

Students Performance Prediction Model Mini-Project Scientific Report

1. Introduction

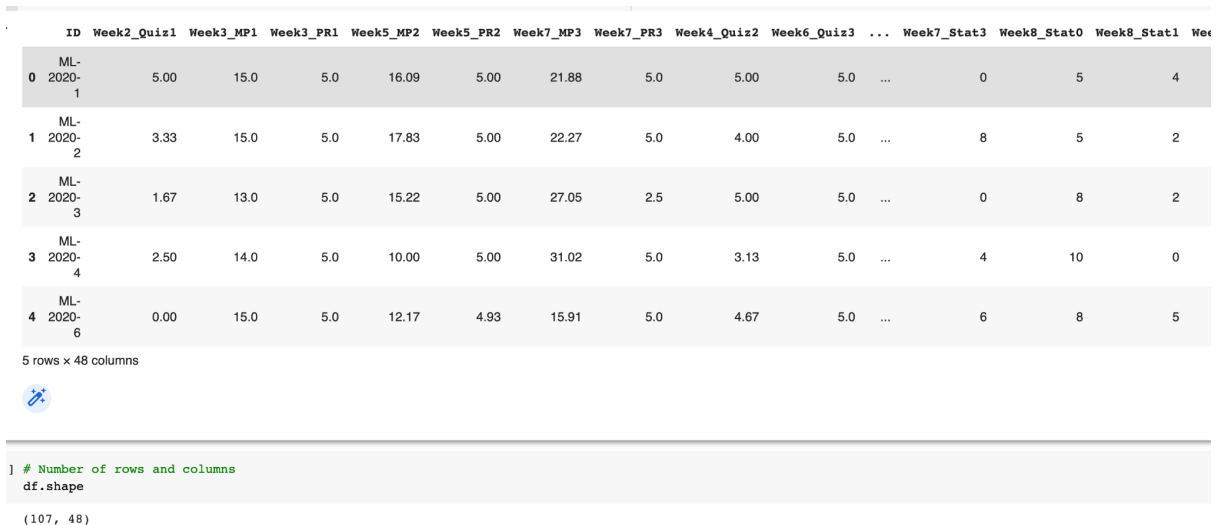
Machine-learning algorithms can be used in many places that need software applications to predict outcomes without being explicitly programmed. They use historical dataset as an input and provide prediction output values.

Calculating marks and assigning grades for students manually are tiresome and error-prone tasks. Alternatively we can build a machine-learning classifier model to predict a student's performance grade. The model can be trained by taking a dataset gathered from previously stored marks of students and give prediction for a new data instance.

This scientific report paper describes how two supervised machine-learning algorithm models are used to predict a student performance grade. The algorithms are Random Forest Classifier and Support Vector Classification(SVC). The dataset was collected from a fully online nine-week-long course on machine learning, hosted on the online learning management system Moodle. The evaluation scores for each model are mentioned in different metrics. In addition the top three important features to predict the students' grade also included in this report.

2. Exploratory Data Analysis(EDA)

The dataset contains student grades from 3 mini-projects, 3 quizzes and 3 peer reviews in addition their activity logs in the online learning management system Moodle.



The screenshot shows a Jupyter Notebook with two cells. The first cell displays a sample of the dataset as a table with 5 rows and 15 columns. The columns are: ID, Week2_Quiz1, Week3_MP1, Week3_PR1, Week5_MP2, Week5_PR2, Week7_MP3, Week7_PR3, Week4_Quiz2, Week6_Quiz3, ..., Week7_Stat3, Week8_Stat0, Week8_Stat1, and Week8_Stat2. The rows represent different students (ML-2020-1 to ML-2020-6). The second cell shows the output of the command `df.shape`, which is `(107, 48)`, indicating 107 rows and 48 columns.

