Machine Learning Coursework - OULAD Analysis

mbtj48

1 Data Gathering & Analysis

Machine learning & data gathering are paramount for modern, cutting edge technologies; thus we have been tasked to develop 2 machine learning models to predict final grades from the OULAD.

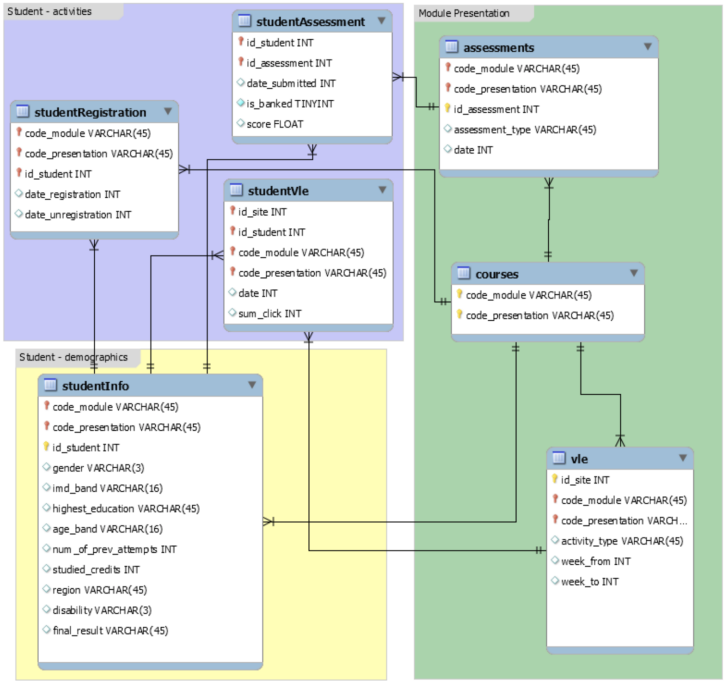

Figure 1: Dataset Schema

Firstly, I noticed useful features such as the score in the studentAssessment table, & 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 & their average mark. I expected these features to show a positive correlation because higher scores and grades generally correlate with high effort (implied by the portal visits). N.b. This is shown in table 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. Further, I calculated how many days early a student submitted coursework using the date-submitted column. Ideally, I expected a positive correlation as the student would be more prepared and committed. In addition, I calculated their summative and formative (where their weight is 0) marks. I eventually included almost all of the data available, so I started to interpret the data differently, including the mean, median, mean absolute deviation, standard deviation & variance for the scores of students' coursework. This, therefore, uses the extrapolated data to find deduce better predictions from the schema. I then produced a correlation heatmap, as well as the sorted numerical correlations.

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 has a poor correlation; in theory, you would expect a mild negative correlation. Although this could be because of the limited data (3 unique ranges). To improve this correlation, I would need specific & precise data.

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 info. I decided to pick one regressor & one classifier to explore: Logistic Regression & 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 specified domain. I gave the model, two

potential sets of combinations, the first, cycled through the C value, which is the regularisation strength; smaller values show a stronger strength, so I started with a strong logarithmic scale to check through, until after enough testing I reached a range of 950-1050. It also cycled through the tolerance value with small values around the default of 0.0001. The other set of combinations check the same values of C and adjust the solver & penalty used. Finally, I removed the second set of combinations as it did not prove to help increase performance.

4 Model B - Random Forest Classifier

Initially, I used a random search to tune the classifier. This randomly picks n combinations to validate the model against from the parameter domain space. I decided to check the number of estimators (trees in the forest), the max depth of each tree in the forest, the minimum number of samples required at a leaf node, & the minimum number of samples required to split an internal node. I moved on to use bayesian optimisation in order to minimise this complex search problem. This uses the previous iterations to strategically pick the next best parameters to pick from the search space with the aim of reducing the loss function. Following on, I removed the unimportant features from the forest's feature importances 6 and hyper tune again in order to further improve the model. I noticed during this, the number of estimators showed little correlation for the model improving the loss function 9.

