# Machine Learning Coursework - OULAD Analysis

mbtj48

## 1 Data Gathering and Analysis

Machine learning and data gathering are paramount for modern, cutting edge technologies. Thus we have been tasked to develop two machine learning models to predict final grades from the OULAD.

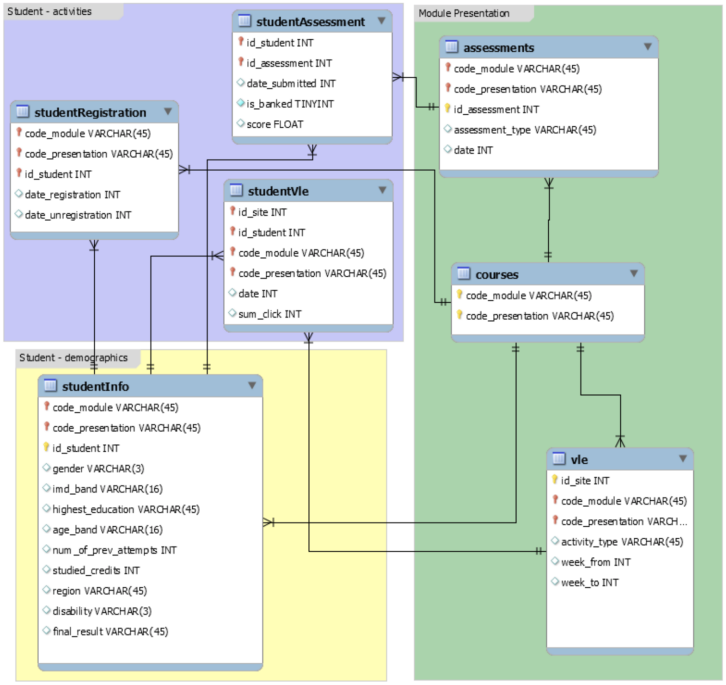


Figure 1: Dataset Schema

Firstly, I noticed useful features such as the score in the studentAssessment table, and sum-click in the studentVle table. Therefore I started by grouping the sum-click and score features, finding the net clicks within the portal for a given student through the year and their average mark. I expected these features to show a positive correlation because higher scores and grades generally correlate with high effort [tab 1]. Then, I plotted the data and noticed that a logistic regression model should perform highly. I added more data to my model; intending to use as much data as possible to aid the model in finding patterns: ie calculating how many days early they submitted coursework and their summative and formative (weight = 0) marks. As the student would be more prepared and committed, I expected a positive correlation from this data.

My interpretation of the data evolved as, I included a wide range of the available data, thus I started to interpret the data differently: including the mean, median, mean absolute deviation, standard deviation and variance for different data in the schema. I then produced a correlation heatmap [fig 11], plus the sorted numerical correlations [tab 1].

| Feature | Correlation |
|---|---|
| daysEarlystdScore | −0.259014 |
| studied-credits | −0.176016 |
| region-Wales | 0.008382 |
| age-band | 0.068551 |
| score | 0.317339 |
| sum-click | 0.376107 |
| totalCoursework | 0.427175 |
| summativeAgainstCredits | 0.490646 |

Table 1: Correlations

Surprisingly, age-band shows a weak correlation; in theory, you would expect a mild negative correlation. However, this could be due to limited data (3 unique ranges - [fig 3]). To improve this correlation, I would need more spread out ranges or use integers instead of the ranges.

After data gathering, I preprocessed the data with an imputer and scaler. The imputer changes all NA values to the median of that feature. While the scaler, normalises features to be within $0-1$, this prevents feature domination with large ranges and makes the features unit dependent. Further, I exchanged region, code module and code presentation to columned data by one-hot encoding those categories.

## 2 Model Selection

The following phase involved selecting models. Here, I split the data into train and test sets with a 75/25 split; then tested a variety of models and compared how they performed in cross-validation on the training data. Generally, there was a wide performance range, with classifiers generally doing better than regressors. See table 3 for further information. I decided to pick one regressor and one classifier to explore: Logistic Regression and Random Forest Classifier.

