DATA MANAGEMENT AND ARTIFICIAL INTELLIGENCE

A Project on

MACHINE LEARNING ACCIDENT ANALYSIS REPORT

Submitted by

FRANCIS GALLO.SAHAYARAJ

MANASA MANVITHA.MUKKA

Contents:

1. Introduction

The aim of this report is to analyze traffic accident data and predict the likelihood of mortality during an accident. We used machine learning models to understand patterns in the dataset and to predict outcomes based on different features.

2. Dataset Overview

We worked with four main datasets containing information on accidents, users, locations, and vehicles. The datasets were merged into a comprehensive dataset to enable more robust analysis.

Data Sources:

- caract-2023.csv: Accident characteristics
- lieux-2023.csv: Accident locations
- usagers-2023.csv: User information
- vehicules-2023.csv: Vehicle information

The merged dataset had **309,341 rows** and **54 columns** after preprocessing.

3. Data Cleaning and Preprocessing

The cleaning process involved handling missing data, merging datasets, and ensuring proper data types. We:

- Merged all four datasets using a common accident identifier.
- Calculated the percentage of missing data per column.
- Dropped columns with more than 50% missing data.

- Filled the missing values for remaining columns using median (for numeric features) or mode (for categorical features).

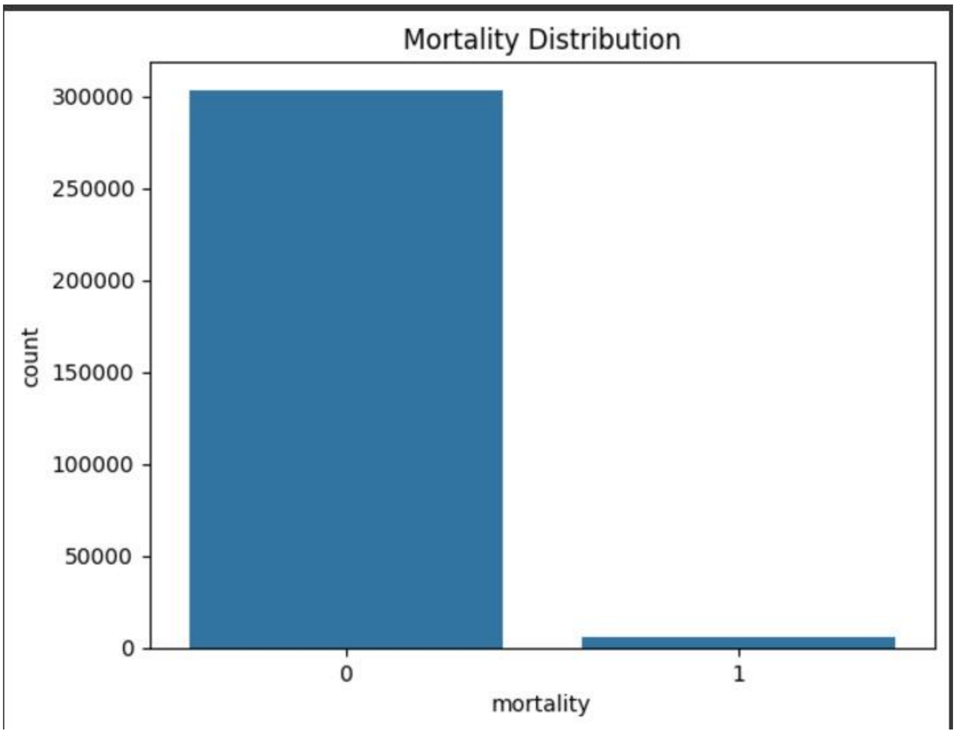
The cleaned dataset had no missing values.

