Machine Learning Assessment: Open University Data

Q1: 20
Q2: 20
Q3: 40
Q4: 10 (Interpretations of results is not as per expectations of the assessor i.e., the learner shall explain how the particular trends or modelled parameters relate to the data or help in future projection related to data)

Total obtained marks 90 out of 100.

Final Grade in This Assessment: **PASS**

Assessed by: Abrar

Author: Daayum Mohsin

Supervisor: Abrar Ahmad

Institution: IT Professional Training

Course: Machine Learning (J1BB 35)

Contents

Summary

This study aimed to predict student final results using machine learning models based on demographic and assessment data from the Open University dataset. The project implemented Logistic Regression, Random Forest and XGBoost to classify students into four categories: Distinction, Pass, Fail and Withdrawn. Among the models tested, XGBoost outperformed Logistic Regression and Random Forest, achieving the highest accuracy (63.46%) and F1- score across multiple categories, followed by Logistic Regression (58.42%) and Random Forest (57.34%).

The study also identified key challenges, including class imbalance, numeric data bias and overfitting risks, which reduced the models' effectiveness in predicting minority categories. To improve on the model's performance, future work should focus on handling class imbalance through applying GridSearchCV for hyperparameter tuning, and incorporating additional categorical variable to enhance classification.

This study highlights the potential of machine learning in educational data analysis but emphasizes the need for further refinements to ensure fairness and reliability in predicting student performance at the Open University.

Introduction & Problem Definition

For the modern educational landscape, predicting student performance has become an essential tool for academic institutions. Machine Learning (ML) techniques offer a data-driven approach to identifying students at risk of failing and optimising intervention strategies. This project leverages demographic and assessment data from the Open University dataset to build predictive models for student results.

To achieve this, three machine learning models were implemented:

- Logistic Regression: A baseline model for understanding linear relationships.
- Random Forest: An ensembled model that spots and captures complex interactions.
- XGBoost: A High-performance boosted model which is optimised for structured data.

Dataset Overview:

- StudentInfo.csv
- Assessments.csv
- StudentAssessment.csv

```
[1]: #Import Libraries
     import pandas as pd
     import seaborn as sns
     import matplotlib.pyplot as plt
     import numpy as np
```

```
[2]: !pip install xgboost
```

```
Requirement already satisfied: xgboost in c:\users\gtb21125\appdata\local\anaconda3\lib\site-packages (2.1.4)
Requirement already satisfied: numpy in c:\users\gtb21125\appdata\local\anaconda3\lib\site-packages (from xgboost) (1.26.4)
Requirement already satisfied: scipy in c:\users\gtb21125\appdata\local\anaconda3\lib\site-packages (from xgboost) (1.11.4)
```

```
[3]: #Load Datasets
     assessments = pd.read_csv('C:/Users/gtb21125/Downloads/assessments.csv')
     student_info = pd.read_csv('C:/Users/gtb21125/Downloads/studentInfo.csv')
     student_assessments = pd.read_csv('C:/Users/gtb21125/Downloads/studentAssessment.csv')
```

```
[6]: merged_data = pd.merge(student_info, student_assessments, on ='id_student')
```

```
[7]: merged_data['final_result'] = merged_data['final_result'].astype('category').cat.codes
```

```
[8]: final_df = pd.merge(merged_data, assessments, on='id_assessment')
```

Problem

Educational institutions continuously seek ways to improve student retention rates and overall academic performance success. Through predicting student outcomes this helps Open University to offer targeted support, allocate resources effectively, and improve learning experiences. However, this task presents several complex challenges:

1. Complex Data Relationships: Student performance is influenced by multiple factors, including demographics and the number of previous attempts.
2. Class Imbalance: Certain outcomes “Withdrawn” occur less frequently and occur more spontaneously due to mitigating circumstances outside of university life.
3. Feature Selection: What attributes correlate most with academic success.
4. Computational Costs: How can we run advanced models that require significant computational power on a laptop.