## 3 Model A - Logistic Regression

Moving on to hyperparameter tuning: For this model, I decided to use a grid search to validate the best combination within the domain. I explored a logarithmic range of the C parameter; until after testing, I reached an optimal range of 950-1100. It also checked tolerance values around the default, ending with 0.0015. The other set of combinations check the same values of C and adjust the solver and penalty used. Finally, I removed the second set of combinations as it did not help to increase performance.

## 4 Model B - Random Forest Classifier Appendix

Initially, I used Random Search to tune the classifier. This generates parameters to cross-validate the model. I decided to check the number of estimators, the maximum depth, the minimum number of samples and the minimum number of samples required to split an internal node. I moved on to use Bayesian optimisation to optimise this search problem. This uses the previous iterations to strategically select the next best parameters from the search space to minimise the loss function. I defined the loss as $3 - \overline{acc} - \overline{acc_{bal}} - \overline{f1_{weighted}}$; this, therefore, seeks to reduce the primary metrics for a classification problem. Next, I removed the unimportant attributes from the forest's feature importances [fig 6] and tuned the model again. I noticed the number of estimators showed little correlation for the model improving the loss function [fig 9] and this second tune slightly improved the model [figs $7 \rightarrow 10$].

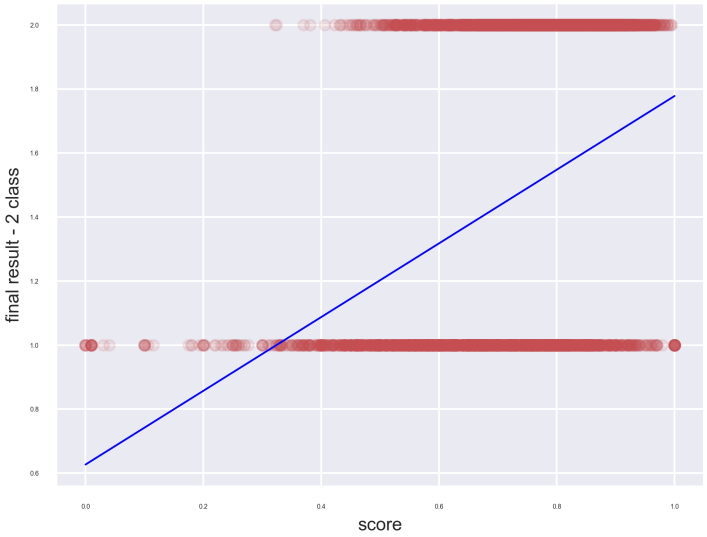


Figure 2: Linear Regression (2 class) against score

## 5 Conclusion

| **Model** | **Logistic** | | **Random Forest** | |
|---|---|---|---|---|
| **Classes** | **2** | **4** | **2** | **4** |
| Explained Var | 0.697 | 0.664 | 0.823 | 0.729 |
| RMSE | 0.275 | 0.585 | 0.213 | 0.523 |
| MAE | 0.076 | 0.300 | 0.045 | 0.263 |
| r2 Score | 0.697 | 0.662 | 0.819 | 0.729 |
| f1 Score (weighted) (Recall and Precision) | 0.924 | 0.706 | 0.955 | 0.739 |
| Accuracy | 0.924 | 0.721 | 0.955 | 0.742 |

