Data Science - Fall 2022 - Mini Project II

Alex Montoya Franco
alex.montoyafranco@abo.fi
EDISS Master's Programme - Åbo Akademi University, Turku, Finland

1 Introduction

In the era of intelligence, the rapid development of information technology has brought a profound impact on all aspects of social life, including education [1].

Technologies such as, e-learning, mobile learning, educational data mining, learning analytics, and artificial intelligence have the potential to enable professors to scientifically analyze students' performance and learning characteristics, so as to carry out differentiated teaching, while also providing students diverse mobile technologies, adaptive content, and learning opportunities for them to independently select learning content, learning methods, and meet their own personalized learning needs [1].

In recent years, these technologies have converged into educational resources such as, online learning platforms, OER (Open Educational Resources) and MOOCs (Massive Open Online Courses).

However, despite its several benefits, including their widely popularity and adoption, retaining students in online learning platforms is challenging [2].

By the time instructors realize that a student is not motivated by a course, is not engaged or does not feel supported during their learning process, feedback is already overdue.

Feedback is a critical element of student-instructor interaction. However, with student to teacher ratios growing rapidly, challenges arise for instructors to provide quality and timely feedback to individual students [3].

Accordingly, how can we provide this feedback on time? How do we make sure that students are engaged and are in a successful path to complete their studies?

Student performance prediction aims to leverage student-related information to predict their future academic outcomes, which may be beneficial to numerous educational applications, such as personalized teaching and academic early warning. It can help instructors provide in-time intervention and personalized guidance in education, especially for those at-risk students [4].

With this report, we seek to predict students' performance by analyzing data collected from a fully online nine-week-long course on machine learning, hosted on the online learning management system Moodle.

The goal of this project is to use two different supervised learning techniques, a random forest classifier and a KNN classifier, to predict students' final grade in an online course.

With this project we intend to answer the following questions:

1. Can we predict the final grade of a student based only on its interactions with a learning platform such as moodle?
2. Are "active" students more likely to succeed in a given course?
3. Which interactions are the most relevant? Lessons? Submissions? Forum?

With this report we would like to help instructors in understanding student behaviors and how they affect their learning outcomes. In this way, instructors can take the required actions to help students in their learning processes. Thus, improving motivation, engagement and a more personalized learning experience.

This report is organized as follows: (1) Data processing, which includes how data is collected and how we perform feature engineering to keep the relevant features for further analysis, (2) Data analysis, which includes dividing the data in an appropriate proportion to train and test the models, training our two classifiers, analyzing their performance, comparing their results, and choosing the most important features. (3) Finally, conclusions are presented describing the scientific bottlenecks of this project and how we overcome them.

2 Data Processing

2.1 Data Collection

The data was collected from a fully online nine-week-long course on machine learning, hosted on the online learning management system Moodle.

The dataset contained anonymized information relating to 107 enrolled students. The data included students' grades (from 3 mini projects, 3 quizzes and 3 peer reviews and the final overall grade) as well as the course logs (9 grades and 36 logs).

- MP: Mini Projects
- PR: Peer Reviews
- Status 0: course / lectures / content related
- Status 1: assignment related
- Status 2: grade related
- Status 3: forum related

Category	Columns
Mini Projects	'Week3_MP1' 'Week5_MP2' 'Week7_MP3'
Quizzes	'Week2_Quiz1' 'Week4_Quiz2' 'Week6_Quiz3'
Peer Reviews	'Week3_PR1' 'Week5_PR2' 'Week7_PR3'
Week 1 Logs	'Week1_Stat0' 'Week1_Stat1' 'Week1_Stat2' 'Week1_Stat3'
Week 2 Logs	'Week2_Stat0' 'Week2_Stat1' 'Week2_Stat2' 'Week2_Stat3'
Week 3 Logs	'Week3_Stat0' 'Week3_Stat1' 'Week3_Stat2' 'Week3_Stat3'
Week 4 Logs	'Week4_Stat0' 'Week4_Stat1' 'Week4_Stat2' 'Week4_Stat3'
Week 5 Logs	'Week5_Stat0' 'Week5_Stat1' 'Week5_Stat2' 'Week5_Stat3'
Week 6 Logs	'Week6_Stat0' 'Week6_Stat1' 'Week6_Stat2' 'Week6_Stat3'
Week 7 Logs	'Week7_Stat0' 'Week7_Stat1' 'Week7_Stat2' 'Week7_Stat3'
Week 8 Logs	'Week8_Stat0' 'Week8_Stat1' 'Week8_Stat2' 'Week8_Stat3'
Week 9 Logs	'Week9_Stat0' 'Week9_Stat1' 'Week9_Stat2' 'Week9_Stat3'
Final Grade	'Week8_Total' 'Grade'

