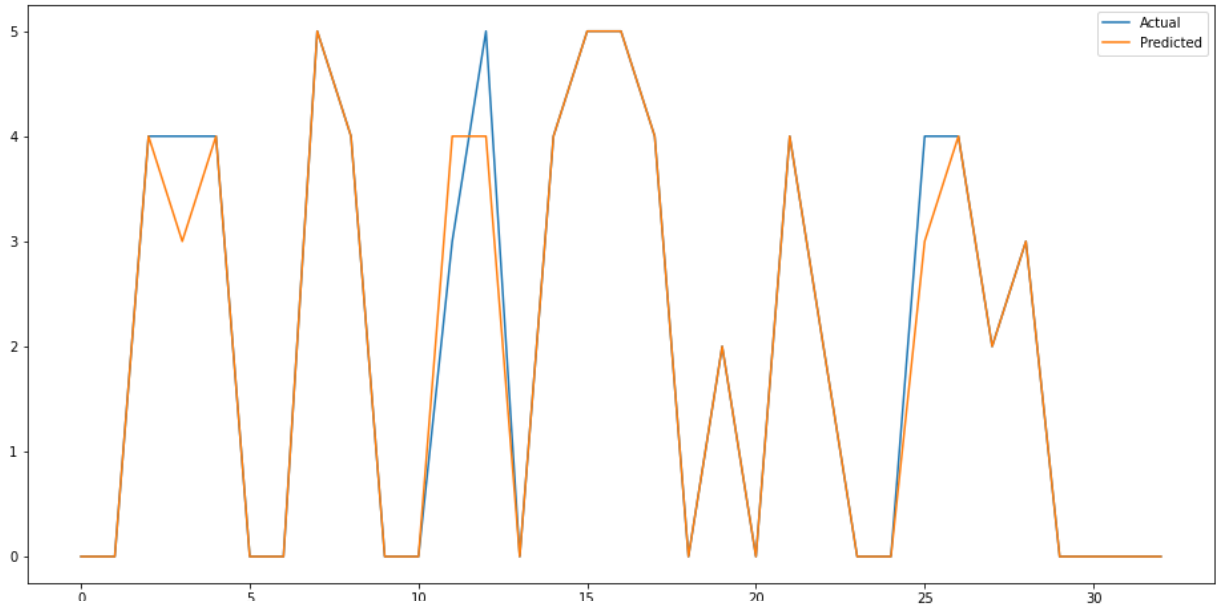
❖ *Classification report:*

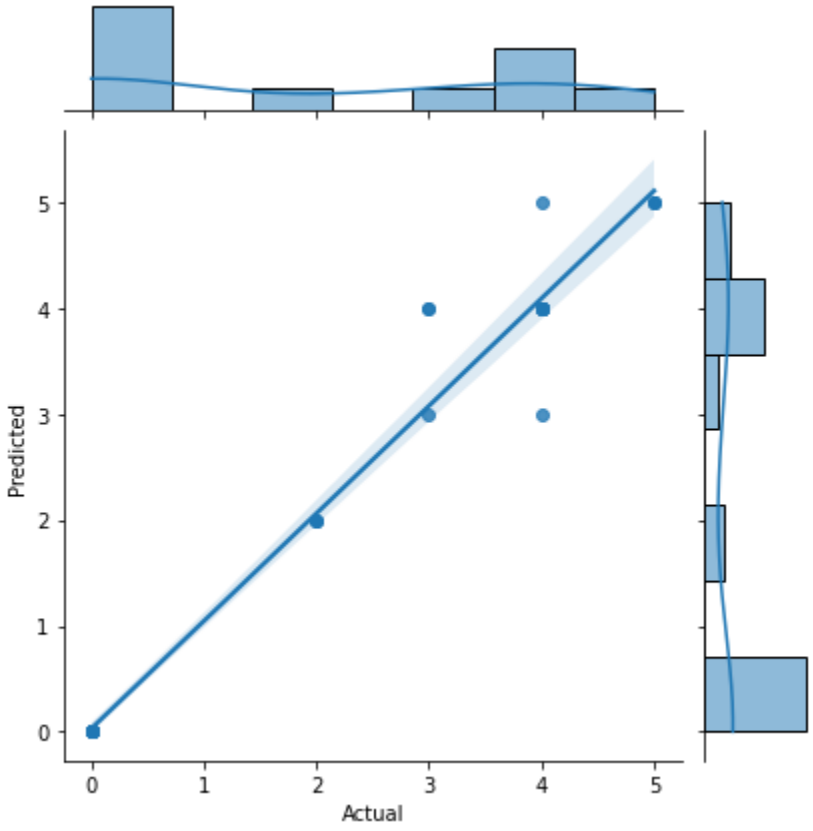
	precision	recall	f1-score	support
0	1.00	1.00	1.00	15
2	1.00	1.00	1.00	3
3	0.50	0.33	0.40	3
4	0.78	0.78	0.78	9
5	0.75	1.00	0.86	3
accuracy			0.88	33
macro avg	0.81	0.82	0.81	33
weighted avg	0.87	0.88	0.87	33