5 Conclusion

Model	**Logistic**		**Random Forest**	
Classes	**2**	**4**	**2**	**4**
Explained Var	TBD	TBD	TBD	TBD
RMSE	TBD	TBD	TBD	TBD
r2 Score	TBD	TBD	TBD	TBD
f1 Score (Recall & Precision)	TBD	TBD	TBD	TBD
f1 weighted average	TBD	TBD	TBD	TBD
Accuracy	TBD	TBD	TBD	TBD

Table 2: Metrics of Final Models

In conclusion, the logistic regression model finished with an accuracy of 0.68%, whereas the Random Forrest Model finished with an accuracy of 0.59%, therefore making the logistic model initially more desirable. Upon further inspection and validation on the testing data set, the maxiumum error was 0.3 lower for the random forrest, & the r2 score was 0.25 higher for random forrest, potentially making random forrest overall a better choice.

Appendix

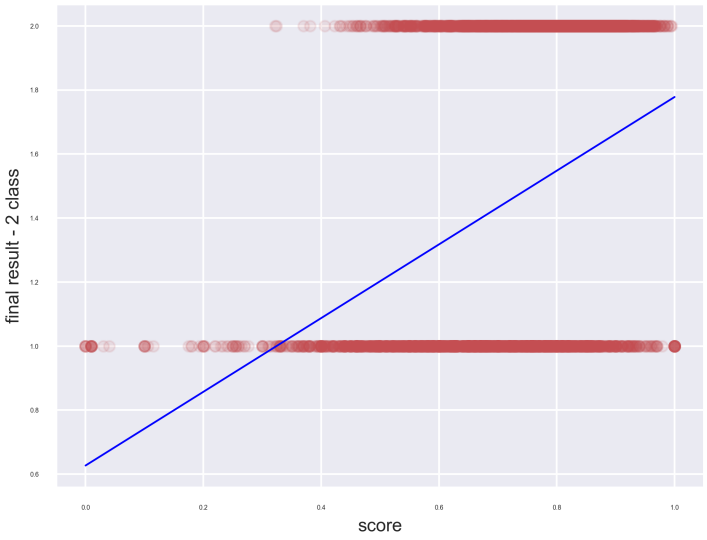

Figure 2: Linear Regression (2 class) against score

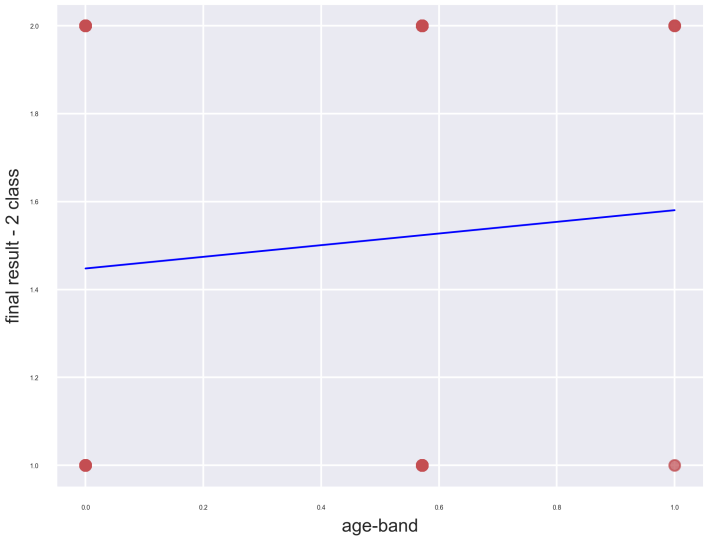

Figure 3: Linear Regression (2 class) against age-band

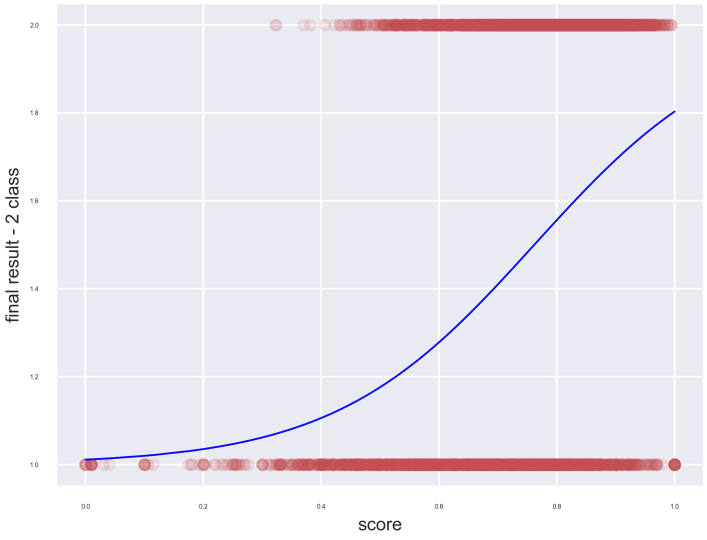

Figure 4: Logistic Regression (2 class) against score

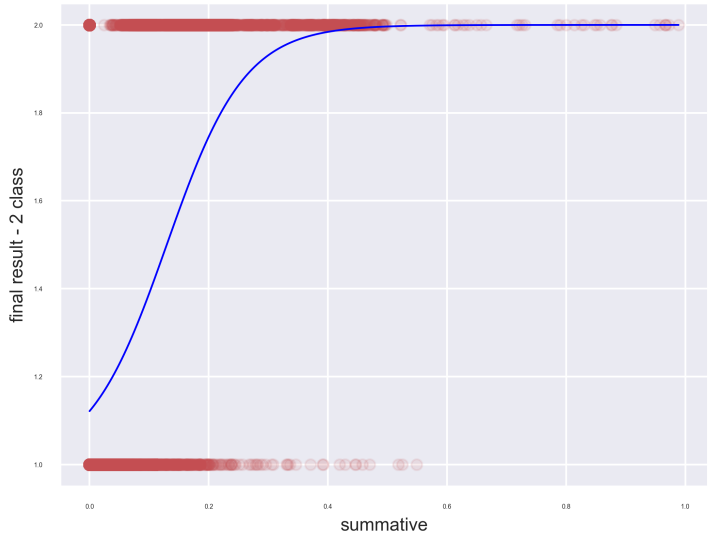

Figure 5: Logistic Regression (2 class) against summative

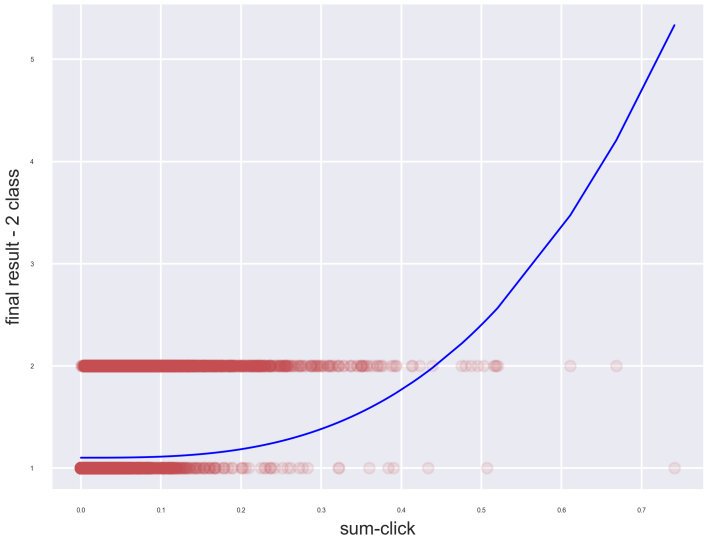

Figure 8: SVR-Poly-Kernel (2 class) against sum-click

