Data Science - Fall 2022 - Mini Project II

Aicha Moussaid
Aicha.moussaid@abo.fi
EDISS Master's Programme - Åbo Akademi University, Turku, Finland

Introduction

The most crucial aspect of learning activities is assessment. Students' accomplishments are determined depending on their final grade in a particular subject. Many grading systems have been devised to help academic processes run smoothly and effectively.

The individual grades of assignments and projects of a particular course can be used to forecast a student's final grade, as they all contribute to a total number of points acquired that then determine, on a 5-grade scale, the final grade to be assigned to the student's overall performance in their transcripts.

This project is being conducted in order to apply diverse classification and regression algorithms that will be used to predict student final grades, applied on data collected from a fully online, nine-week machine learning course housed on the online learning management system Moodle.

These models will be trained accordingly and have a generated performance evaluation and feature importance comprehension. The analysis of the results presented in this report will evaluate and compare the models and help us gain an idea on which algorithm could be the optimal choice for such a prediction problem, and what features are key to making or breaking the accuracy of the prediction.

Data Collection

The given dataset includes anonymized information on 107 enrolled students. Students' grades (from three mini projects, three quizzes, three peer reviews, and the final aggregate grade) were provided, as well as course logs. The deadlines for the three mini projects were in weeks 3, 5, and 8 of the course, while the deadlines of the quizzes were in weeks 2, 4, and 8.

The course logs enable you to follow student footprints on Moodle, beginning with the first time they accessed the course. It allows you to see what contents they have viewed and what activities they have engaged in, as well as the time and date they accessed it, and an overall log of assessments and posts related to the student activity. The dataset provided contains:

- Status0: course / lectures / content related (Course module viewed, Course viewed, Course activity completion updated, Course module instance list viewed, Content page viewed, Lesson started, Lesson resumed, Lesson restarted, Lesson ended)
- Status1: assignment related (Quiz attempt reviewed, Quiz attempt submitted, Quiz attempt summary viewed, Quiz attempt viewed, Quiz attempt started, Question answered, Question viewed, Submission re-assessed, Submission assessed, Submission updated, Submission created, Submission viewed)
- Status2: grade related (Grade user report viewed, Grade overview

report viewed, User graded, Grade deleted, User profile viewed, Recent activity viewed, User report viewed, Course user report viewed, Outline report viewed)

- Status3: forum related (Post updated, post created, Discussion created, some content has been posted, Discussion viewed)
- 9 grades (Week2_Quiz1, Week3_MP1, ... Week7_MP3)
- 36 logs (Week1_Stat0, Week1_Stat1, Week1_Stat2, Week1_Stat3, ... Week9_Stat0, Week9_Stat1, Week9_Stat2, Week9_Stat3)

Data Analysis

Before we give the data to the models we are adopting in this project, we first need to analyze it and have a "cleaner", more relevant data that can help our models have a respected performance. This is why we have to check if there are any missing values, study the features and their impact on our target, and decide if we should drop any of them or keep all. This step is important because it is crucial to cancel any noise created by irrelevant features that can only serve as a source of confusion to the models.

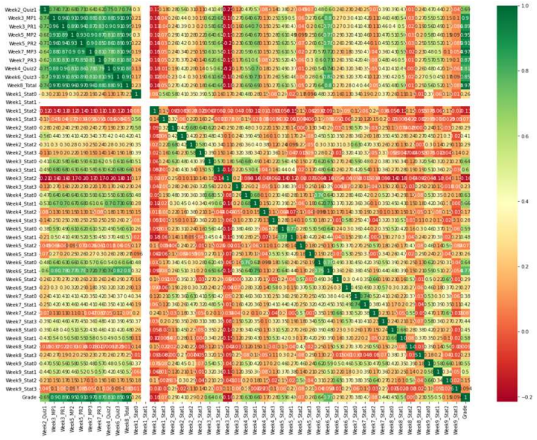


Figure 1. General Correlation Heatmap of the Dataset

Correlation heatmaps are graphical representations of the strength of correlations between numerical data. Correlation plots are used to determine which variables are related to one another and how strong this relationship is. Figure 1 illustrates this overview of our features and their relationship with our target "Grade". As shown, our focus goes to the positively correlated features that have a score of 0.4 and above, colored light to dark green.

For the implementation side, we found it necessary to drop different columns based on different correlation values for both models. Generally, we dropped the column of our target "Grade", "Week8_Total" because it will not allow the models to learn properly since they will rely heavily on it, and the white horizontal and vertical white blank created by "Week1_Stat1", which only contains zero values.

When it comes to Random Forest, only eight features were adopted for the training, with correlation scores higher than 0.8, because it will help the model to generalize more and have a reliable performance. Figure 2.a. shows the heatmap dedicated to random forest.

