In undertaking this assignment, the Machine Learning Lifecycle was closely followed; this report entails a discussion of experimental procedure and performance comparison for 2 machine learning methods used to predict student performance.

i. Gathering data

Utilized data was obtained from the OULAD dataset, which consists of 7 anonymized dataframes presented in CSV format. For visualization of the data schema, see [1].

ii. Preparing data

In preparing this dataset for machine learning, the pandas library was used to carry out data analysis and manipulations. The first preprocessing step involved merging the 7 files along a unique identifier; upon inspection, it was determined that the grouping {"code_module", "code_presentation", "id_student"} uniquely identified each record.

Table 0. Assembling the Data Set

Dataset	Files	Merged On
Student Registration /Course Data	studentRegistration.csv, studentInfo.csv, courses.csv	{"id_student", "code_module", "code_presentation"}
Assessment Data	assessments.csv, studentAssessment.csv"	"id_assessment"
VLE Data	vle.csv, studentVle.csv	{"id_site", "code_module", "code_presentation"}
ML Dataset	above 3 data frames	{"id_student", "code_module", "code_presentation"}

After performing feature selection and engineering within the 3 respective dataframes, they were merged into one comprehensive dataset.

Feature Selection, Engineering

It was noted that the dataset contains many missing data values, particularly with regards to assessment and VLE data. Likewise, different numbers of assessments and VLE interactions were recorded per unique identifier, causing disproportionate representation for students who took more assessments (whether as a result of particular module structure or otherwise) and interacted with VLE resources on more separate occasions (i.e. on different days, while not necessarily registering more clicks).

New features were engineered to consolidate assessment and VLE data: avg_TMA_score, avg_CMA_score, total_VLE_clicks, avg_VLE_clicks, total_VLE_visits. For VLE data, total_click, avg_click, and total_visit features were also engineered on a per category basis for individual VLE resource types. A summary of features is displayed in the adjacent tables:

module_presentation_length, weight, date_submitted, is_banked, week_from, week_to, date_unregistration and date features were not utilized and thus removed from the dataframe.

Table 1. OULAD Dataset Features Before Selection and Engineering

File (.csv)	# Records	Features
courses	22	code_module, code_presentation, module_presentation_length
studentInfo	32593	code_module, code_presentation, id_student, gender, region, highest_education, imd_band, age_band, num_of_prev_attempts, studied_credits, disability, final_result
studentRegistration	32593	code_module, code_presentation, id_student, date_registration, date_unregistration
assessments	196	code_module, code_presentation, id_assessment, assessment_type, date, weight
studentAssessments	173740	id_assessment, id_student, date_submitted, is_banked, score
vle	6365	id_site, code_module, code_presentation, activity_type, week_from, week_to

Table 2. OULAD Dataset Features After Selection and Engineering

Feature Category	# Features	Features
Student Registration Info	4	code_module, code_presentation, student_id, date_registration
Student Demographics	9	gender, region, highest_education, imd_band, age_band, num_of_prev_attempts, studied_credits, disability, final_result
Assessment Info	3	avg_TMA_score, avg_CMA_score, Exam_score
VLE Info (across all VLE resource types)	3	total_VLE_clicks, avg_VLE_clicks, total_VLE_visits
VLE Resource-Specific Info	3 * 20 categories = 60	total_VLE_clicks_(category), avg_VLE_clicks_(category), total_VLE_visits_(category), where resource type categories include {resources, oucontent, url, homepage, subpage, glossary, forumng, oucollaborate, dataplus, quiz, ouelluminate, sharedsubpage, questionnaire, page, externalquiz, ouwiki, dualpane, repeatactivity, folder, htmlactivity}

Next, missing data values were addressed:

- Missing avg_TMA/CMA scores replaced with mean of column
- Missing Exam scores replaced with '0'
- Records with missing date_registration and imd_band removed from dataset
- Missing VLE data (overall and resource-specific) replaced with mean of column
  - VLE category features missing >50% of data were dropped from the dataset

Categorical and binned features were transformed using mapping and one-hot encoding techniques:

- imd_band: replace with mean of band range, remove '%' symbol
- age_band: replace with mean of band range, modify '55<=' to 55-90 (thus maintaining interval band size of 35)

- disability: mapped using sklearn LabelEncoder(); N = 0, Y = 1
- gender: mapped; F = 0, M = 1
- region, highest_education: one-hot encode categorical features using pandas.get_dummies()
- final_result: mapped “Pass”/“Distinction” = 1, “Fail”/“Withdrawn” = 0
  - 2 variants of each model were trained:
    - *Pass/Distinction - Fail*: as many ‘Withdrawn’ instances lack comprehensive assessment and VLE data, they were dropped from the dataset. ‘Distinction’ instances were merged with ‘Pass’ instances.
    - *Pass/Distinction - Fail/Withdrawn*: after preprocessing, it was noted that the spread of student outcomes was imbalanced (Table 4). It was of interest to see if this imbalance would have a significant effect on model performance. Thus, in the second variant, positive instances (Pass, Distinction) were mapped to 1 and negative instances (Withdrawn, Fail) to 0, producing a binary classification problem in which the dataset was relatively balanced for the target feature.