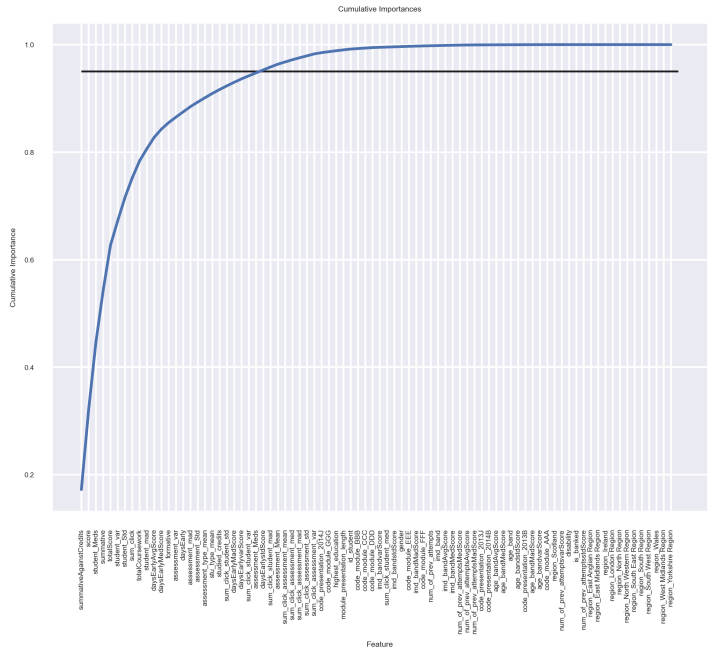

Figure 6: Cumulative Importances of Features

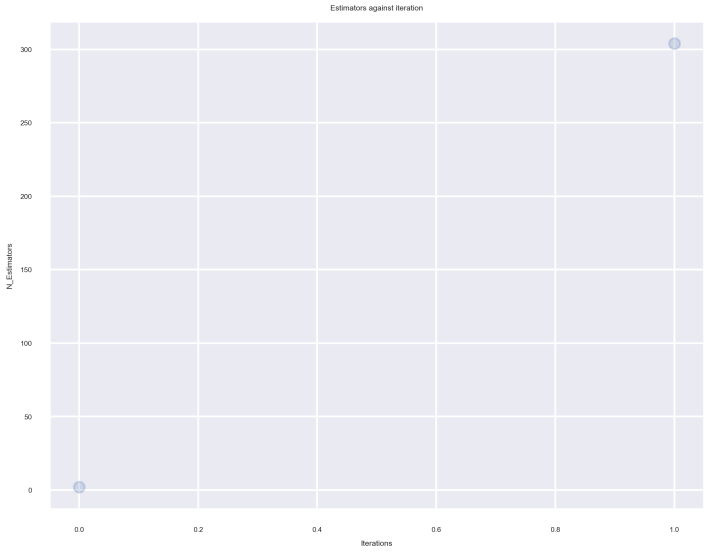

Figure 9: Selected n-Estimators againts iteration

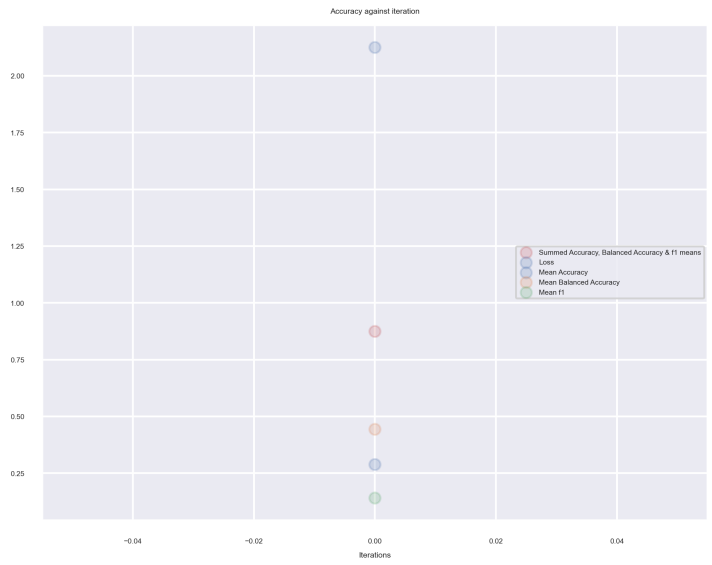

Figure 7: Metrics against iteration 1

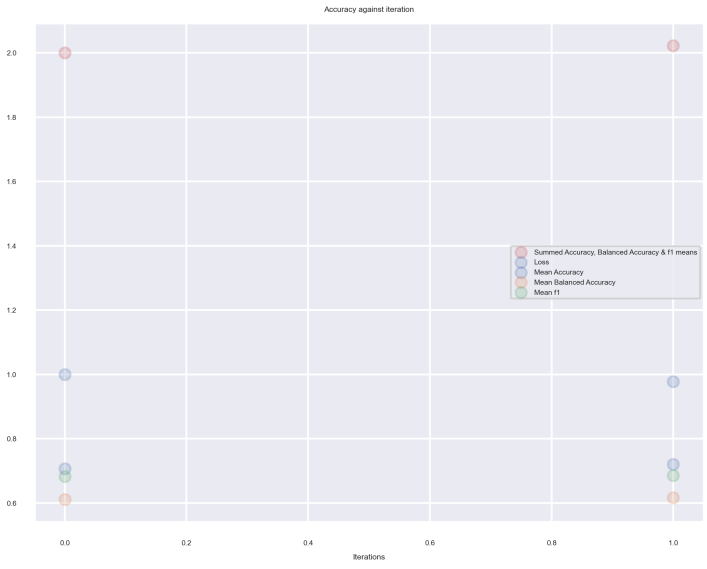

Figure 10: Metrics against iteration 2

Model	n-Classes	Mean	Standard Deviation
Linear Regression	4	0.604	0.010
Logistic Regression	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
Decision Tree	4	0.674	0.003
-	3	0.755	0.005
-	2	0.915	0.002
Random Forest	4	0.748	0.005
-	3	0.809	0.005
-	2	0.942	0.003