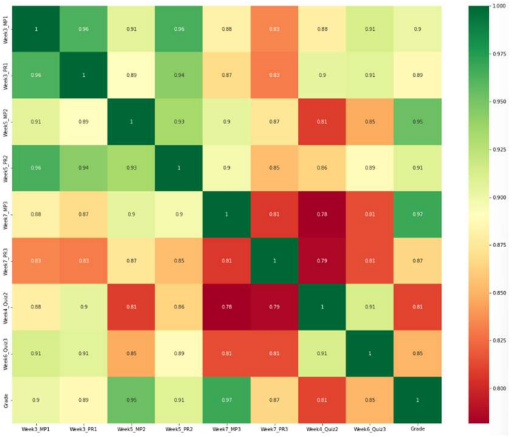


Figure 2.a Focused Correlation Heatmap for Random Forest

As for SVM, we decided to keep more features, fourteen to be precise, with correlation scores higher than 0.6. Figure 2.b. shows the heatmap dedicated to random forest.

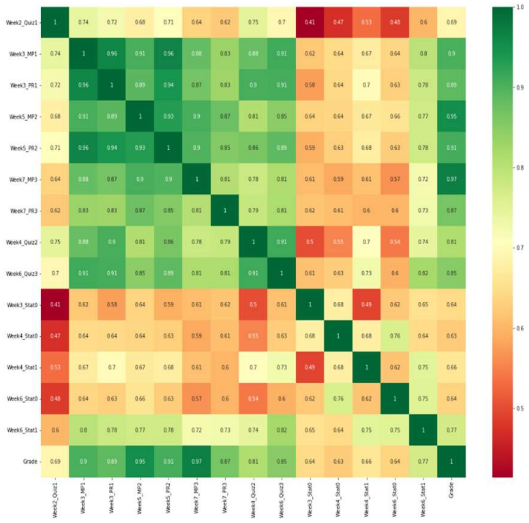


Figure 2.b Focused Correlation Heatmap for SVM

Now that the data is "cleaner" and ready to be fed to the models, the next step was to prepare the training and testing sets, and since only 107 rows are available to learn from, it was initially decided to allocate 90% (96 rows) of the dataset to the training and only 10% (11 rows) to the test set. This was later on changed because it impacted the correct prediction of the grade classes available, and after multiple tries and tweaks, it was decided to adopt a wider testing range of 30%.

Training and Evaluation

For the training section, the main two classifiers used were Random Forest Classifier and Support-vector machine (SVM) classifier. The prediction of the student grades was set to be a classification problem as it has target 5-scale points to have as a grade.

The accuracy of both models, as shown in Figure 3., have demonstrated an elevated level of prediction as they both scored 94% accuracy. We can also see how the features were prioritized as shown in Figure 4., where Random Forest relied heavily on Week7_MP3, Week5_MP2, and Week3_MP1.

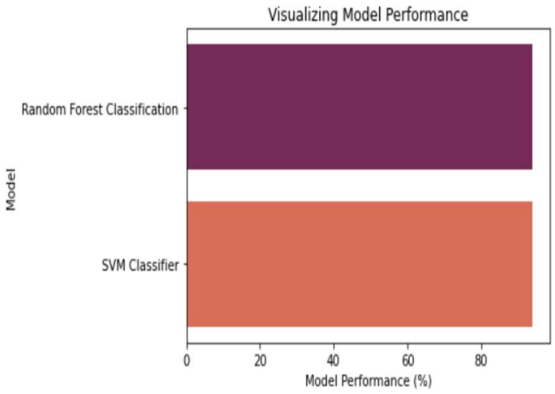


Figure 3. Model Performance

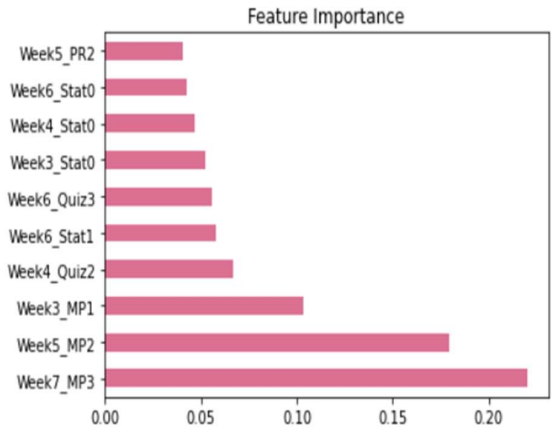
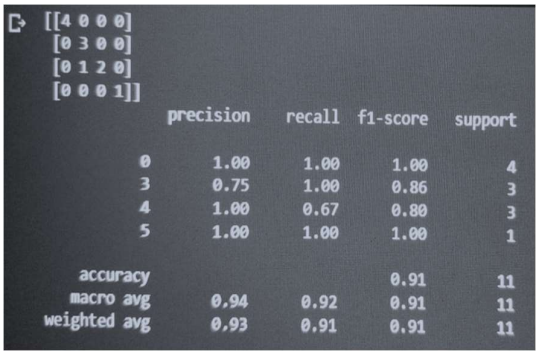