	ID	Week2_Quiz1	Week3_MP1	Week3_PR1	Week5_MP2	Week5_PR2	Week7_MP3	Week7_PR3	Week4_Quiz2	Week6_Quiz3	...	Week7_Stat3	Week8_Stat0	Week8_Stat1	Week8_Stat2
0	ML-2020-1	5.00	15.0	5.0	16.09	5.00	21.88	5.0	5.00	5.0	...	0	5	4	
1	ML-2020-2	3.33	15.0	5.0	17.83	5.00	22.27	5.0	4.00	5.0	...	8	5	2	
2	ML-2020-3	1.67	13.0	5.0	15.22	5.00	27.05	2.5	5.00	5.0	...	0	8	2	
3	ML-2020-4	2.50	14.0	5.0	10.00	5.00	31.02	5.0	3.13	5.0	...	4	10	0	
4	ML-2020-6	0.00	15.0	5.0	12.17	4.93	15.91	5.0	4.67	5.0	...	6	8	5	

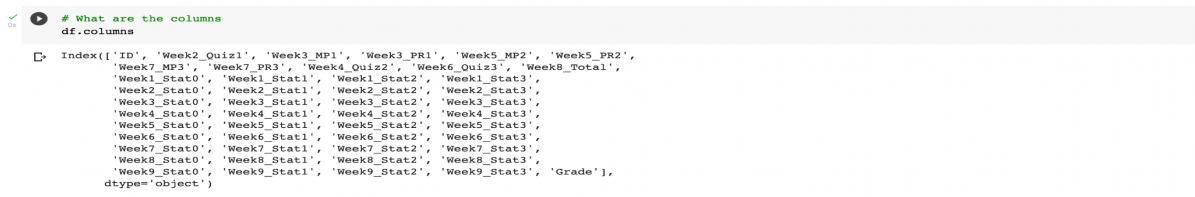
5 rows x 48 columns

```
# Number of rows and columns
df.shape

(107, 48)
```

Figure 1: Sample records and the dataset shape

There are 107 rows and 48 columns. The columns include 9 assessment grades, 36 course logs, 1 ID , 1 total assessment marks and 1 final grade column.



The screenshot shows a Jupyter Notebook with a cell containing the command `df.columns`. The output is a list of 48 column names, including IDs, weekly quiz and project scores, weekly peer review scores, weekly statistics, and a final grade column.

```
# What are the columns
df.columns

Index(['ID', 'Week2_Quiz1', 'Week3_MP1', 'Week3_PR1', 'Week5_MP2', 'Week5_PR2', 'Week7_MP3', 'Week7_PR3', 'Week4_Quiz2', 'Week6_Quiz3', 'Week8_Total', 'Week1_Stat0', 'Week1_Stat1', 'Week1_Stat2', 'Week1_Stat3', 'Week2_Stat0', 'Week2_Stat1', 'Week2_Stat2', 'Week2_Stat3', 'Week3_Stat0', 'Week3_Stat1', 'Week3_Stat2', 'Week3_Stat3', 'Week4_Stat0', 'Week4_Stat1', 'Week4_Stat2', 'Week4_Stat3', 'Week5_Stat0', 'Week5_Stat1', 'Week5_Stat2', 'Week5_Stat3', 'Week6_Stat0', 'Week6_Stat1', 'Week6_Stat2', 'Week6_Stat3', 'Week7_Stat0', 'Week7_Stat1', 'Week7_Stat2', 'Week7_Stat3', 'Week8_Stat0', 'Week8_Stat1', 'Week8_Stat2', 'Week8_Stat3', 'Week9_Stat0', 'Week9_Stat1', 'Week9_Stat2', 'Week9_Stat3', 'Grade'],
      dtype='object')
```