Table 4. Label Distribution (4 Classes)

Result	Labeled Data Count
Pass	11827
Withdrawn	6978
Fail	6537
Distinction	2825

Table 5. Label Distribution (P/D - F)

Result	Labeled Data Count
Pass/ Distinction	14652
Fail	6537

Table 6. Label Distribution (P/D - F/W)

Result	Labeled Data Count
Pass/ Distinction	14652
Fail/ Withdrawn	13515

Once features were numerically encoded, the id {“code_module”, “code_presentation”, “id_student”} was removed; otherwise, the numeric values stored in these columns may bear unnecessary influence and result in bias when training the model. Features were scaled, normalizing data values to the range [0,1].

iii. Choosing models
2 machine learning models were trained on the prepared dataset: random forest and logistic regression.

The random forest model was selected due to its compatibility with both numerical and categorial features, as well as support for high-dimensional datasets. An RF-classifier was used to suit the binary classification required by the problem. By incorporating bagging and bootstrapped data samples, random forests are more accurate and less prone to overfitting than a single decision tree alone. Similarly, logistic regression was chosen to suit the binary nature of the dependent variable. Logistic regression models are best suited for predicting a categorical response variable with 2 outcomes.

iv. Training
The target feature was set to ‘final_result’. train_test_split() was used to create training and test sets, comprising 80% and 20% of the dataset, respectively. Models were then trained, and their performance evaluated on the unseen test set.

v. Evaluation, hyper-parameter tuning
To account for possible bias and overfitting, k-fold cross validation was performed with k=5. In this resampling procedure, the training set is split into k smaller folds; for each fold, the model is trained using k-1 folds and validated on remaining data. Sample cross-validation output is provided for each model; model performance is comparable across all folds, suggesting confidence in predicting ability.

Table 7. Random Forest: 5-Fold Cross Validation Sample Output

Target Feature	Fold Accuracy
Pass/Distinction - Fail	0.889, 0.880, 0.896, 0.893, 0.904
Pass/Distinction - Fail/ Withdrawn	0.870, 0.869, 0.872, 0.867, 0.859

Table 8. Logistic Regression: 5-Fold Cross Validation Sample Output

Target Feature	Fold Accuracy
Pass/Distinction - Fail	0.856, 0.853, 0.866, 0.868, 0.862
Pass/Distinction - Fail/ Withdrawn	0.868, 0.872, 0.870, 0.871, 0.8589

At this stage, further feature experimentation was conducted. The models were trained using different feature combinations; for example, with and without assessment data, VLE data, demographic features, and so on. For brevity, the feature combination returning the highest prediction accuracy (Table 2) is utilized in this report.

Next, hyperparameter tuning was performed; grid search techniques (using sklearn GridSearchCV) and individual hyperparameter vs. AUC metrics were used to optimize both models. See source code for parameter grids and optimization functions used for each model. Hyper-parameters included:
- *Random Forest*
  - N-estimators: number of trees in the forest

- Max-depth: depth of each tree in the forest
- Min_samples_split: minimum # of samples required to split an internal tree node
- Min_samples_leaf: minimum # of samples required to be at a leaf node

- *Logistic Regression*
  - C-value: regularization parameter, 1/λ
  - Penalty: regularization technique (lasso, ridge regression)

Visualization of individual hyperparameters is provided for the Pass/Distinction-Fail/Withdrawn classification, aiming to maximize AUC score for the test set (denoted by dashed line). For brevity, similar visualizations for Pass/Distinction-Fail are omitted.