Table 3: Model Selection CV Accuracy Metrics

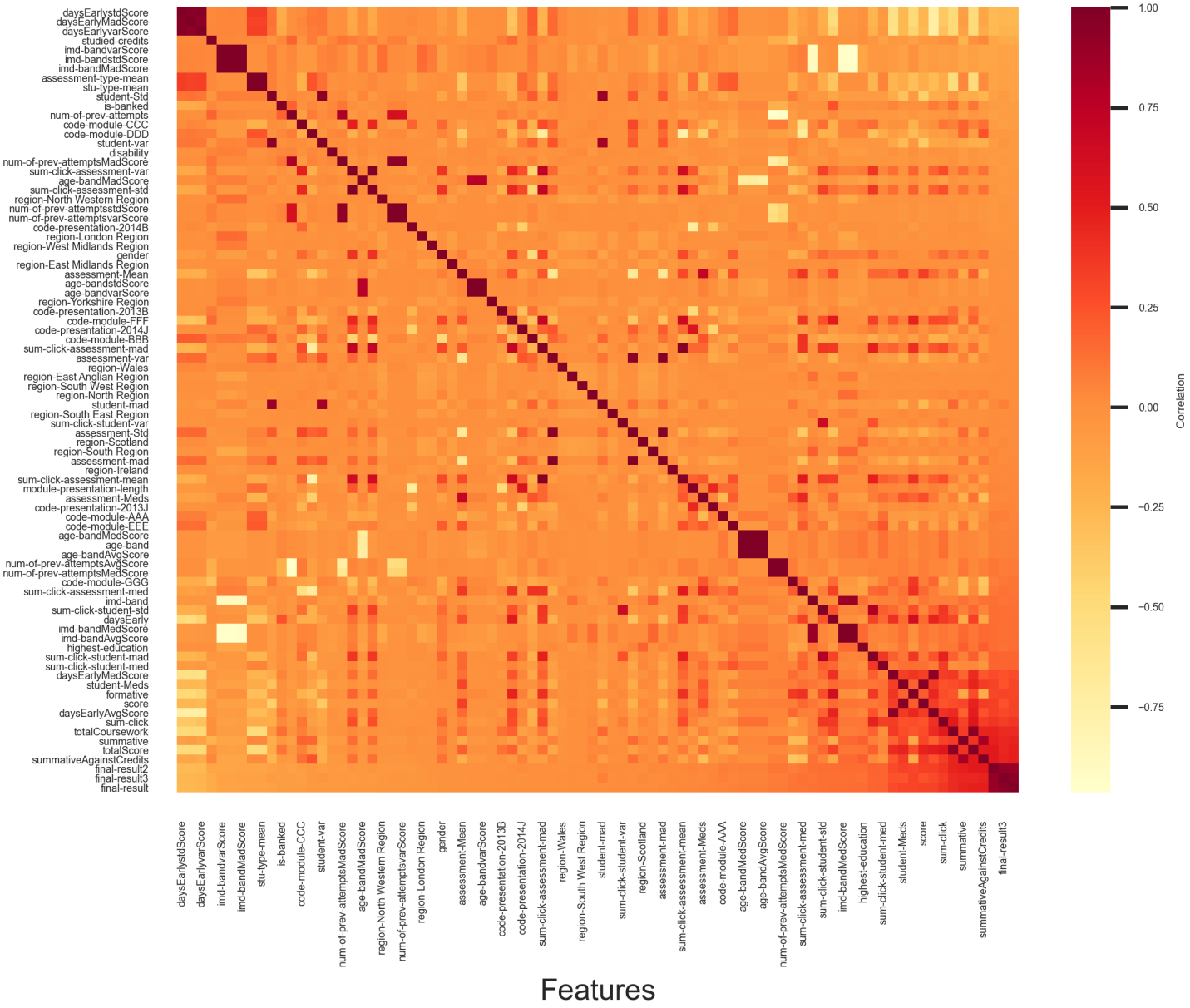

Figure 11: Correlation Heatmap