Table 2: Metrics of Final Models

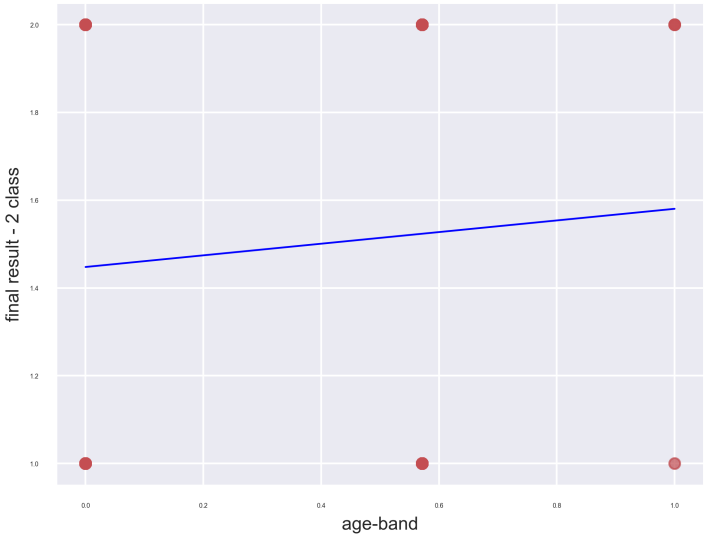


Figure 3: Linear Regression (2 class) against age-band

The logistic regression model finished with an accuracy of 0.721%, while the random forest classifier finished with 0.742%. Although notably, the weighted average of the f1 score for both models are lower; which is likely a better metric to measure, as it reduces metric skew on imbalanced data. Furthermore, upon analysing the classification report, classes with low balance have lower performance due to the data imbalance. The confusion matrices [tables 4,5,6,7] showed that "close" classes such as withdrawn and fail and pass and distinction are easily mistaken. Finally, note the two-class metrics (these merge withdrawn and fail & pass and distinction, as these are subclasses of the primary labels) which are generally high performing. The random forest 2-class model would be good to implement to determine whether a given student is likely to fail, due to the high accuracy and f1 score [2].