Figure 2: Original dataset Columns

Students Performance Prediction Model Mini-Project Scientific Report

Before starting to train a model data preprocessing is necessary. After reviewing the dataset since the ID column doesn't have a value to the classification modeling task, it is removed from the dataset. As Figure 3 depicts there is no missing value found in the dataset.

```
# Check for missing values
df.isnull().sum()

Week2_Quiz1    0
Week3_MP1     0
Week3_MP1     0
Week5_MP2     0
Week5_MP2     0
Week7_MP3     0
Week7_MP3     0
Week8_Quiz2   0
Week8_Quiz2   0
Week8_Quiz3   0
Week8_Total   0
Week1_Stat0   0
Week1_Stat1   0
Week1_Stat2   0
Week1_Stat3   0
Week2_Stat0   0
Week2_Stat1   0
Week2_Stat2   0
Week2_Stat3   0
Week3_Stat0   0
Week3_Stat1   0
Week3_Stat2   0
Week3_Stat3   0
Week4_Stat0   0
Week4_Stat1   0
Week4_Stat2   0
Week4_Stat3   0
Week5_Stat0   0
Week5_Stat1   0
Week5_Stat2   0
Week5_Stat3   0
Week6_Stat0   0
Week6_Stat1   0
Week6_Stat2   0
Week6_Stat3   0
Week7_Stat0   0
Week7_Stat1   0
Week7_Stat2   0
Week7_Stat3   0
Week8_Stat0   0
Week8_Stat1   0
Week8_Stat2   0
Week8_Stat3   0
Week9_Stat0   0
Week9_Stat1   0
Week9_Stat2   0
Week9_Stat3   0
Grade         0
dtype: int64
```

Figure 3: missing value report

The target class for this dataset is the Grade column which has five distinct values. The values are 0, 2, 3, 4, and 5. There are 48 instances of grade 0, 24 instances of grade 4, 17 instances of grade 3, 13 instances of grade 5 and 5 instances of grade 2. Figure 4 indicates that there are more sample records who have a class zero grade, only few records for a class two grade and none records for a class grade one.

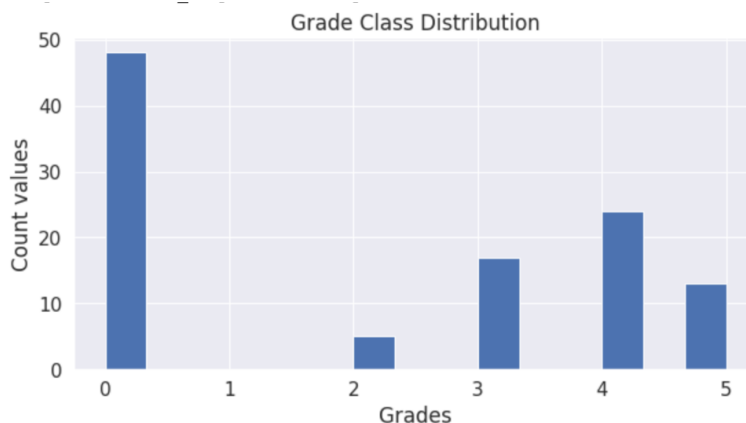


Figure 4: Grade class distribution histogram

For the random forest classifier model, there are potentially 46 columns that can be included in the feature set, however since we have a limited number of sample data having all the features in the model will degrade the performance of the model as well as it will add unnecessary time overhead. To reduce the number of features for the model, the statistical relationship(correlation) between the feature variables and the target class are calculated and only the variables who have 0.5 and above correlation point value with the target class are selected.

Students Performance Prediction Model Mini-Project Scientific Report

Other criteria was also considered to minimize the features, for example calculating the correlation and if two variables have more than 90% correlation then select only one variable out of the two, however these approaches don't help the model to have a better prediction performance. Finally only 19 variables are picked to be in the feature list. One of the feature in this list is Week8_Total feature which is basically the summation of all the other 9 grades. When this feature is not included in the model, it performs less with about 81% of accuracy, whereas adding this feature to the model increases the accuracy to about 95%.

```
Index(['Week2_Quiz1', 'Week3_MP1', 'Week3_PR1', 'Week5_MP2', 'Week5_PR2',  
      'Week7_MP3', 'Week7_PR3', 'Week4_Quiz2', 'Week6_Quiz3', 'Week8_Total',  
      'Week3_Stat0', 'Week3_Stat1', 'Week4_Stat0', 'Week4_Stat1',  
      'Week5_Stat0', 'Week6_Stat0', 'Week6_Stat1', 'Week8_Stat1',  
      'Week9_Stat0'],  
      dtype='object')
```

Figure 5: Random Forest Model Feature list

The support vector classification model performs better with a large number of feature list with limited number of samples. First the 19 features were used in the svc model and the accuracy was about 86%, when almost all variables except the Week8_Total are included and the model scores better with about 95% accuracy. Thus, two separate feature lists are used for each model.