Aims

This project aims to build, compare and evaluate machine learning models to determine the most effective approach for predicting student success. By addressing these challenges, the study serves to provide actionable insights for educational data analysis for the Open University and lay the foundation for future improvements into student performance.

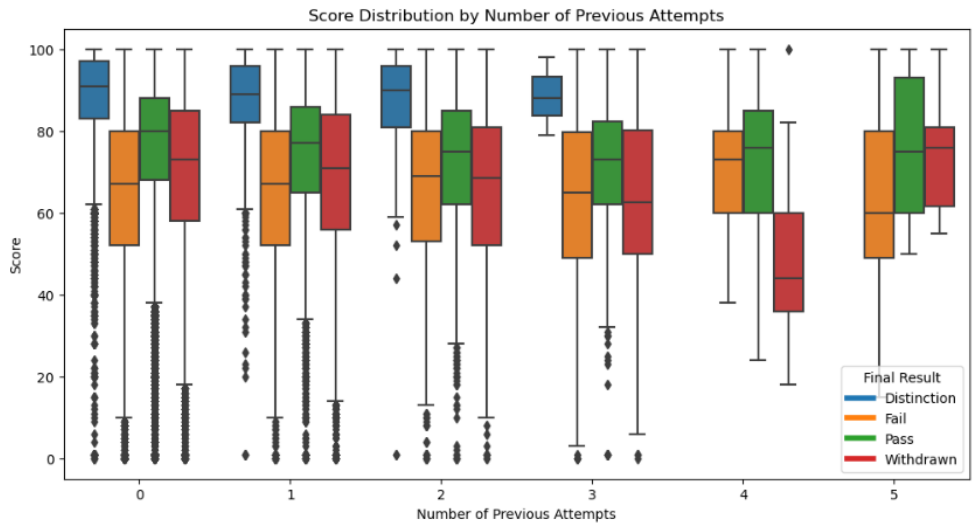
Exploratory Data Analysis (EDA)

Missing Values: The dataset was refined by identifying and removing missing values from two sections: imd_band (9315 missing values) and date (4018 missing values). These columns were dropped to ensure data integrity and streamline further analysis.

```
[13]: missing_vals = final_df.isnull().sum()
      print(missing_vals[missing_vals >0])

      imd_band    9315
      date        4018
      dtype: int64

[14]: final_df = final_df.drop(columns=['imd_band', 'date'])
```

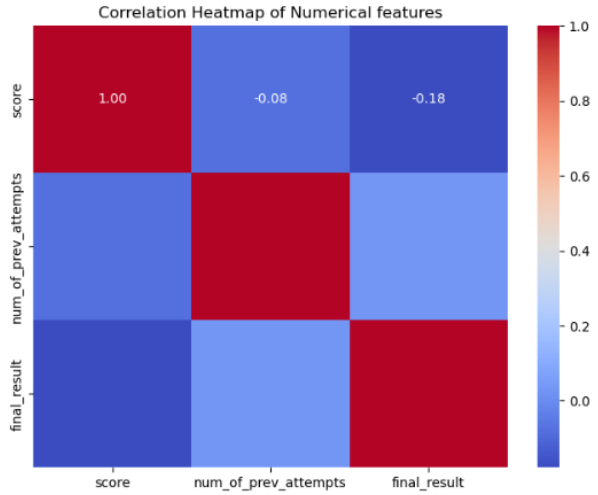
Histogram Analysis: The dataset was refined by removing entries where the number of previous attempts was 6, lacked Pass and Fail data. This ensures a more balanced representation of final result distributions across different attempt levels.

```
[23]: # Remove data where num_of_prev_attempts is 6 (as it lacks Pass and Fail data)
      filtered_df= final_df[final_df['num_of_prev_attempts'] <6]

[24]: plt.figure(figsize=(12, 6))
      sns.boxplot(x='num_of_prev_attempts', y='score', hue='final_result', data=filtered_df)
      plt.title("Score Distribution by Number of Previous Attempts")
      plt.xlabel("Number of Previous Attempts")
      plt.ylabel("Score")
      plt.legend(handles=legend_labels,title="Final Result", loc="best")
      plt.show()
```