2.2 Feature Engineering

The dataset contains 107 rows and 48 columns. For a dataset with 107 samples, having almost half that number of features seems like a dimensionality issue. Having this into account, we first filter these features to have the most relevant ones for future analysis.

We first explore the relevance of the grades for our future predictions. At first glance, we argue that it is not correct to predict a final grade based on all the grades in a course, since this does not need a model prediction but a rather simple mathematical expression. However, we want to explore these features first to understand the data and find possible new details about the grades in the dataset. This exploration ultimately helped in refining the questions that guide this project.

After analyzing Grade-related features, we assume that if a student gets the max value in every Mini project, Quiz and Peer Review, his/her 'Week8_Total' grade would be 100 and therefore the final 'Grade' will be 5. This show us that grades have a direct correlation to the final grade but are not relevant predictors unless we choose a subset of them. For example, the grades for the first N weeks.

Let's analyze the correlation matrix among all the grades to see if we find any highlights that might guide us to keep a number of these features.

The correlation matrix shown in **Figure 1** indicates that grades are generally highly correlated between each other, which means that they are redundant for our analysis.

We don't consider keeping any of these features for further analysis since Quizzes and Peer Reviews are redundant and have a small contribution to the final grade, while mini projects would be the strongest predictors, giving a false impression that our model is successfully predicting students performance.

From now on, we will focus on the course logs a.k.a stats to see if they could become interesting predictors for the students' final grade. We hope to identify if they are valid features to predict students' final grades. And analyze if they provide interesting and relevant insights into what makes a successful student, successful.

Although we have removed 10 columns related to students' grades. We still have 38 columns in our dataset. With this

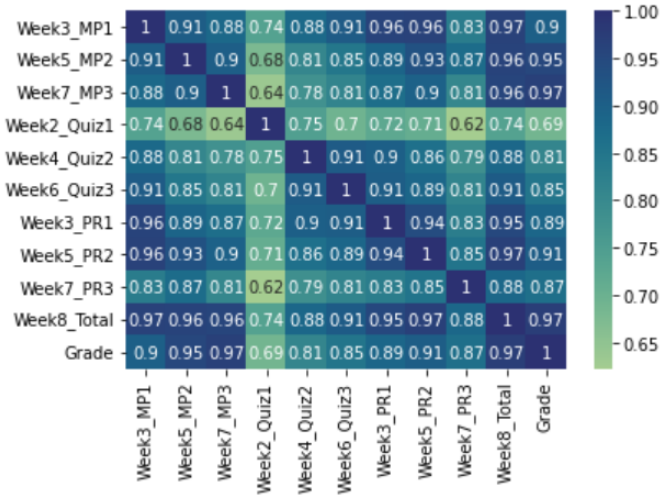

Figure 1: **Correlation matrix of Grade-related features**

in mind we will proceed with 3 approaches in order to realize which one will give us better results when applying our models and which provide more comprehensive answers to our research questions.

- **Grouping data by week.** Is there a specific week where students' chance to pass or fail the course increases? (Combining this with the information of when are the assignments' deadlines could help in balancing out activities - Are students more active or less active in weeks with deadlines?).
- **Grouping data by stat.** Is a student more successful in the course if it participates more in the forum? if it checks more the lessons? (Is there a specific stat that relates to successful students?
- **Keeping all stats features.** This might provide more comprehensive answers since it would directly lead us to a specific week and stat. For example, is early participation in forums an initial sign of students success? (important features in here could serve as a new set of features for predicting students' grades).