3. Split Dataset To Train and Test Data

There are 107 sample examples are found in the original dataset. The model should be tested in a new dataset that was not used in the training phase. 80:20 ratio is used to split the dataset into training and testing data, hence 85 examples are used for the training and 22 examples are used for testing.

The data is splitted randomly by using the train_test_split method of scikit learn library. However, as it is explained in figure 4, there is some imbalance between the classes of the sample data, therefore instead of using random sampling, stratified random sampling will be better in this case. Stratified sampling ensures splitting the data randomly and keeping the same imbalanced class distribution for both the training and testing set. To implement the stratified sampling, stratify parameter is set in the train_test_split method.

4. Train the Model

Two supervised machine learning algorithms are used to train the model. The first model is Random Forest Classifier and the second one is Support Vector Classification(SVC) classifier.

The scikit-learn RandomForestClassifier constructor receives different parameters, however only random_state parameter was given in this case to keep the prediction result consistent. The other important parameter that was considered but not given is n_jobs parameter which is used to run the training process in parallelism manner. Although this parameter is very helpful when there is a large amount of data, for our dataset 1 CPU core is enough to do the training process.

SVC is one of the algorithm that is categorised under Support Vector Machine(SVM). This algorithm can be implemented using the SVC class from scikit-learn library. One of the parameter of this class constructor is kernel. Different kernel values can be used

Students Performance Prediction Model Mini-Project Scientific Report

to transform the training data to be suitable for the model. In our model since we have many features linear kernel is applied, it is mostly used when there are a large number of features in the dataset.

The two models perform better in different feature set. The random forest model scores about 95% accuracy with only limited number of features rather than using more features whereas the support vector classifier has better accuracy with large number of features with about 95% accuracy. Although, accuracy is not the only performance evaluation metrics, infact other important metrics are also used to measure the performance of our models.

5. Performance Evaluation

In Addition to accuracy, other evaluation metrics: precision, recall, f1-score and confusion matrix are computed to evaluate the performance of the models. The accuracy of both models is 95.455% with different feature list. The features used for random forest classifier doesn't work properly for the svc classifier.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	10
3	0.75	1.00	0.86	3
4	1.00	1.00	1.00	5
5	1.00	1.00	1.00	3
micro avg	0.95	1.00	0.98	21
macro avg	0.94	1.00	0.96	21
weighted avg	0.96	1.00	0.98	21

Figure 6: Classification Report For Random Forest Classifier

The random forest classifier model predicts with a better precision score for classes 0, 4, and 5. The f1-score is 100% for these three classes and 86% for class 3. However, the model doesn't predict any sample example as class 2. The reason for this error is the collected sample data has less number of examples with class 2 than the other classes.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	10
2	1.00	1.00	1.00	1
3	1.00	1.00	1.00	3
4	0.83	1.00	0.91	5
5	1.00	0.67	0.80	3
accuracy			0.95	22
macro avg	0.97	0.93	0.94	22
weighted avg	0.96	0.95	0.95	22

Figure 7: Classification Report For SVC Classifier

The SVC model has 100% f1-score for class 2 in comparison with the random forest model which has 0% f1-score for this class.

Students Performance Prediction Model Mini-Project Scientific Report

The other evaluation metrics that help us to understand where does the model confused to predict the actual class is a confusion matrix.