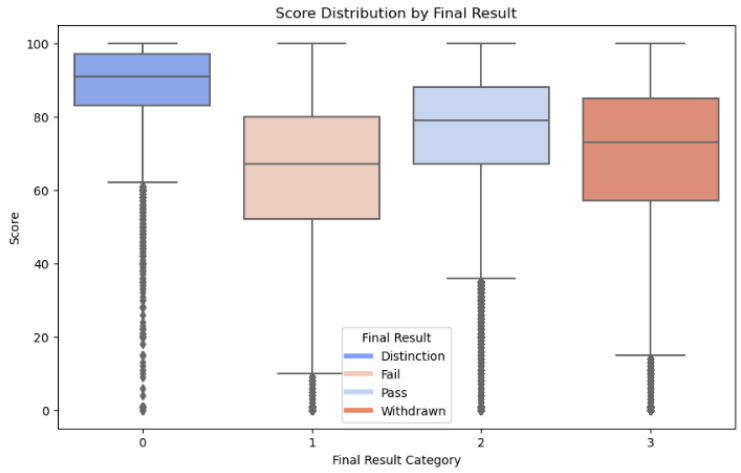
Correlation Heatmap: The heatmap displays a Pearson correlation coefficient between three numerical variables: score, number of pervious attempts and final result. The colour scale represents the strength and direction of the correlation, with red indicating a strong positive correlation (1.00) and blue indicating a negative correlation (-1.00).

```
[19]: plt.figure(figsize=(8,6))
      sns.heatmap(final_df[numerical_features].corr(), annot=True, cmap="coolwarm", fmt=".2f")
      plt.title("Correlation Heatmap of Numerical features")
      plt.show()
```



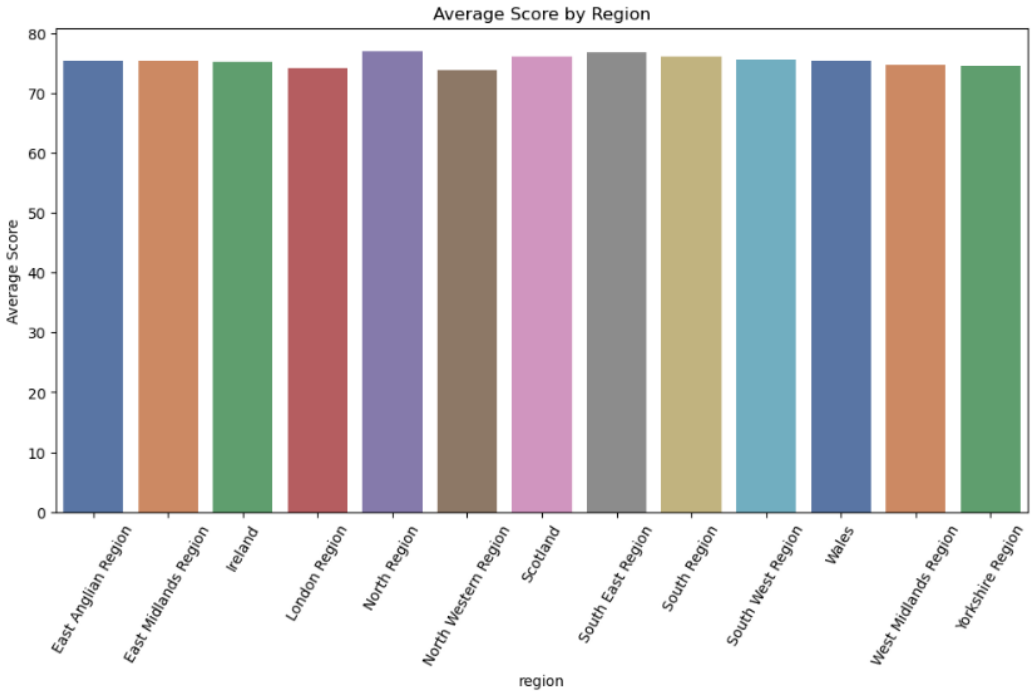
Boxplot Analysis: The spread of student scores across the result four categories; Distinction, fail, pass and withdrawn Each box represents the interquartile range (IQR), with the median score shown as a central line. The whiskers to show data range, excluding outliers, which are plotted as individual points. The Distinction category shows the highest scores, while the fail category exhibits a wider spread with lower median values.

```
[27]: legend_labels = {
          0: "Distinction",
          1: "Fail",
          2: "Pass",
          3: "Withdrawn"}

      legend_palette = {
          0: (0.484, 0.622, 0.975),
          1: (0.947, 0.795, 0.717),
          2: (0.754, 0.830, 0.961),
          3: (0.932, 0.519, 0.406)}

      plt.figure(figsize=(10, 6))
      sns.boxplot(x='final_result', y='score', data=filtered_df, palette=legend_palette)
      plt.title("Score Distribution by Final Result")
      plt.xlabel("Final Result Category")
      plt.ylabel("Score")

      legend_handles = [plt.Line2D([0], [0], color=legend_palette[i], lw=4, label=legend_labels[i]) for i in legend_palette.keys()]
      plt.legend(handles=legend_handles, title="Final Result", loc="best")

      plt.show()
```