Figure 0. Random Forest: Hyperparameter Tuning - Num_Estimators

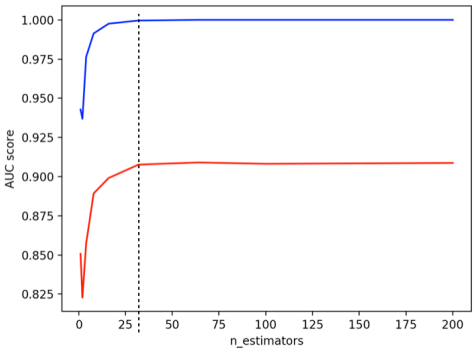


Figure 1. Random Forest: Hyperparameter Tuning - Max Depth

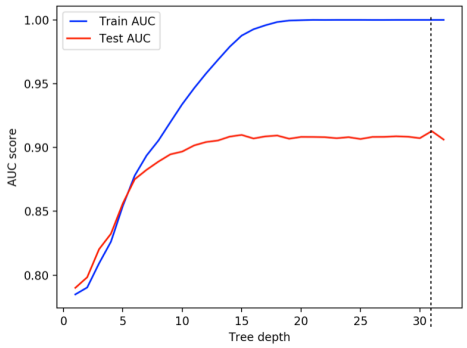


Figure 2. Random Forest: Hyperparameter Tuning - Min Samples Split

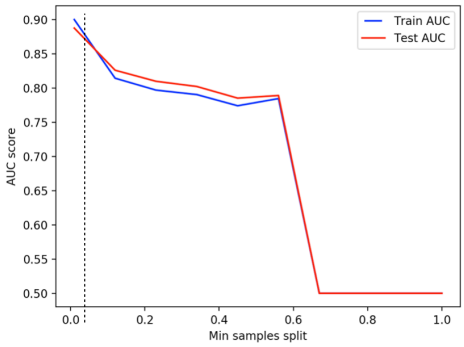


Figure 3. Random Forest: Hyperparameter Tuning - Min Samples Leaf

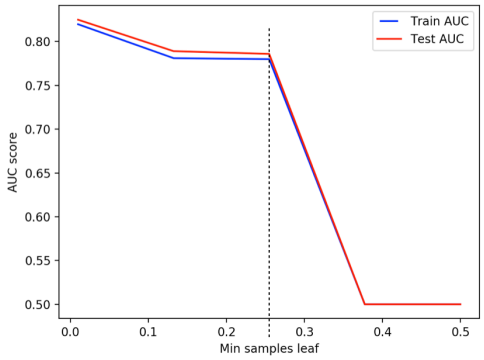
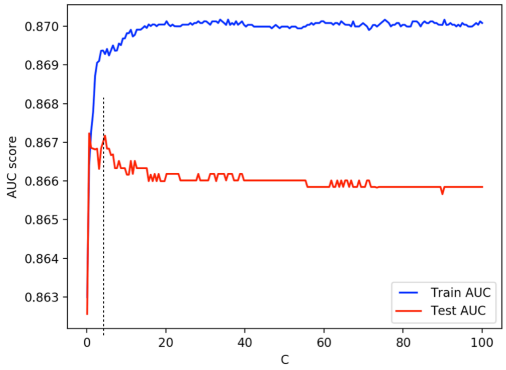


Figure 4. Logistic Regression: Hyperparameter Tuning - C-value



vii. Making predictions, comparing model performance

A range of metrics were used to assess performance, including accuracy, precision, recall, f1-score, and AUC score. Correct and incorrect predictions are further visualized in a confusion matrix, output when running source code.

a) Classifying Pass/Distinction - Fail/Withdrawn

Table 9. Performance Metrics: Random Forest (P/D - F/W)

Binary Classification	Precision	Recall	F1-Score	Support
Pass/ Distinction	0.93	0.90	0.92	3086
Fail/ Withdrawn	0.88	0.92	0.90	2548
Average	0.91	0.91	0.91	5634

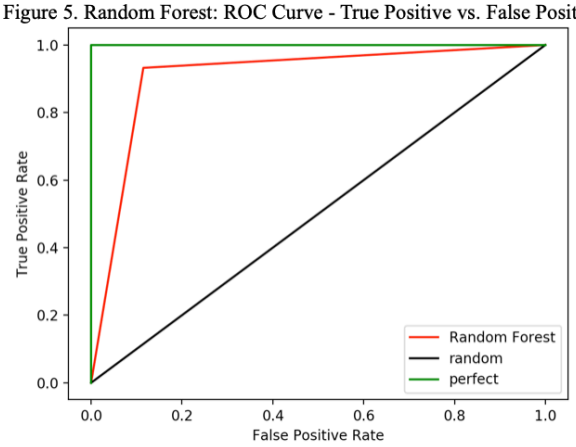
Accuracy ≈ 0.91072

Table 10. Performance Metrics: Logistic Regression (P/D - F/W)

Binary Classification	Precision	Recall	F1-Score	Support
Pass/ Distinction	0.87	0.87	0.87	2968
Fail/ Withdrawn	0.86	0.86	0.86	2666
Average	0.88	0.87	0.87	5634