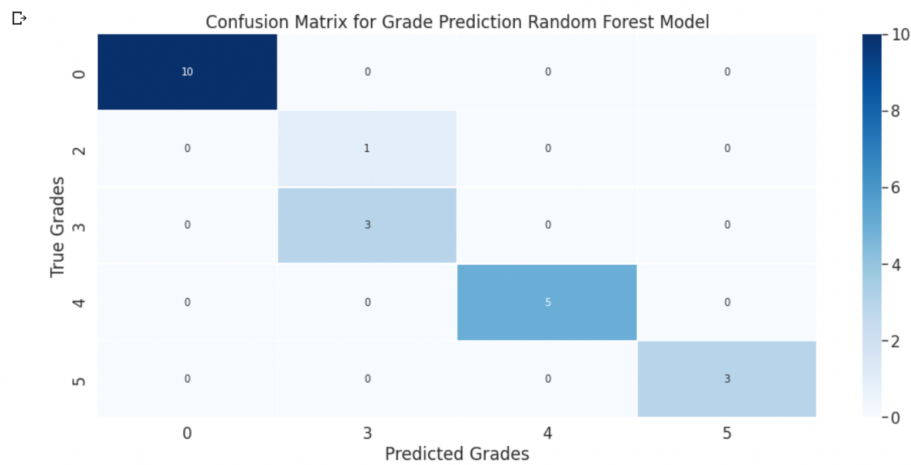


Figure 8: Confusion Matrix for Random Forest Model

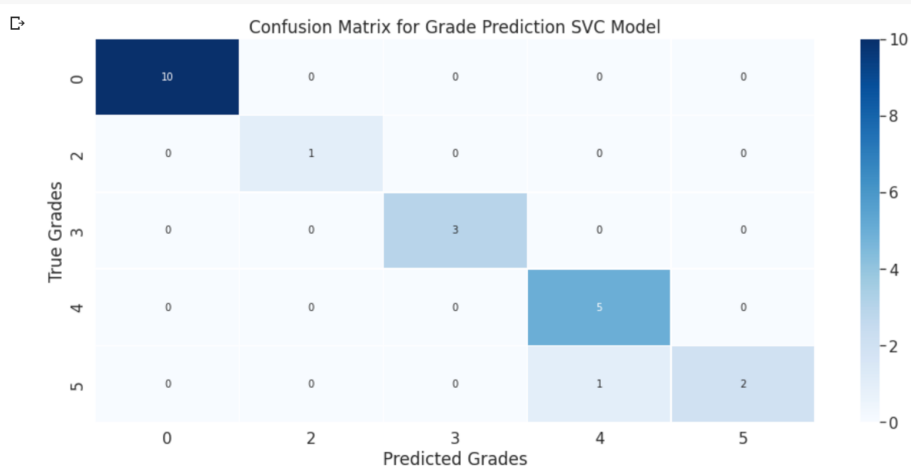


Figure 9: Confusion Matrix for SVC Model

As it is shown in the figures, the random forest model predicts an example of class 2 as class 3 whereas the SVC model predicts an example of class 5 as class 4.

As a whole both models perform well on their own different feature set. To make sure that whether they keep their performance on different validation dataset or not, cross-validation evaluation technique is applied. Cross-validation is used to estimate the skill of a machine learning model on unseen data. To implement this technique, there is a method called `cross_val_score` which will return a list of scores computed in different folds. The default k-fold number which 5 is used in our model and the random score model scores better than the svc model with accuracy of about 93% and 74% respectively.

Students Performance Prediction Model Mini-Project Scientific Report

The three most important features that the random forest classifier model uses to predict the final grade of student's grade are Week8_total, Week7_MP3, and Week5_MP2.

```
# The three important features
three_important_features = sorted(list(zip(rf_X_train, random_forest_clf.feature_importances_)), key=lambda tup: tup[1], reverse=True)[:3]
three_important_features
```

```
[('Week8_Total', 0.241951811440013),
 ('Week7_MP3', 0.13666694375187838),
 ('Week5_MP2', 0.13464600006629746)]
```

Figure 10: Three Top Important Features

6. Conclusion

To conclude, the aim of this mini project is to build a random forest classifier model and any other classification algorithm model to predict the final grade of students using a dataset collected from an online learning platform.

While doing the mini-project, selecting the second algorithm and choosing or engineering the feature set for the models were the major experience needed tasks. After researching and experimenting different approaches selecting a feature list based on a correlation relationship between the variables and the target class was chosen for the random forest classifier model. Knowing the data what we have and molding it to be suitable for the selected machine-learning algorithm is a required skill to have a better machine-learning model.

Understanding the techniques(including maths) used by the different machine-learning algorithms to predict the output and in which type of dataset they will perform better was the other difficult task to select the other algorithm other than the random forest classifier. By reading different blogs and sklearn documentation SVC classifier is chosen to be the second classifier algorithm.

Appendix:

- [Google colab link](#)