We have already removed grade columns based on Domain Knowledge, and we now analyze if we have missing values and low variance in stat-related columns in order to reduce the feature space more.

No missing values were identified. However, the Variance analysis tell us that the column 'Week1_Stat1' has a very low variance i.e., at least 90% of the values in this column are the same. For our dataset that means 96 out of 107 are just the same value. The variance analysis was also run for 80% and 70% and 'Week1_Stat1' was still the only column with low variance showing a clear distinction from other columns.

We finally concluded that there is not only a low variance in this column, but actually is just a column of zeros. This column won't help the model to find any patterns so it was removed.

We don't consider more strategies to directly remove current features given that we want to keep certain balance between weeks and stats. Let's then continue with our 3 proposed approaches: (1) Group by week, (2) Group by stat, and (3) Keep all stat features.

For the dataframe grouped by week, we just aggregate all the stats per week so that we might get an idea of crucial weeks for the success or failure of students.

This subset of data now has 9 columns which allows a more clear analysis of the features available and also simplifies model training in a way that could help produce more accurate results.

For the dataframe grouped by stats, we just aggregate all the weeks per stat, which gives a much smaller feature space, and hopefully provides different insights about our data.

3 Data Analysis

3.1 Training & Test Dataset

Given the size of the dataset, we consider that we need the most data points possible to train the models. Therefore, we choose a split of 80/20 (80% training - 20% testing). We also tried a different split of the data (70/30) while training the models in order to see if there will be any effect on the predictions. There was not a noticeable effect, so the 80/20 ratio was kept.

Listing 1: Split dataset into train and test with sklearn

```
from sklearn.model_selection import train_test_split

# Split for dataframe grouped by week
X_train_weeks, X_test_weeks, y_train_weeks, y_test_weeks =
    train_test_split (
        weeks_agg_df,
        students_df.Grade,
        test_size=0.20,
        shuffle=True,
        random_state=42
    )
```

3.2 Models Training

We predict the students' final grade using (1) a Random Forest Classifier and (2) a KNN Classifier.

3.2.1 Random Forest

Listing 2: Training and Testing Random Forest

```
from sklearn.ensemble import RandomForestClassifier
from sklearn import metrics

# Random Forest with 100 trees
clf_weeks_1 = RandomForestClassifier(n_estimators=100)

# Training the model in our first set of data (grouped by week)
clf_weeks_1.fit(X_train_weeks,y_train_weeks)

# Use the model to predict the test data
y_pred_weeks_1 = clf_weeks_1.predict(X_test_weeks)

# Model Accuracy (how often is the classifier correct?)
# How accurate does the model predict the students final grade
print(metrics.accuracy_score(y_test_weeks, y_pred_weeks_1))

> Accuracy: 0.6363636363636364
```

This accuracy tell us that our model is not very good at classifying new data. After training the model multiple times, ranging the estimators between 50 and 150 (sometimes even 200 or 300) we can conclude that the model is not getting any better than 63% (sometimes 68%) at predicting new values.

Training the models with the other two approaches discussed during our feature engineering process did not produce better results.

We can conclude at this point that the model does not get better than 68% accuracy with all the course logs. Staying most of the trainings between 54% and 63%. Since this is one of the best classifiers methods out there we start getting an idea of what can be answered with the available data and the chosen features.

Thus, can we predict the final grade of a student based only on its interactions with a learning platform such as moodle?

Answer: No, the model still has a lot of uncertainty with the data available. With the current accuracy is basically as if we were rolling a dice to decide students' performance. Hence, possible actions like feedback or interventions would probably not be correctly targeted.

Given these results, question 2 cannot be answered either with the current state of our data and models. Since we cannot reliably predict the final grade, we therefore cannot guarantee that any type of student is more likely to pass or fail the course with certain behavior.

Adding some of the grades would clearly improve our model's performance. However, we still consider that the grades are not appropriate features for this problem, even though they will clearly help with prediction, we think this might be misleading. Other papers have explored the use of past grades to predict a final grade, but this is mostly related to state-level exams or past courses which can give an idea of an student performance in the past.