The classification report compares different classification metrics on each predicted class:

- ➢ **Precision** tells what proportion of the positive prediction was correct. The classifier produced no false positives for grades '0' and '2' with a precision of 1. Whereas, grade '3', grade '4' and grade '5' would be only 50%, 78% and 75% correct of the time respectively.
- ➢ **Recall** tells what proportion of the predictions of the class were actually labelled right. The classifier labels grades '0' and '2' correctly. Whereas only 33% and 78% of grades '3' and '5' were identified correctly.
- ➢ **F1-score** defines the harmonic mean of precision and recall. Grades '0','2' and '5' have good f1-scores.
- ➢ **Support** tells the number of occurrences of the class in the dataset. Grade '0' has more occurrences and the best precision value. Even though grade '2' has fewer occurrences than grade '4', grade '2' has the best precision score.

❖ Visualization of the actual vs predicted values of the Decision tree classifier:





The above graph shows that the predicted values are not that far from the actual values. So, above all Decision tree is a good classifier for the problem.

2. ***RandomForest Classifier:***

- accuracy_score() function is used for checking the accuracy of the trained model by matching the predicted values of the testing subset to the actual values of the testing subset.
- The trained model gave an accuracy of 88%.
- *Confusion matrix:*

Predicted	0	2	3	4	5
Actual					
0	15	0	0	0	0
2	0	3	0	0	0
3	0	0	2	1	0
4	0	0	0	9	0
5	0	0	0	0	3

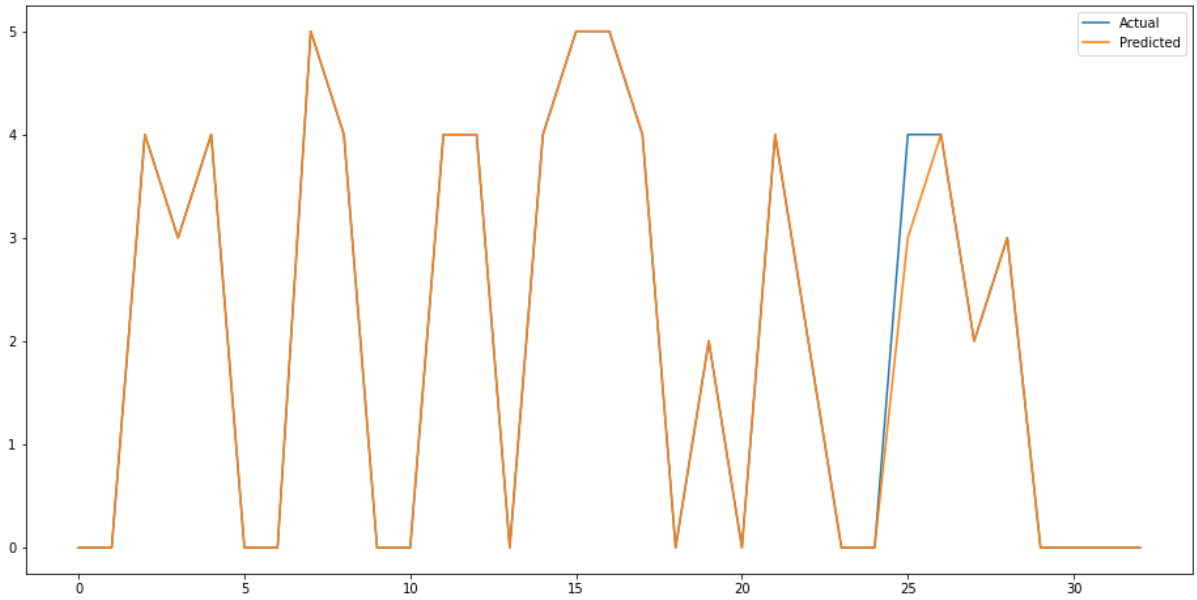
Findings of the confusion matrix:

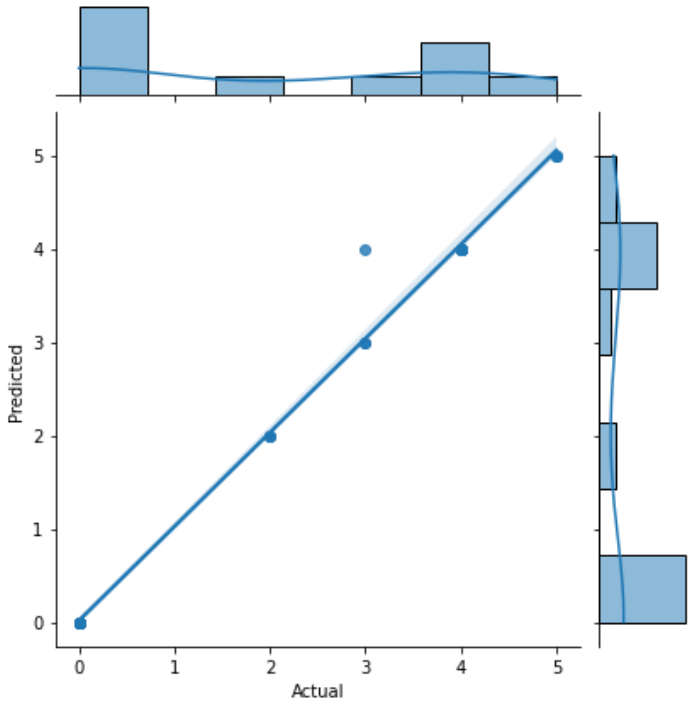
- Predictions were made on 5 classes: '0', '2', '3', '4', '5'
- The classifier has correctly predicted all the occurrences of grades '0', '2', '3', and '5'
- The classifier has made only one wrong prediction of labelling grade '4' as grade '3' once.

- *Classification report*

	precision	recall	f1-score	support
0	1.00	1.00	1.00	15
2	1.00	1.00	1.00	3
3	1.00	0.67	0.80	3
4	0.90	1.00	0.95	9
5	1.00	1.00	1.00	3
accuracy			0.97	33
macro avg	0.98	0.93	0.95	33
weighted avg	0.97	0.97	0.97	33

- Visualization of the actual vs predicted values of the RandomForest classifier:



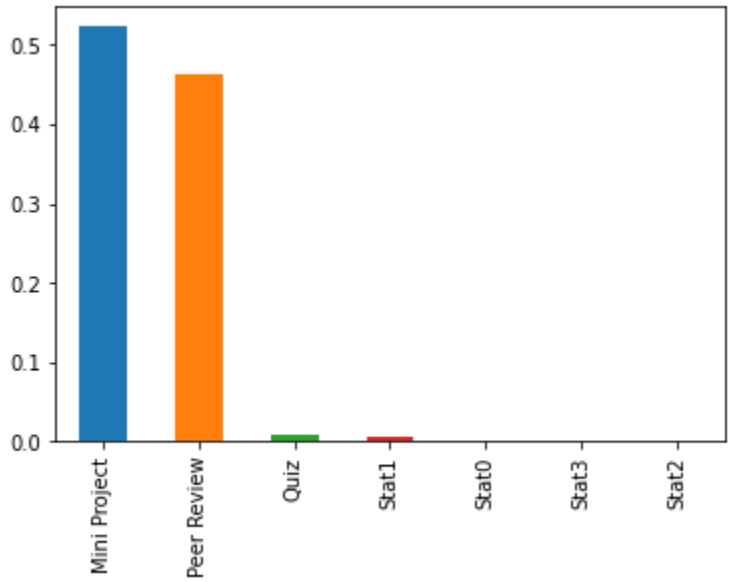


It is clear from the graph that, most of the predicted values of the classifier are right. On comparing RandomForest Classifier with Decision Tree Classifier, it is found that the RandomForest classifier is the best model for this project with an accuracy of 97%.

Important features

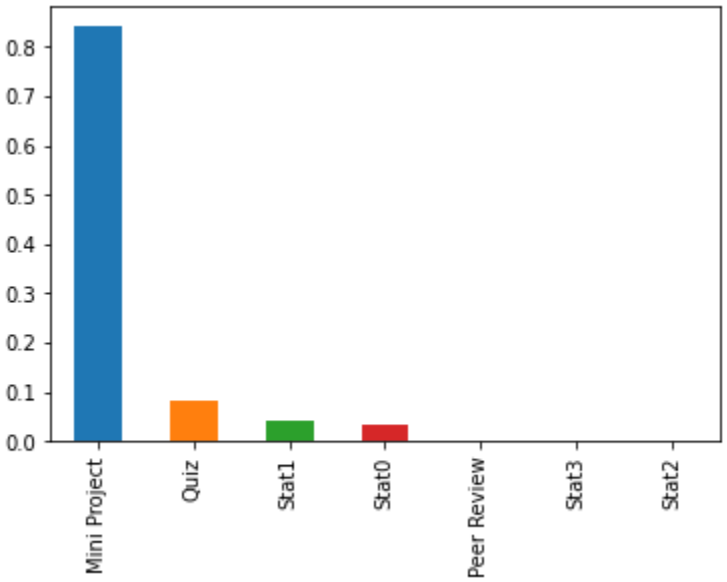
Important features of both the classifiers are calculated by an attribute of TreesClassifier called, feature_importances_. They are calculated as the mean and standard deviation of impurity decrease accumulation inside each tree.

- The top 3 important features of the RandomForest classifier are: Mini Project, Peer Review and Quiz




```
Mini Project    0.523649
Peer Review     0.462049
Quiz            0.007608
```

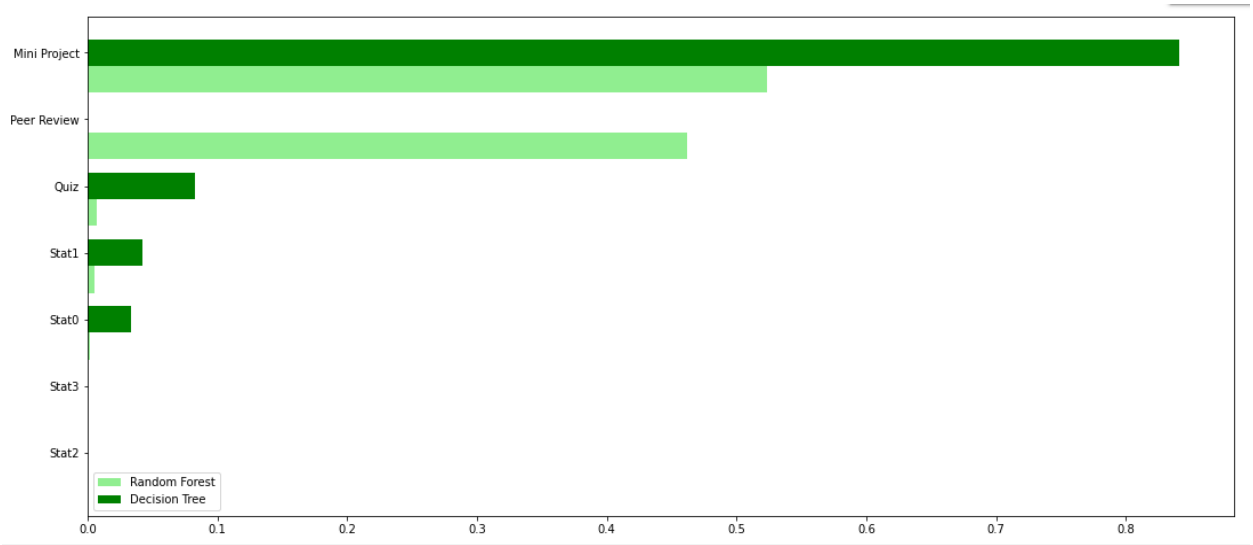
- The top 3 important features of the DecisionTree classifier are: Mini Project, Quiz and Stat1(assignment related)




```
Mini Project    0.841452
Quiz            0.083031
Stat1           0.041972
```

