Assignment 2 report.

Juan Sebastian Rodriguez Reyes – c0915840@mylambton.ca

1. Dataset Description

This dataset aims to predict student placement status based on educational, professional, and test scores, with status being the target variable, includes different columns like:

- **ssc_p**: Secondary Education percentage (10th Grade).
- **hsc_p**: Higher Secondary Education percentage (12th Grade).
- **degree_p**: Degree percentage.
- **workex**: Work experience status (Yes/No).
- **etest_p**: Employability test percentage.
- **specialisation**: MBA specialization (e.g., Marketing & HR, Marketing & Finance).
- **status**: Target variable indicating job placement status (e.g., Placed, Not Placed).

2. Data Preprocessing Steps

To ensure model compatibility and performance, the following preprocessing steps were applied:

1. **Categorical Encoding**: Categorical features were encoded to numerical values for compatibility with machine learning models. The columns with only 2 categories where imputed with label encoding, and the columns that presented 3 categories where encoded using one hot encoding.
2. **Data Splitting**: The data was split into training and testing sets, with a 70% for training and 30% for testing. I split the data before filling the missing values and applied SMOTE because in order to avoid data leakage I applied these techniques only in the training dataset
3. **Handling Missing Values**: Missing values occurred only in the salaries column. To fill those missing values, I implemented target imputation using the degree column using only in the training dataset to calculate the average salaries. The average salaries per degree resulting in the training dataset was applied for the missing values in the testing dataset.
4. **Balancing the data:** Due to the class imbalance in the target feature, I implemented SMOTE only in the training dataset, also in order to avoid data leakeage, getting the same quantity of records per target category .

3. Model Selection and Rationale

Four models were chosen for evaluation, each with unique strengths suited for this classification problem:

- **Logistic Regression**: Selected for its simplicity and interpretability in binary classification tasks.
- **Random Forest**: Used for its robustness to overfitting and ability to capture non-linear patterns.

- **Support Vector Machine (SVM)**: Implemented due to its effectiveness in high-dimensional spaces, aiming to create optimal decision boundaries.
- **XGBoost**: Selected for its performance in handling complex relationships through gradient boosting.
- **Voting Classifier**: An ensemble approach combining the above models to improve prediction accuracy and reduce individual model bias. It incorporates the 4 models previously mentioned.

Every model was evaluated with GridSearchCV to select the model with the best hyperparameters.

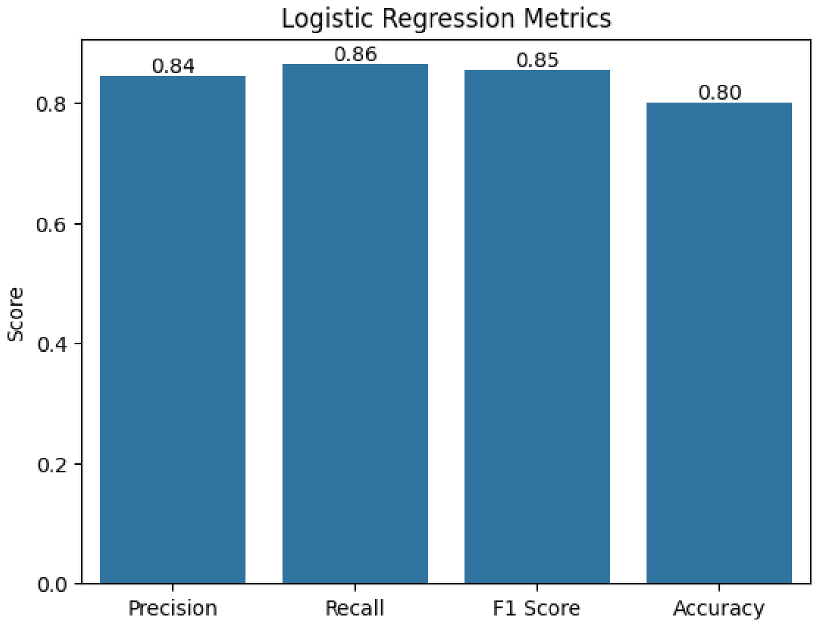
4. Evaluation Metrics

Each model and the Voting Classifier were evaluated using a set of metrics relevant to classification tasks:

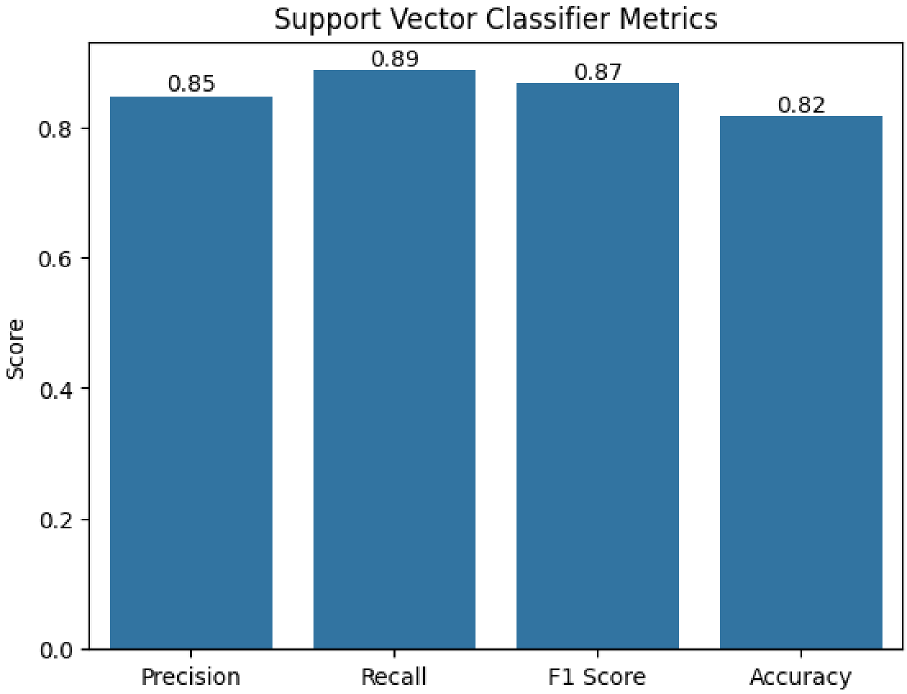
- **Accuracy**: Measures the overall correctness of predictions.
- **Precision**: Evaluates the proportion of positive identifications that were correct.
- **Recall**: Determines the model's ability to correctly identify all relevant instances.
- **F1 Score**: A harmonic mean of precision and recall, balancing both metrics.

These metrics provide a comprehensive evaluation of model performance, particularly the balance between sensitivity and specificity. The results are the following:

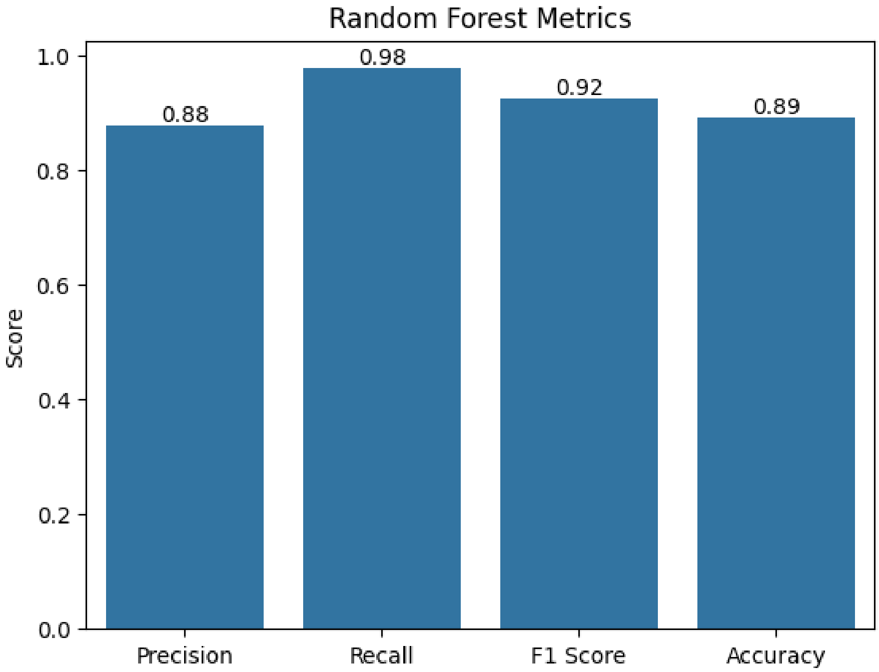
a. Logistic regression



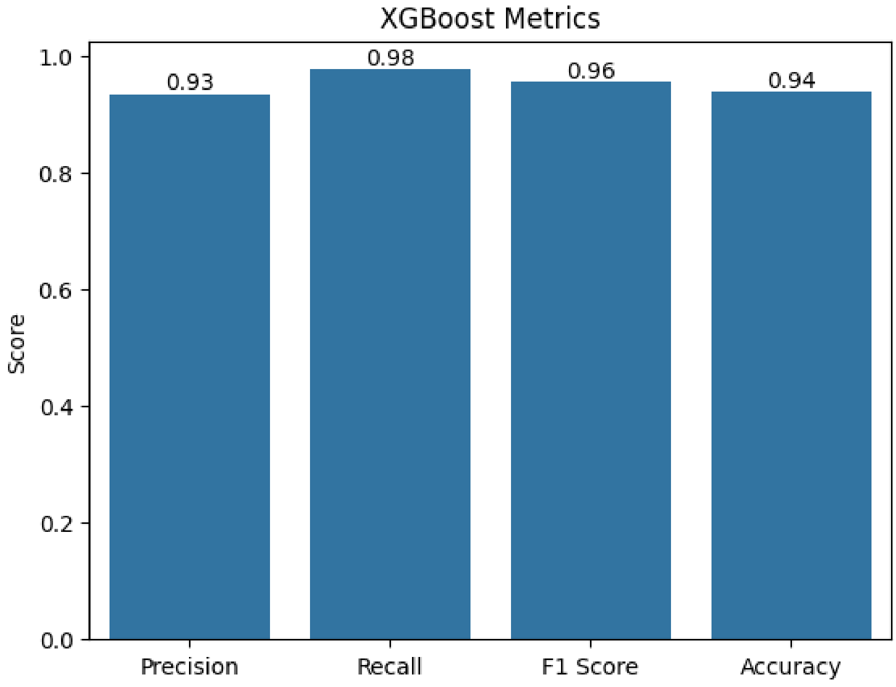
b. Support Vector Machines



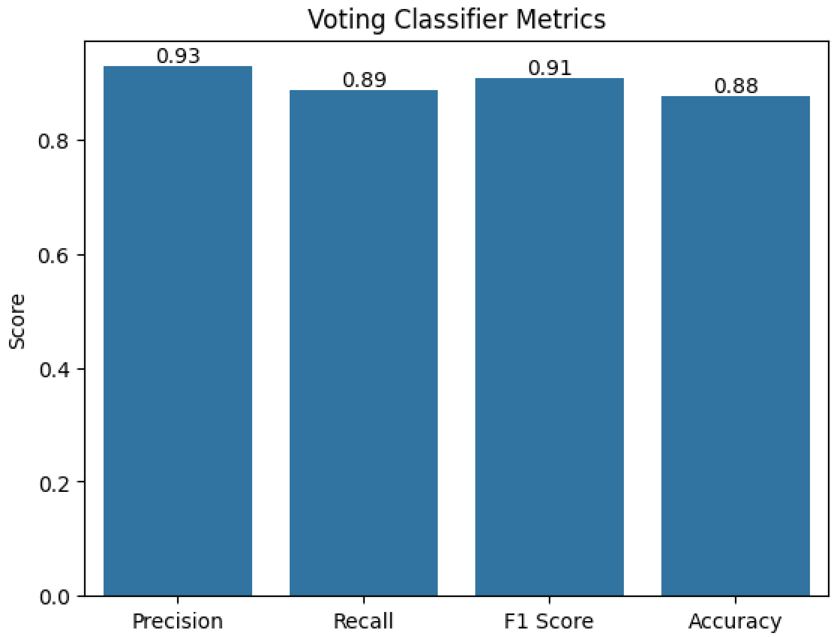
c. Random Forest



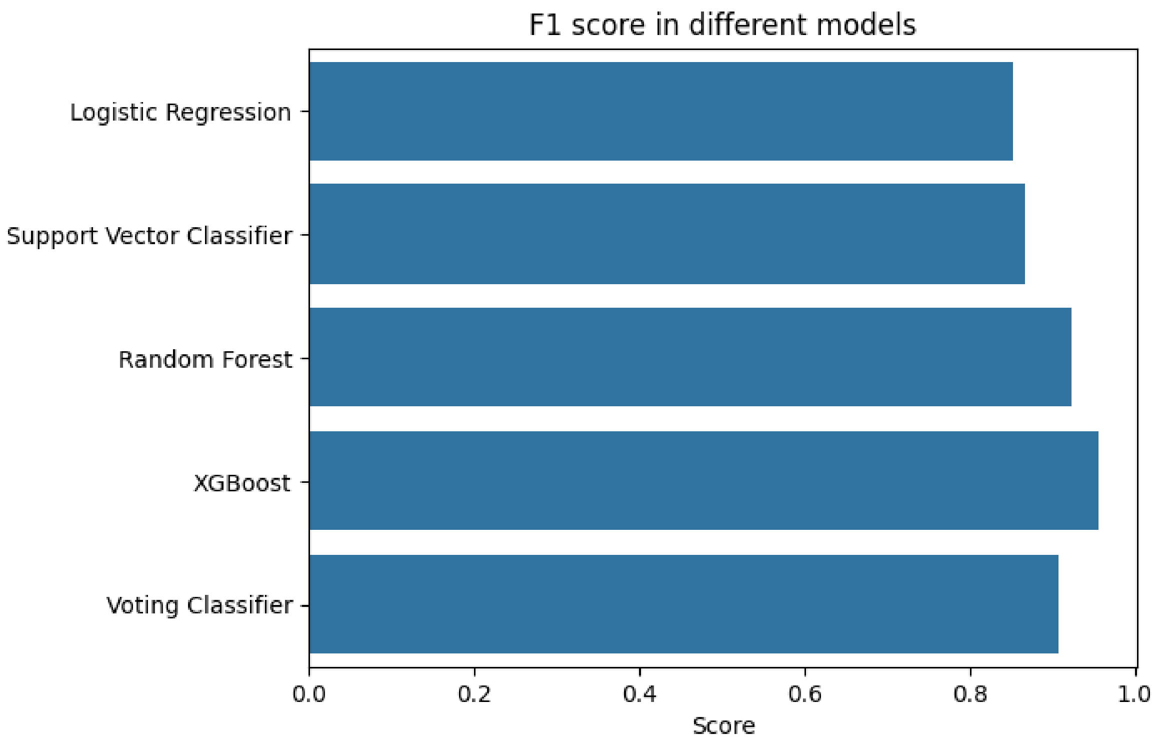
d. XGBoost



e. Voting classifier



As the final result I compared the F1 score of the 5 models, considering F1 as the best metric giving the class imbalance:

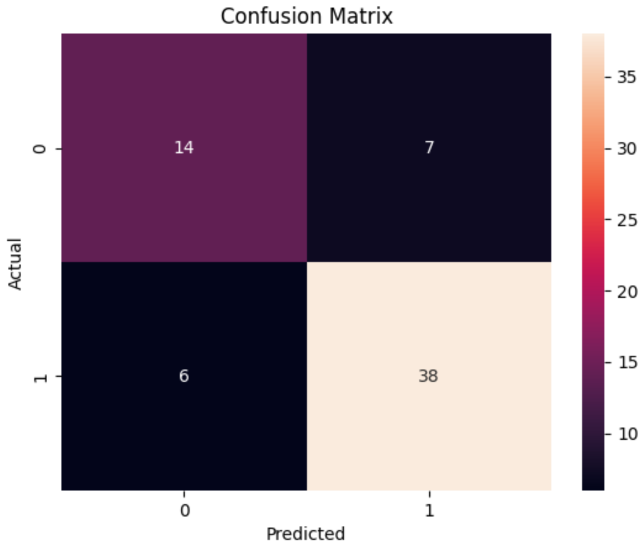


5. Confusion Matrices and Visualizations

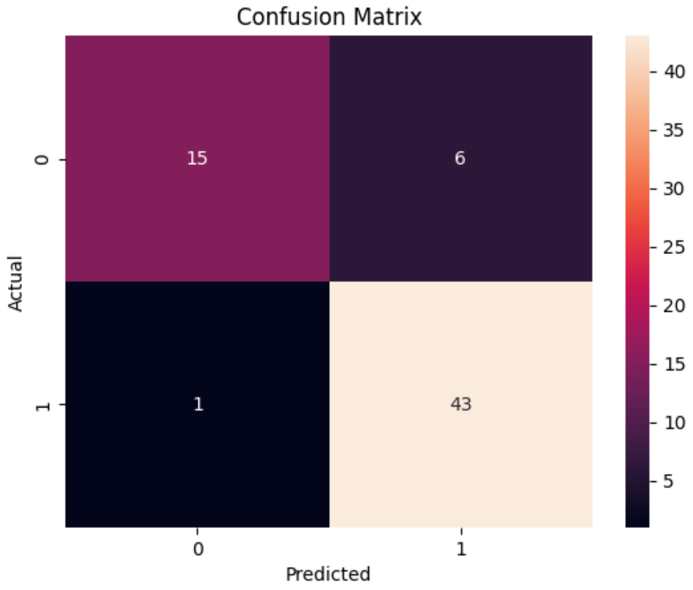
The following visualizations provide insights into model performance:

- **Confusion Matrices**: These illustrate the true positive, true negative, false positive, and false negative counts for each model, helping identify common misclassification patterns.