To answer the question: Are "active" students more likely to succeed in a given course? we need more data to help the model differentiating between active and not active students. For example, not only having statistics about how many times a student sees a content, but perhaps, how much time does he/she spends on it? - In terms of forum, not only posts

created/updated but perhaps how are the ratings for those posts?, among others.

As for question 3, we believe it could still be answered in part by highlighting the most relevant features for the current model using the random forest function designed for it. See section 3.4 Important features for this.

3.2.2 K-Nearest Neighbors (KNN)

Listing 3: Training and Testing KNN

```
from sklearn.neighbors import KNeighborsClassifier

# See the distinct classes in order to define the neighbours
students_df.Grade.unique()
> array([4, 3, 2, 0, 5])

knn_weeks_model = KNeighborsClassifier(n_neighbors=5)

knn_weeks_model.fit(X_train_weeks, y_train_weeks)

y_pred_knn_weeks = knn_weeks_model.predict(X_test_weeks)

print(metrics.accuracy_score(y_test_weeks, y_pred_knn_weeks))

> Accuracy: 0.5454545454545454
```

There is not an improvement in the accuracy when using KNN. It actually got a smaller worse result for our last dataframe which shows a small advantage by the random forest classifier, showing that having a ensemble method can clearly improve our model performance. However, this also shows that no matter how good your model is, your data is the key. We can now conclude that the available data and the chosen features do not describe well the target. We do not deny that perhaps another set of features might have produced better results. So we reserve the right to judge the quality of the data until further analysis.

Neither of the models predict the students' final grade with high confidence. With that in mind Random Forest Classifier is the clear winner out of this. Given its way of working, it produces slighly different results when repeating the training process with the same data, ranging between 54% - 68% accuracy. On the other hand, KNN never got a better accuracy than 54% and it actually decreased for one of the scenarios until simply 50% (missing the right answer half the time!).

The most direct comparison which we can make right now is how they both work. KNN is a simple type of model which uses the distances between data points to form groups and classify data, while Random Forest uses an esemble of decision trees which can produce more accurate results by protecting trees from the errors of other estimators.

For this setting, we believe that the Random Forest classifier performed better since it had the possibility to try different configurations to train the model (one per estimator), with data small in number of features and number of samples.

We also acknowledge that perhaps KNN is not the most fair comparison companion for Random Forest. Perhaps another ensemble method should be used to see what kind of results it might produce.

3.3 Performance Evaluation

Models Performance has been partially cover in the previous section in order to have a cohesive order in training a model and then checking its accuracy right away. We add new performance metrics such as, Confusion Matrix, Precision, Recall, F1-score, and also discuss what could be possible changes to improve our models. Furthermore, we also visualize our models performance to get a better understanding of where the model performed well and where it did not.

Although the accuracy gives us a first hint at how our models are doing, the accuracy is not a great measure of classifier performance when the classes are imbalanced. We need more information to understand how well the model really performed for each class. Did it perform equally well for each class? Were there any pairs of classes it found especially hard to distinguish? [1].

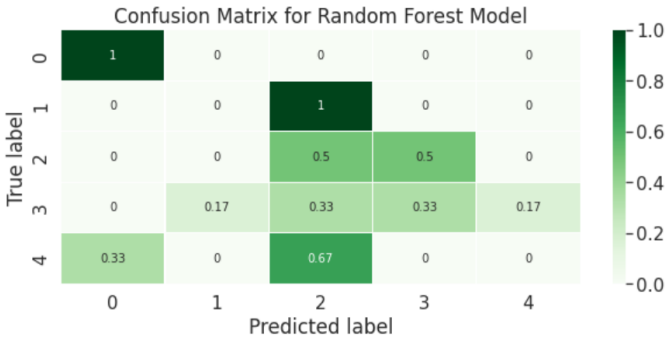

Figure 2: **Confusion matrix for random forest classifier**

Figure 2 shows a confusion matrix, which is a way to express how many of a classifier's predictions were correct, and when incorrect, where the classifier got confused. Values on the diagonal represent the number (or percent, in a normalized confusion matrix) of times where the predicted label matches the true label. Values in the other cells represent instances where the classifier mislabeled an observation. This is a convenient way to spot areas where the model may need a little extra training [2].