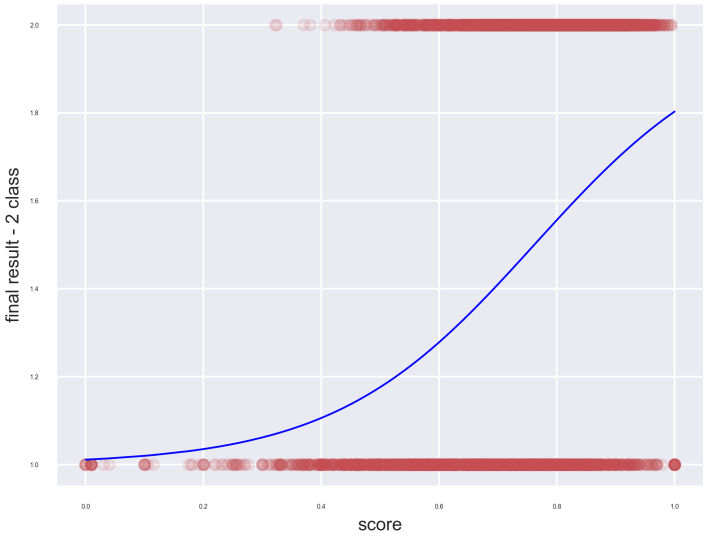


Figure 4: Logistic Regression (2 class) against score

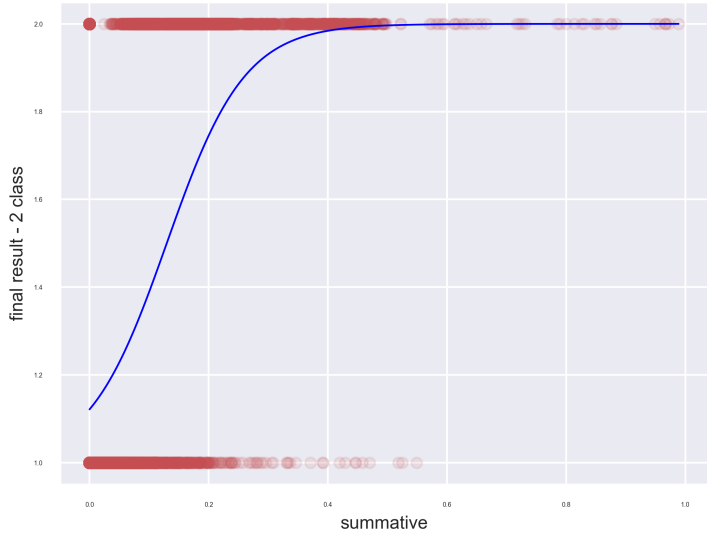


Figure 5: Logistic Regression (2 class) against summative

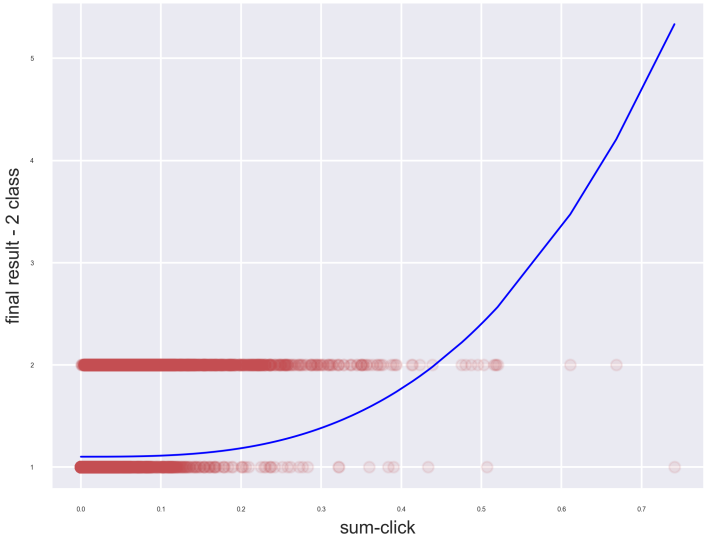


Figure 8: SVR-Poly-Kernel (2 class) against sum-click

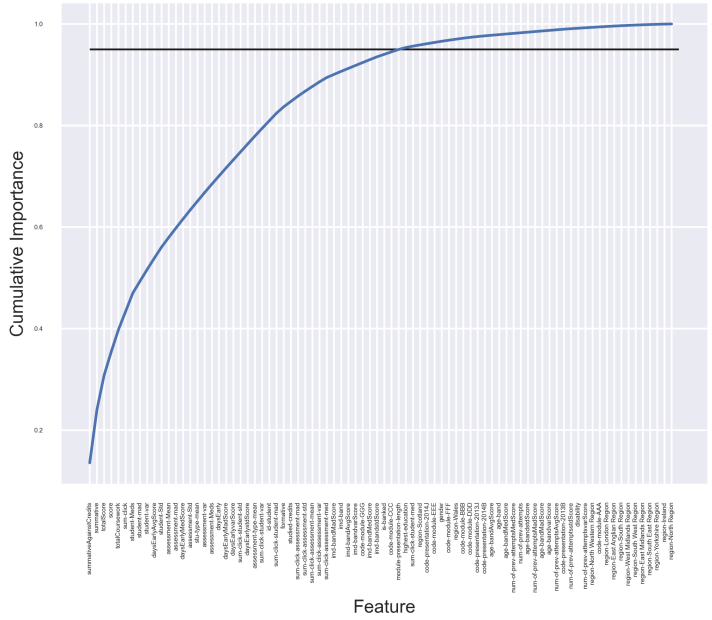


Figure 6: Cumulative Importances of Features

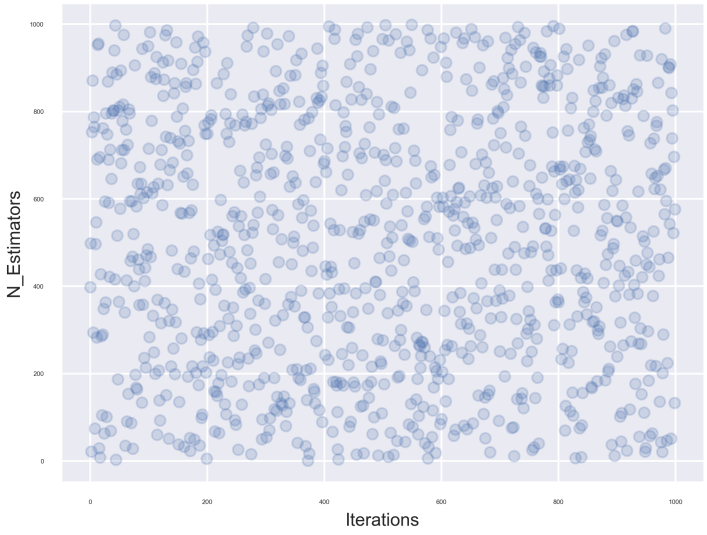


Figure 9: Selected n-Estimators againts iteration

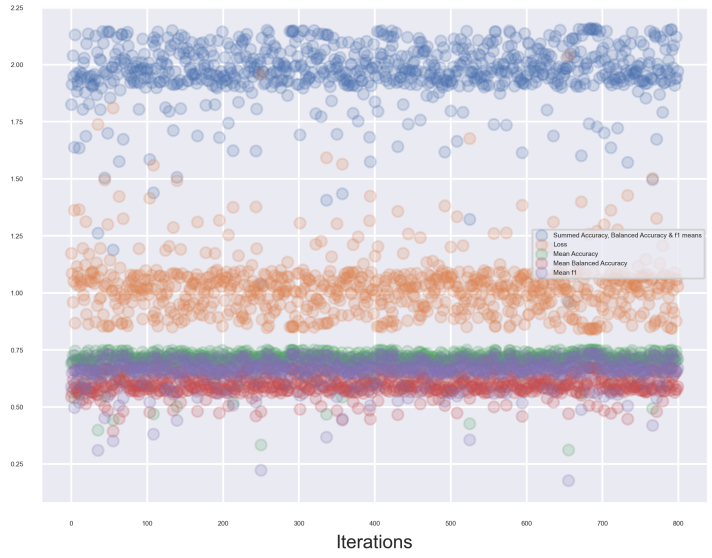


Figure 7: Metrics against iteration 1

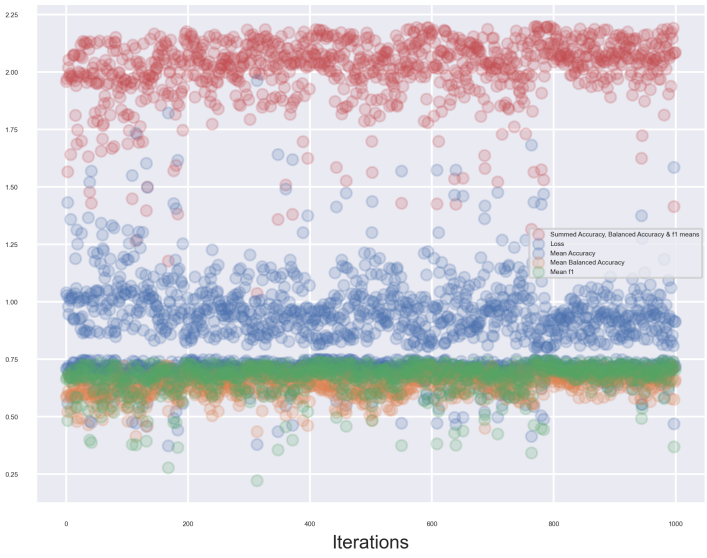


Figure 10: Metrics against iteration 2

| Model | Class | Mean | SD |
|---|---|---|---|
| Linear | 4 | 0.604 | 0.010 |
| Logistic | 4 | 0.702 | 0.007 |
| - | 3 | 0.767 | 0.005 |
| - | 2 | 0.915 | 0.003 |
| SVR Linear | 4 | 0.575 | 0.013 |