Bar Chart: The average score by region bar chart displays the mean scores of students across various regions of the United Kingdom and Ireland.

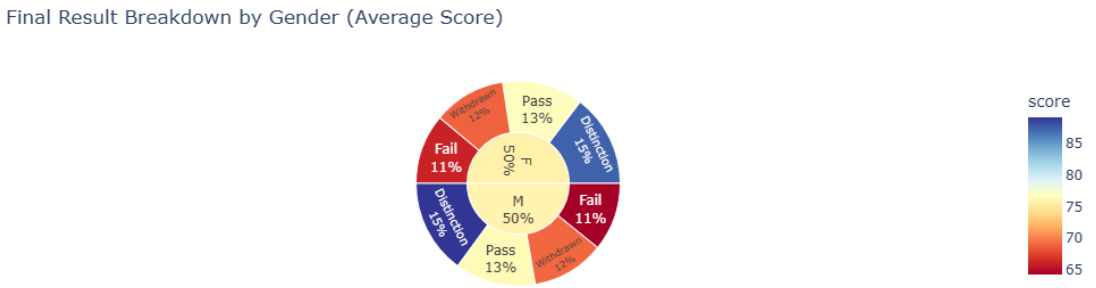
```
[29]: region_scores = filtered_df.groupby("region")["score"].mean().reset_index()

plt.figure(figsize=(12,6))
sns.barplot(x='region', y='score', data=region_scores, palette='deep')
plt.xticks(rotation=60)
plt.title("Average Score by Region")
plt.ylabel("Average Score")
plt.show()
```