Now it's easier to see that our classifier struggled at predicting multiple labels. For example, Class "2", which was mislabeled half the time as class "3".

Could our model be improved?

Yes, it could be. Our model, perhaps any model, is as good as the data we put into it. With that in mind, possible improvements in our dataset and models are:

- Add new features which provide a more clear representation of students interactions with the platform. For example, how much time do they spend in certain content? how are their forum posts rated?
- Combine the final grades into two classes (pass or fail), could lead us to create a binary classifier which could

[1] https://medium.com/analytics-vidhya/evaluating-a-random-forest-model-9d165595ad56

[2] https://medium.com/analytics-vidhya/evaluating-a-random-forest-model-9d165595ad56

focus on those two classes and find new relationships in the data.

- Pay more attention to the balance of the classes when splitting the data between training and testing. Using 'stratify=y' tells train_test_split to make sure that the training and test datasets contain examples of each class in the same proportions as in the original dataset. This is especially important to do because of how imbalanced the classes are. A random split could easily end up with all examples of the smallest class in the test set and none in the training set, and then the model would be unable to identify that class [3].
- Increase the size of the dataset, by just adding more samples from the learning platform or using a technique for data augmentation.
- For the random forest classifier, pay closer attention to the model parameters beyond the number of estimators, perhaps the model itself can deal with unbalanced classes.
- For the KNN model, pay close attention to the distance measures. These distances are used to calculate the similarity between data points. A good distance metric boosts the performance of a model. Explore different measure for our KNN model might provide better accuracy or at least give better competition to the random forest model.

3.4 Important features

Since we have concluded that our models do not accurately predict students' performance, we consider the following features are not the most important for a correct prediction, but instead the most relevant for our current model. What our models consider better to predict a grade, which might clearly be inaccurate given our results.

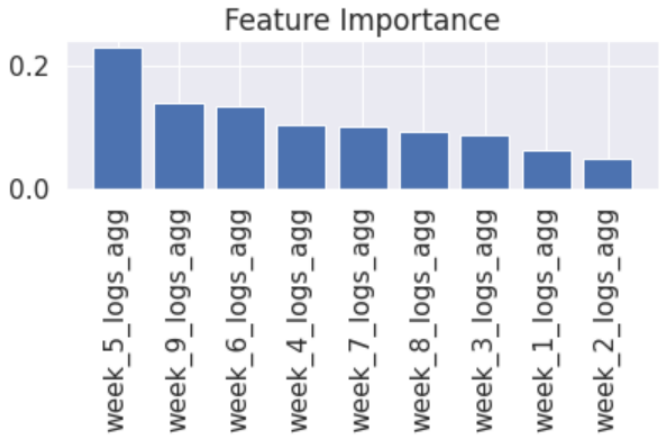

Figure 3: **Important features for data grouped by week**

The three most relevant features are the aggregated logs for weeks 5, week 9 and week 6 which might be an indication on how interactions with the platform increases towards the middle and end of the course. Further interpretations are discussed in our Conclusion section.

[3] https://medium.com/analytics-vidhya/evaluating-a-random-forest-model-9d165595ad56

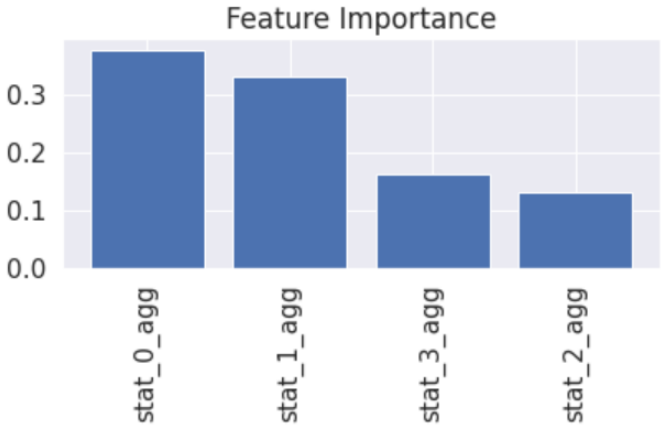

Figure 4: **Important features for data grouped by stats**

Stat 0 corresponds to the modules and lessons while Stat 1 is related to assignments (including quizzes attempts and submissions), and Stat 3 to forum posts.