| SVR Poly | 4 | 0.746 | 0.008 |
| SVR RBF | 4 | 0.723 | 0.009 |
| SVC | 4 | 0.681 | 0.007 |
| - | 3 | 0.782 | 0.005 |
| - | 2 | 0.932 | 0.003 |
| DT | 4 | 0.674 | 0.003 |
| - | 3 | 0.755 | 0.005 |
| - | 2 | 0.915 | 0.002 |
| RF | 4 | 0.748 | 0.005 |
| - | 3 | 0.809 | 0.005 |
| - | 2 | 0.942 | 0.003 |

Table 3: Model Selection CV Accuracy Metrics

| | With | Fail | Pass | Dist |
|---|---|---|---|---|
| With | 2056 | 462 | 22 | 4 |
| Fail | 606 | 841 | 251 | 3 |
| Pass | 8 | 205 | 2615 | 293 |
| Dist | 0 | 0 | 248 | 535 |

Table 4: RF 4 Class Confusion Matrix

| | Fail | Pass |
|---|---|---|
| Fail | 3937 | 308 |
| Pass | 60 | 3844 |

Table 5: RF 2 Class Confusion Matrix

| | With | Fail | Pass | Dist |
|---|---|---|---|---|
| With | 2097 | 373 | 72 | 2 |
| Fail | 723 | 640 | 335 | 3 |
| Pass | 84 | 125 | 2748 | 164 |
| Dist | 1 | 3 | 390 | 389 |