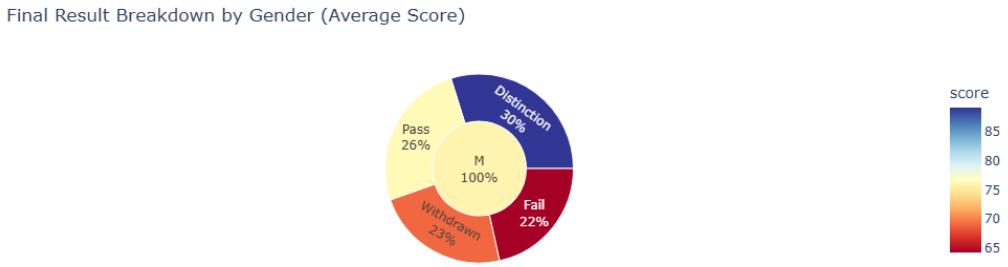
Plotly Analysis; A Python library used for Interactive visualisations.
http://localhost:8889/notebooks/ML_Assessment.ipynb

```
[60]: final_result_mapping = {
    0: 'Distinction',
    1: 'Fail',
    2: 'Pass',
    3: 'Withdrawn'
}
filtered_df['final_result_label'] = filtered_df['final_result'].map(final_result_mapping)
avg_score_df = filtered_df.groupby(['gender', 'final_result_label'])['score'].mean().round(2).reset_index()
fig = px.sunburst(avg_score_df, path=['gender', 'final_result_label'], values='score',
                  title="Final Result Breakdown by Gender (Average Score)", color='score',
                  color_continuous_scale='RdYlBu')
fig.update_traces(textinfo='label+percent entry')
fig.show()
```

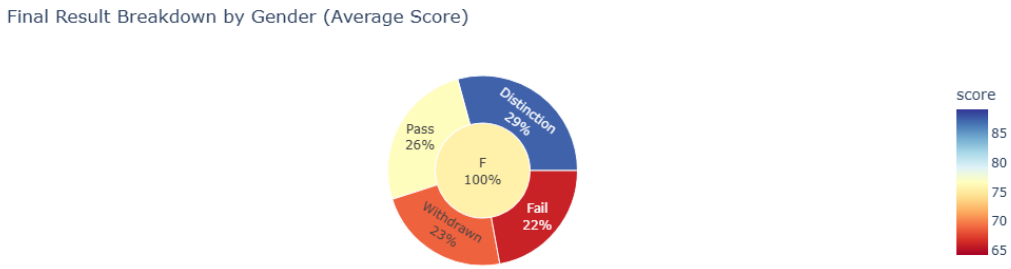
Sunburst Chart: The distribution of student final results based on gender, with average scores represented through a colour gradient. The colour gradient is to represent the score 65 to 100.



Sunburst Chart (M): Male students who achieved Distinction attained the highest average score of 89.16, followed by those who Passed with an average score of 76.24. Students who Withdrew recorded an average score of 68.96, while those who Failed had the lowest average score of 64.23.



Sunburst Chart (F): Female students who achieved Distinction recorded the highest average score of 87.4, followed by those who Passed with an average score of 76.57. Students who Withdrew had an average score of 68.75, while those who Failed obtained the lowest average score of 65.98.



Model Selection & Performance Evaluation

Model Selection and Training: To predict student results, multiple machine learning models were selected and trained to compare their effectiveness.

[33]:
```
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

[34]:
```
scaler = StandardScaler()
numeric_columns = final_df.select_dtypes(include=['int64','float64']).columns
final_df[numeric_columns] = scaler.fit_transform(final_df[numeric_columns])
```

[35]:
```
non_numeric_columns = final_df.select_dtypes(exclude=['number']).columns.tolist()
if 'final_result' in non_numeric_columns:
    non_numeric_columns.remove('final_result')
final_df = final_df.drop(columns=non_numeric_columns, errors='ignore')
```

[36]:
```
X = final_df.drop(columns=['final_result'], errors='ignore')
y = final_df['final_result']
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.3, random_state=0)
```

Logistic Regression:

[36]:
```
log_reg = LogisticRegression()
log_reg.fit(X_train, y_train)
y_pred_log = log_reg.predict(X_test)
log_accuracy = accuracy_score(y_test, y_pred_log) * 100
print(f"Logistic Regression Accuracy:{log_accuracy:.2f}%")
```