We can interpret from this that the most relevant interactions that students have with the platform are checking learning content and submitting assignments. Stat 3 is forum related and it could hint at the value of checking the forum to receive help or offer help during the course.

The "All logs dataframe" did not provide relevant information in terms of feature importance. Given the amount of features the importance was dispersed among the features and did not show a clear trend. Nevertheless, we explore some of these results in the next section.

4 Conclusion

Any attempt to use data as the fuel to produce knowledge, is not as good as the quality of the motor (a.k.a the model), but as good as the data itself.

During this project three questions were asked, for which not satisfying answers were obtained. At first, we consider that the most challenging part of this project was going to be the training and testing of the machine learning models given the small previous experience in the topic. However, by the end of this report, we conclude that these type of models have already been figure out by the industry, in most instances they are now just methods where you input the data and get your results.

In terms of our first project question, we conclude that we cannot predict the grade of a student based only on its interactions with the learning platform, at least considering the data available. The uncertainty is still high and more data both in terms of samples and features is necessary. This was a surprising conclusion for us since reducing the feature space was one of our first priorities since the beginning. However, we do consider that in order to improve our models and have a chance to answer this question, more data is needed. How much time do students spend reviewing the learning content? How many attempts do they used for given exams and projects? Do students get better grades by submitting more times?

We believe our second question follows the uncertainty of our first so we don't venture to get any conclusions in here.

In terms of our third question, the feature importance visualizations for our two representations of the data (grouped by week and stat) give us some interesting insights. While at first our question just considered the relevance of the interactions by themselves (lessons, submissions, forum, etc), we conclude that there is not only relevance in these stats, but in their relation to specific weeks (which consequently relates to specific content and submissions).

Figure 3 tell us that week 5 hold more relevance for our models prediction, which we *could* relate to students being more active during that week. The data description tells us that the deadline for the three mini projects fell within weeks 3, 5 and 8 of the course. So we could assume that Mini project 2 was probably scheduled for week 5 and made students interact more with the platform. Was it a more challenging project? Was this a decisive point for many students in the course?

Furthermore, we can interpret from Figure 4 that the most relevant interactions that students have with the platform are submitting assignments, reviewing learning content and participating in the forum.

Finally, we understand that asking the right questions and finding the right answers through data-driven inquiry is an iterative process. Not because we did not find the answers we wanted in here, means that the data holds no value, new iterations of this process with refined features, new aggregates, perhaps just more samples might provide a big difference for future models.

References

[1] Y. Lian, "Smart education: Education reform in the age of intelligence," in *2021 5th International Conference on Education and E-Learning*, ICEEL 2021, (New York, NY, USA), p. 41–45, Association for Computing Machinery, 2021.

[2] R. Panigrahi, "Online learning: Improving the learning outcomes," in *Proceedings of the 2017 ACM SIGMIS Conference on Computers and People Research*, SIGMIS-CPR '17, (New York, NY, USA), p. 203–204, Association for Computing Machinery, 2017.

[3] S. Nicoll, K. Douglas, and C. Brinton, "Giving feedback on feedback: An assessment of grader feedback construction on student performance," in *LAK22: 12th International Learning Analytics and Knowledge Conference*, LAK22, (New York, NY, USA), p. 239–249, Association for Computing Machinery, 2022.

[4] J. Zong, C. Cui, Y. Ma, L. Yao, M. Chen, and Y. Yin, "Behavior-driven student performance prediction with tri-branch convolutional neural network," in *Proceedings of the 29th ACM International Conference on Information Knowledge Management*, CIKM '20, (New York, NY, USA), p. 2353–2356, Association for Computing Machinery, 2020.