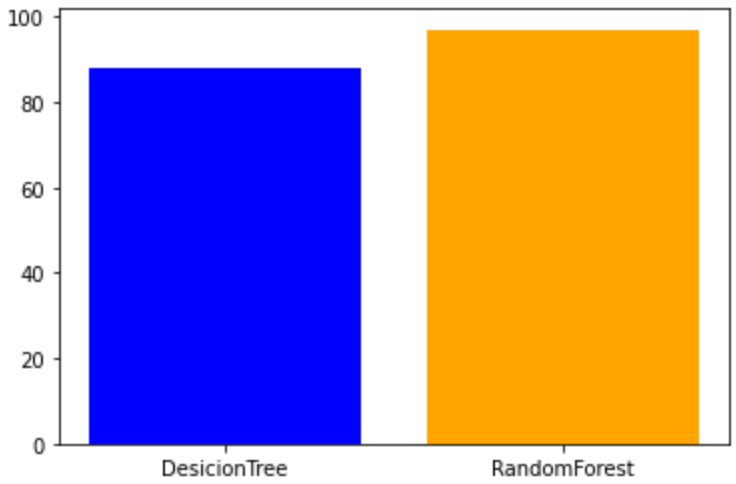
Insights:

1. **Data Processing:**
   There were no missing values in the dataset. Redundant and unimportant columns were discarded (like ID, Week8_Total, Week1_Stat1). A new dataset was created based on the aggregation of closely related columns to reduce the dimensionality (like Stat0, Stat1, Stat2, Stat3, Quiz, Mini Project and Peer Review).
2. **Comparing the Classifiers:**
   Based on the accuracy and other performance metrics (shown in the classification report) applied to each model, RandomForest Classifier is found to be better than DecisionTree Classifier.



The reason why the RandomForest classifier did better than the DecisionTree classifier is that unlike in DecisionTree where the model gave high importance to a particular feature for creating the decision tree, the RandomForest model randomly chooses a set of features to create multiple decision trees(for each feature) and aggregate the output of all the decision trees to generate the final output. This aggregating of different models helps improve the overall model performance especially in generalizing new data.

3. **Performance evaluation:**
   - ***DecisionTree Classifier:*** The model with default parameters gives an accuracy of 87.88%. Accuracy of the model after tuning the parameters like splitter and criterion gives only 72.73% accuracy. No increase in performance was found.
   - ***RandomForest Classifier:*** The model with default parameters gives an accuracy of 87.88%. Accuracy of the model after tuning the parameter 'n_estimators' gave an accuracy of 96.97%. A significant increase in the performance of the model was found after tuning.



Comparison of two classifiers

Conclusion:

I have gained hands-on experience with feature scaling, applying supervised classification models and compare them. The first bottleneck I encountered was feature scaling, there were a lot of features in the dataset. Having a lot of features can impact the performance and the execution time of the model. Unimportant and redundant features were discarded and since some of the grade features (like quiz, mini project and peer review) are defined for each week. These columns were aggregated into a new dataset. Another challenge was tuning the model to increase its accuracy. I have learnt what, when and how to use different parameters of models to increase their performance.