# Data Science – Mini Project 2

## Introduction

Data is used more and more in all sorts of industries to improve services, and to increase business value. One way data is used is to run e.g. customer data through a machine learning algorithm, with the purpose of getting insights into customer behavior and how the business can be improved upon.

Education is a field which can benefit from data-driven decision making and insights. Universities and schools offer education through courses, which usually contain lectures and different kinds of exercise assignments, through which a student is evaluated on what they have learned on the course, and a final assessment is given through a grade (often 0-5).

In this project, set of data is given, which represents students' performance on a particular course. The data is processed and ran through a machine learning algorithm, with the goal to predict future students' grades based on performance and activity metrics during the course. This can help identify which parts of the course are the most important in determining a student's success, and which parts need to be improved on, if it seems that a student performing a task in that particular part impacts their success negatively.

## Method and process

For this project, a Random Forest Classifier is used to predict the grades of students. The first step is to prepare the data set so that it can be effectively used for the machine learning model (see Part 1 in the code). The data has many features, including scores on assignments (quizzes, mini projects, peer reviews), activity metrics (watching lecture videos, posting on the forum, reviewing grades) and more. Right away it is clear that several of the columns have no or next to no impact on the final grade, since there is almost no activity recorded in those columns. They are removed from the data set.

Furthermore, several students have received a failing grade (0). It seems that most of the failing grades have not participated in the course at all, or done a little and given up early. A few seem to have legitimately tried but failed nonetheless. For this project, we only consider only passing grades, as an overwhelming majority of the failing grades belong in the group that did not participate as intended. As such, all rows with a grade of 0 are removed from the data set.

Next, the data is divided into training and test data (Part 2 in the code). A ratio of 75/25 is used for the training and test set. The data set is rather small, so it has to be ensured that the absolute amount of samples in each set is sufficient, both for properly training the classifier, but also to see whether the model performs well. Then the model is trained, and an accuracy score is calculated from the test data.

On the first run, the model performs poorly; it has an accuracy of around 60%. On closer examination, it turns out that some hyperparameter default values are not optimal. By using GridSearchCV to test different hyperparameter values for number of estimators in the classifier

and the maximum number of features it considers, better hyperparameter values are found and the accuracy seems to improve (part 3 in the code).

## Conclusion

By running the code model.feature_importances_, a list is generated which tells what are the most important features the classifier considers when predicting a grade. From the list it is seen that several features are ignored, and by far the most important value is Week 8 Total. By selecting the most important features, a new model is created on only those features. The old model and the new model are run 1000 times, and a mean accuracy score is calculated. The old model score receives a mean accuracy of around 95%, while the new model receives a score of around 99% (part 4 in the code).

However, this score does not quite fulfill the purpose of the project, which was to find early indicators of a student's success. By week 8 the course is nearly over and the total score obviously correlates strongly to the final grade. To find out if there are earlier indicators for predicting reliably a student's grade, the Week 8 Total is dropped from the original data frame, and a new model is trained and scored against this new frame 1000 times. This time the model receives a mean accuracy score of around 95%, and the overwhelmingly most important feature the classifier considers is Week1_Stat0, which means that students who watch lectures and view course content in the first week are more likely to succeed later in the course (part 5 in the code).

There are some bottlenecks that need to be taken into account when considering the validity of this result. Firstly, the data set is rather small. For a large data set, there could be a lot more variation in all of the features, and the machine learning model might consider different features as important in its prediction. Second, since no students with a failing grade are considered in the model, it is also possible that some other feature would jump out as of particular interest to the classifier, if it turns out the student participates actively in the Week 1 lectures but still receives a poor grade.