```
Logistic Regression Accuracy:58.42%
```

Random Forest Classifier:

[46]:
```
rf = RandomForestClassifier(random_state=0)
rf.fit(X_train, y_train)
y_pred_rf = rf.predict(X_test)
rf_accuracy = accuracy_score(y_test, y_pred_rf) * 100
print(f"Random Forest Accuracy: {rf_accuracy:.2f}%")
```

```
Random Forest Accuracy: 57.34%
```

XGBoost Accuracy:

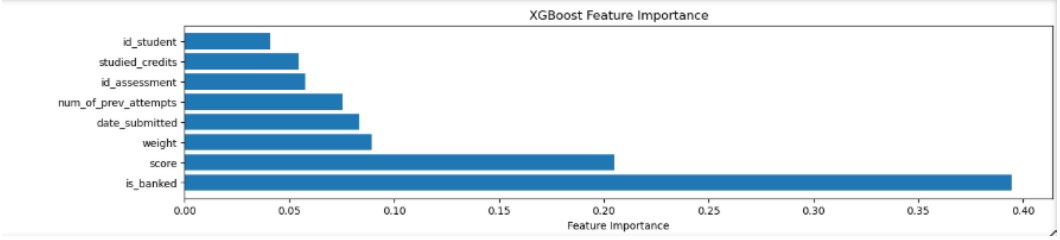
[38]:
```
from xgboost import XGBClassifier
```

[48]:
```
import warnings
warnings.filterwarnings("ignore")
```

[49]:
```
xgb = XGBClassifier(use_label_encoder=False, eval_metric='mlogloss', random_state=0)
xgb.fit(X_train,y_train)
y_pred_xgb = xgb.predict(X_test)
xgb_accuracy = accuracy_score(y_test, y_pred_xgb) * 100
print(f"XGBoost Accuracy:{xgb_accuracy:.2f}%")
```

```
XGBoost Accuracy:63.46%
```

[52]:
```
importances = xgb.feature_importances_
sorted_indices = np.argsort(importances)[::-1]
plt.figure(figsize=(15,3))
plt.barh(range(len(importances)), importances[sorted_indices], align='center')
plt.yticks(range(len(importances)), np.array(X_train.columns)[sorted_indices])
plt.xlabel("Feature Importance")
plt.title("XGBoost Feature Importance")
plt.show()
```



Classification Report: The comparative analysis of F1-scores across Logistic Regression, Random Forest, and XGBoost highlights distinct performance variations in predicting different student outcomes: Distinction (0), Fail (1), Pass (2), and Withdrawn (3).

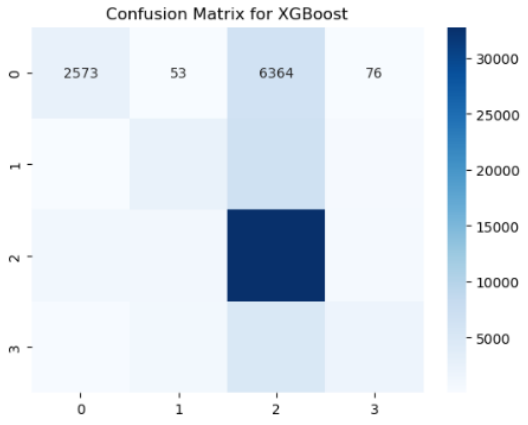
[40]:
```
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
import seaborn as sns
```

[41]:
```
print("Logistic Regression Performance:")
print(classification_report(y_test,y_pred_log))

print("Random Forest Performance:")
print(classification_report(y_test, y_pred_rf))

print("XGBoost Performance:")
print(classification_report(y_test, y_pred_xgb))
```

```
Logistic Regression Performance:
              precision    recall  f1-score   support

           0       0.73      0.03      0.05      9066
           1       0.37      0.09      0.14      9810
           2       0.59      0.97      0.74     35529
           3       0.49      0.12      0.19      7791

    accuracy                           0.58     62196
   macro avg       0.55      0.30      0.28     62196
weighted avg       0.57      0.58      0.47     62196