Table 6: Logistic 4 Class Confusion Matrix

| | Fail | Pass |
|---|---|---|
| Fail | 3889 | 356 |
| Pass | 261 | 3643 |

Table 7: Logistic 2 Class Confusion Matrix

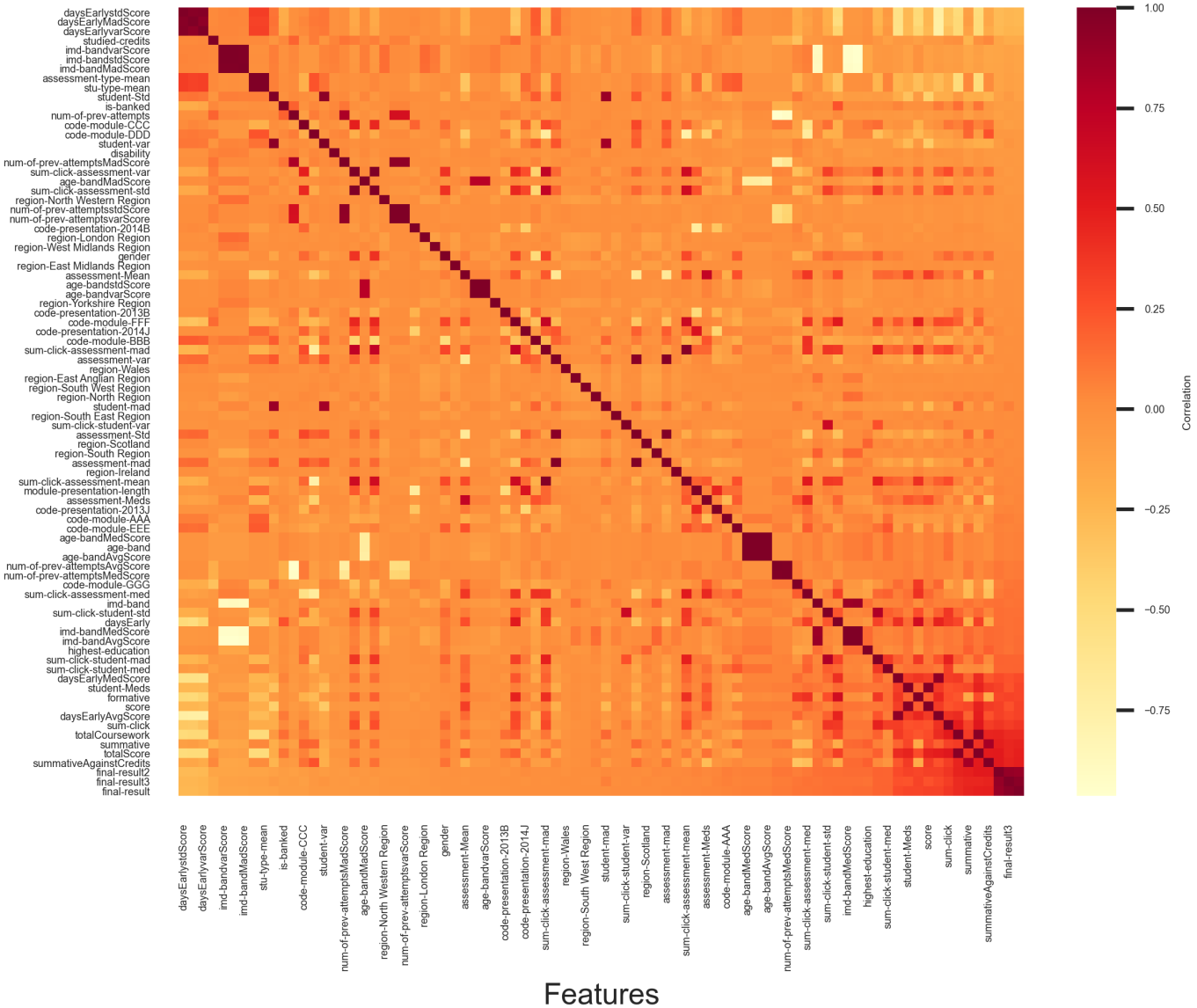


Figure 11: Correlation Heatmap