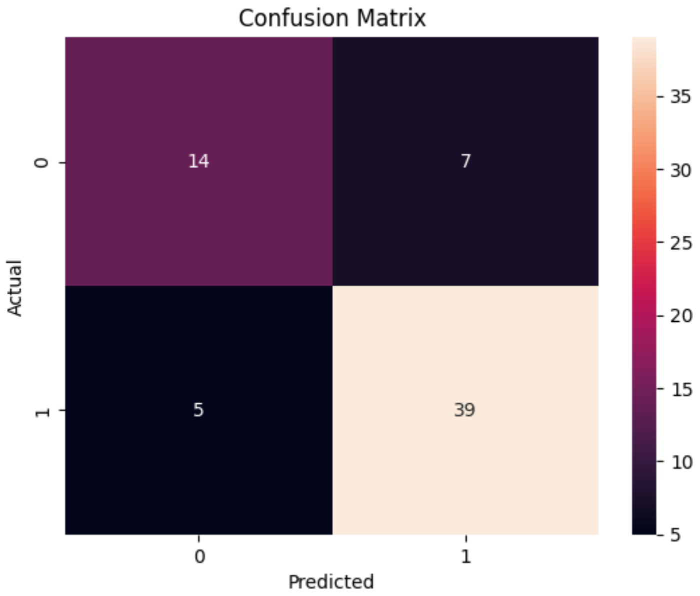
a. Logistic regression



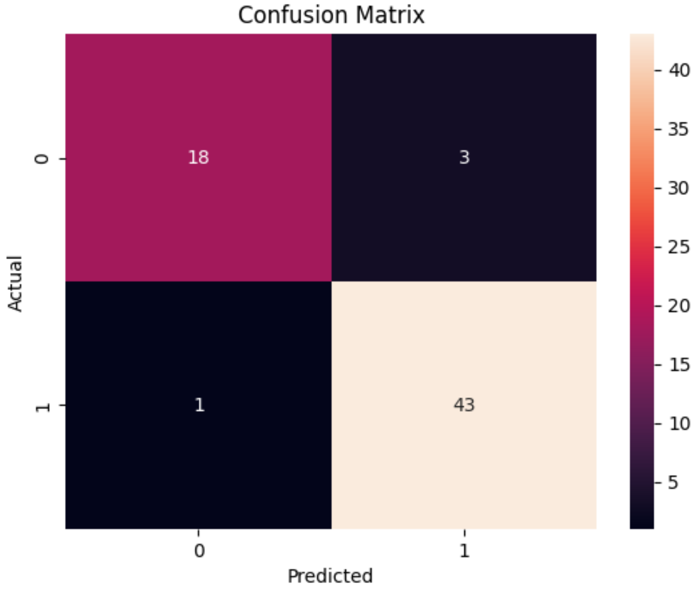
b. Support Vector Machines



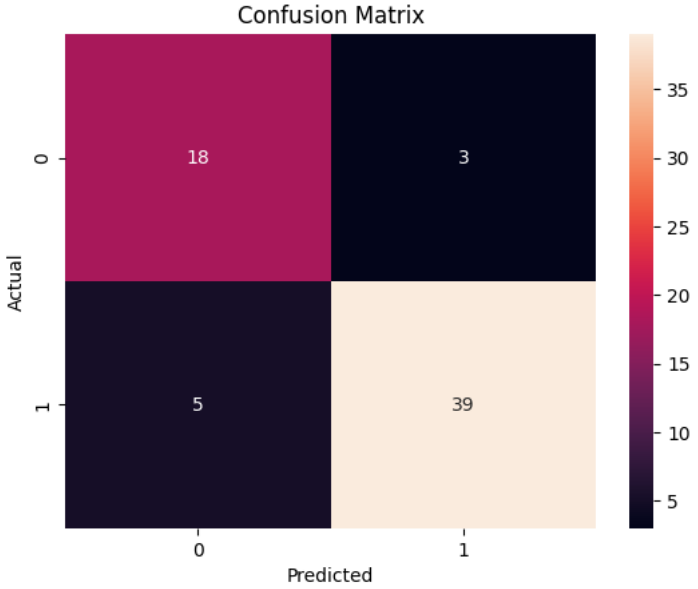
c. Random Forest



d. XGBoost



e. Voting Classifier



6. Conclusion

After comparing the models, the **XGBoost** model emerged as the top performer based on the metrics. This model demonstrated superior accuracy, F1 score, and balanced precision-recall values, indicating it effectively captures the underlying patterns in the data. Additionally, it presents the lowest signs of overfitting getting the closest gaps between the metrics in the training and testing datasets. The performance in the voting classifier is good but models that are not performing really well like Logistic Regression and Support Vector Machines are decreasing the quality of the overall classification, therefore XGBoost model is the best one.