Figure 4. Model Feature Importance

The models performed an optimal performance, but there were some tweaks needed for to get the current accuracy. In fact, when studying the confusion matrix of Random Forest and SVM, two things popped up; the first being that there was no class of grade set to "1". This was revised in the dataset, and it was found that no student had a grade of "1", therefore, the model not knowing it is a possibility. The second thing is that the confusion matrix did not display grade of class "2", the necessary tweaking needed was to increase the testing set from 10% to 30%, which

surprisingly, displayed the class “2” and increased the accuracy to 94%, as shown in Figure 5. and Figure 6.

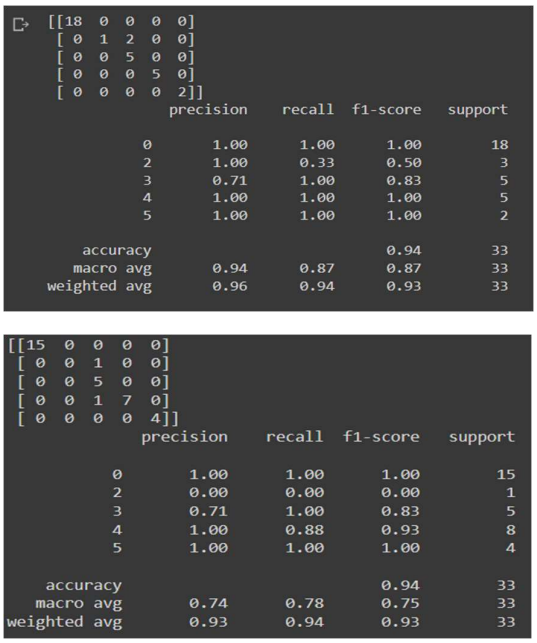


```
[[4 0 0 0]
 [0 3 0 0]
 [0 1 2 0]
 [0 0 0 1]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         4
           3       0.75      1.00      0.86         3
           4       1.00      0.67      0.80         3
           5       1.00      1.00      1.00         1

    accuracy                           0.91        11
   macro avg       0.94      0.92      0.91        11
weighted avg       0.93      0.91      0.91        11
```

Figure 5. Example of Confusion Matrix Before Tweaks



```
[[18  0  0  0  0]
 [ 0  1  2  0  0]
 [ 0  0  5  0  0]
 [ 0  0  0  5  0]
 [ 0  0  0  0  2]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        18
           2       1.00      0.33      0.50         3
           3       0.71      1.00      0.83         5
           4       1.00      1.00      1.00         5
           5       1.00      1.00      1.00         2

    accuracy                           0.94        33
   macro avg       0.94      0.87      0.87        33
weighted avg       0.96      0.94      0.93        33
```

```
[[15  0  0  0  0]
 [ 0  0  1  0  0]
 [ 0  0  5  0  0]
 [ 0  0  1  7  0]
 [ 0  0  0  0  4]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        15
           2       0.00      0.00      0.00         1
           3       0.71      1.00      0.83         5
           4       1.00      0.88      0.93         8
           5       1.00      1.00      1.00         4

    accuracy                           0.94        33
   macro avg       0.74      0.78      0.75        33
weighted avg       0.93      0.94      0.93        33
```

Figure 5. Confusion Matrix for SVM and Random Forest after Tweaks

As the performance of both models is on the same level, and there are no other parameters to consider when evaluating them against each other (like execution time, memory taken, or resources needed), it is a bit complex to compare both of them and decide which would be more suitable. It can of course be further investigated with more demanding datasets, which can put to test the real performance of these models given a specific problem.

Conclusion

This project brought up to the table the interesting side of training models and its complex steps. It made it clear that, first of all, managing the data firsthand and processing it before feeding it to the models can make or break the performance.

Second of all, assigning the training and testing is differing from a model to another. In other words, the tweaking of the models and their related aspects has to be closely monitored to watch its effect on the accuracy of our predictions.

Lastly, choosing the correct models to use depends solely on the type of dataset you have, how large it is, the type of problem you are handling, and analyzing the strengths of each available model, which can help us decide the optimal model that can, both, save us time and allow us the best prediction.

Working on this situation paved way to extensions of such a problem that we could apply further using machine learning, as we could for example try predicting the student grades based on previous semesters, or build a solid prediction system that can serve as an educational monitoring tool, where educational parties can predict the student’s final grade based on his performance in a set time slot, which can help approaching students, figuring out their impediments and assisting them in getting back on a track that would allow them to pass their courses.