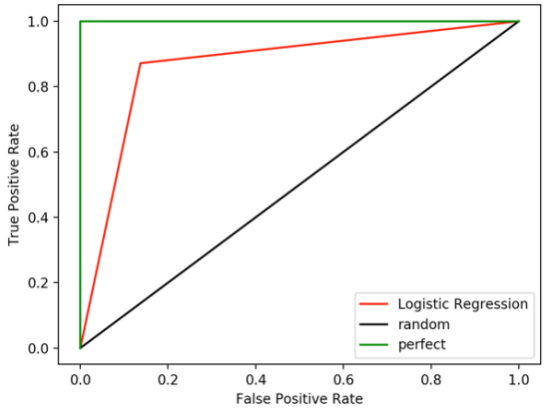
Accuracy ≈ 0.86564

Figure 5. Random Forest: ROC Curve - True Positive vs. False Positive Rate



AUC ≈ 0.90494

Figure 6. Logistic Regression: ROC Curve - True Positive vs. False Positive Rate



AUC ≈ 0.86717

The RF-classifier marginally outperformed the logistic regression model, returning ≈4-5% higher accuracy and AUC-score. Both models were more successful predicting Pass/Distinction than Fail/Withdrawn instances, as evidenced by higher precision and recall rates.

b) Classifying Pass/Distinction - Fail

Table 11. Performance Metrics: Random Forest (P/D - F)

Binary Classification	Precision	Recall	F1-Score	Support
Pass/ Distinction	0.94	0.90	0.92	3127
Fail	0.75	0.85	0.79	1111
Average	0.89	0.88	0.89	4238

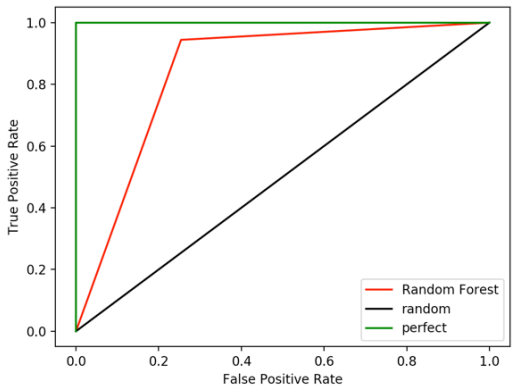
Accuracy ≈ 0.88485

Table 12. Performance Metrics: Logistic Regression (P/D - F)

Binary Classification	Precision	Recall	F1-Score	Support
Pass/ Distinction	0.91	0.88	0.90	3083
Fail	0.71	0.78	0.74	1155
Average	0.86	0.85	0.86	4238

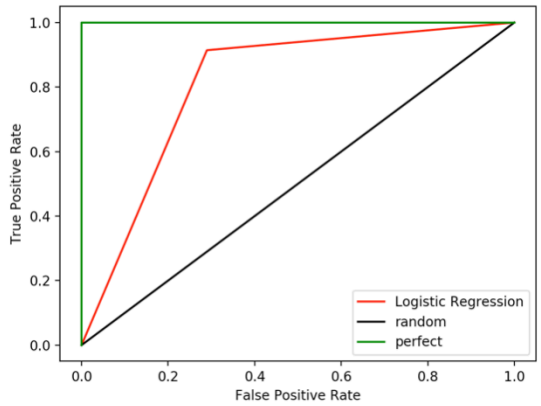
Accuracy ≈ 0.85323

Figure 7. Random Forest: ROC Curve - True Positive vs. False Positive Rate



AUC ≈ 0.84499

Figure 8. Logistic Regression: ROC Curve - True Positive vs. False Positive Rate



AUC ≈ 0.81225

Again, the RF-classifier outperformed logistic regression across all performance metrics. Both models' overall performance deteriorated slightly after Withdrawn labels were removed, with ≈3% and 1% drops in prediction accuracy versus the P/D-F/W classifier. This is evidenced by a widening gap in prediction accuracy across the 2 output classes; while performance improved for Pass/Distinction instances, both models returned significantly lower precision, recall, and F1-scores for Fail instances, suggesting poorer prediction performance for the Fail class. This trend is similarly reflected in decreased AUC-scores for both models.

Nevertheless, in both P/D-F/W and P/D-F classifier variations, the two models exhibit strong performance on the test set. Going forward, to improve performance and provide reliable prediction across all 4 categories, perhaps it would be beneficial to obtain a larger, more diverse datapool that is balanced across the 4 output classes. It may also be of interest to consider temporal aspects of the data, such as timing of student VLE interactions or enrollment/withdrawal tendencies relative to module start date.

References

[1] https://analyse.kmi.open.ac.uk/open_dataset