Random Forest Performance:
              precision    recall  f1-score   support

           0       0.45      0.34      0.39      9066
           1       0.39      0.29      0.33      9810
           2       0.65      0.78      0.71     35529
           3       0.38      0.25      0.30      7791

    accuracy                           0.57     62196
   macro avg       0.47      0.42      0.43     62196
weighted avg       0.54      0.57      0.55     62196

XGBoost Performance:
              precision    recall  f1-score   support

           0       0.63      0.28      0.39      9066
           1       0.55      0.24      0.34      9810
           2       0.64      0.92      0.76     35529
           3       0.59      0.23      0.33      7791

    accuracy                           0.63     62196
   macro avg       0.60      0.42      0.45     62196
weighted avg       0.62      0.63      0.58     62196
```

Confusion Matrix for XGBoost: Confusion Matrix for the XGBoost model illustrating the classification performance across four student outcome categories: Distinction (0), Fail (1), Pass (2), and Withdrawn (3).

[42]:
```
cm= confusion_matrix(y_test, y_pred_xgb)
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
plt.title("Confusion Matrix for XGBoost")
plt.show()
```



Conclusion & Future Work

Evaluation Data Analysis (EDA)

For analysing our merged dataset there were over 9315 values found in the 'imd_band' category and 4018 values in the 'date' column that were removed due to these restrictions. In conducting Boxplot analysis, there were wider distributions for the pass student group as the number of previous attempts increased. The distinction student group saw a shorter distribution as the number of previous attempts increased until after four previous attempts where a distinction was ineligible. In further boxplot analysis, it revealed a higher and more narrow distribution of scores for the pass student group compared to the fail student group which showed a broader and lower distribution of scores. To analyse the average scores by region a bar chart of all regions in the UK and Ireland were performed. The highest performing region by average score was the northern England region while the northwest England region had on average the worst performing scores. The 2nd worst performing region was London which could be due to a greater variety in university enrolment options. The heatmap shows weak correlations, suggesting scores and attempts have minimal impact on results; other factors likely influence performance. To provide an overview of open university scores by gender, a sunburst chart was generated using Plotly. Male students outperformed female students in the Distinction category, achieving an average score of 89.16 to 87.4 for females. However, female students had a marginally higher average score in the pass category 76.57 compared to Male students 76.24. The withdrawn category showed minimal difference, with males averaging 68.96 and females 68.75, suggesting similar disengagement patterns. This highlights that males tend to dominate the top-performing category, whereas female students maintain a more balanced performance across all outcomes.

Model Performance

The performance of the machine learning models- Logistic Regression, Random Forest and XGBoost were selected to demonstrate significant challenges in classified underrepresented categories due to class imbalance. The Distinction (0) and withdrawn (3) groups have notably lower F1-scores which is indicated by the confusion matrix. Among the three models tested, XGBoost demonstrated the highest accuracy (63.46%) and F1-score across multiple categories, outperforming Logistic Regression (58.42%) and Random Forest (57.34%).

Future Work

A key challenge in this study is the class imbalance issue, particularly for the Distinction (0) and withdrawn (3) categories, which are underrepresented and consequently at a high rate. This issue may be exacerbated by analysing only numeric data, which skews classification towards majority groups, limiting the model's ability to accurately capture minority class patterns. A low predictive accuracy for these categories highlights the need for alternative preprocessing techniques such as hyperparameter tuning and incorporating categorical variables.

Future work should involve GridSearchCV to refine model parameters and reduce variance. Another concern is the presence of confounding variables, particularly in withdrawn cases, where external factors beyond academic performance such as financial difficulties and personal circumstances which contribute to withdrawal. Furthermore, feature importance analysis from XGBoost suggests that certain features, 'is_banked' dominate predictions, potentially overlooking other relevant attributes that may improve student classification.