4. Exploratory Data Analysis (EDA)

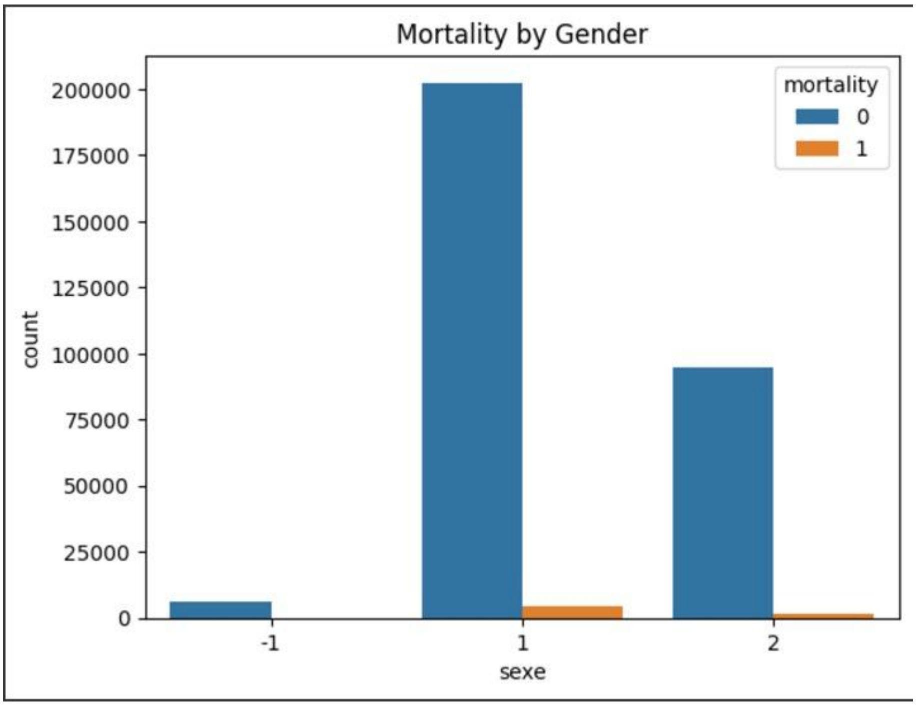
We performed exploratory data analysis to gain insights into accident characteristics and visualize different aspects of the dataset.

Visualizations:

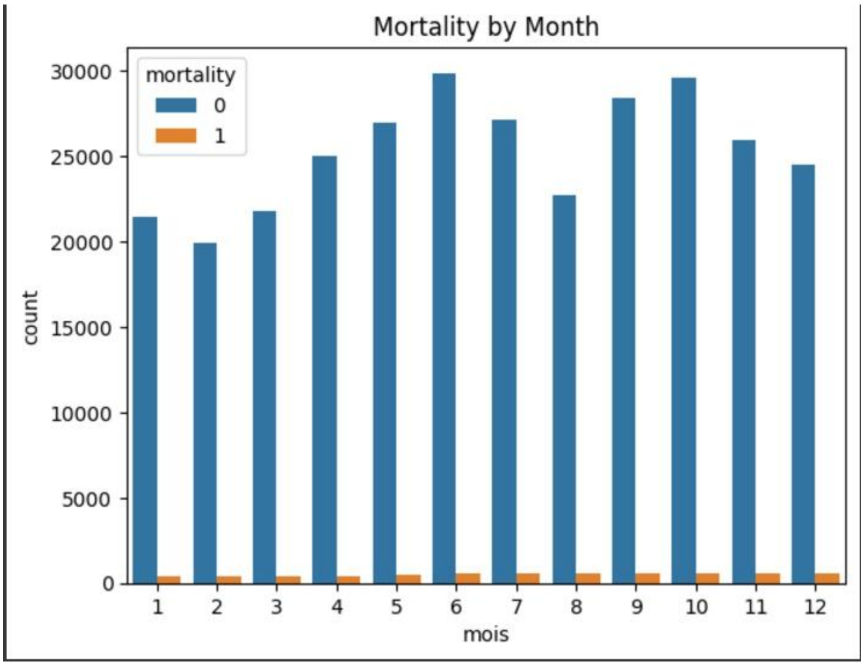
- **Mortality Distribution**:



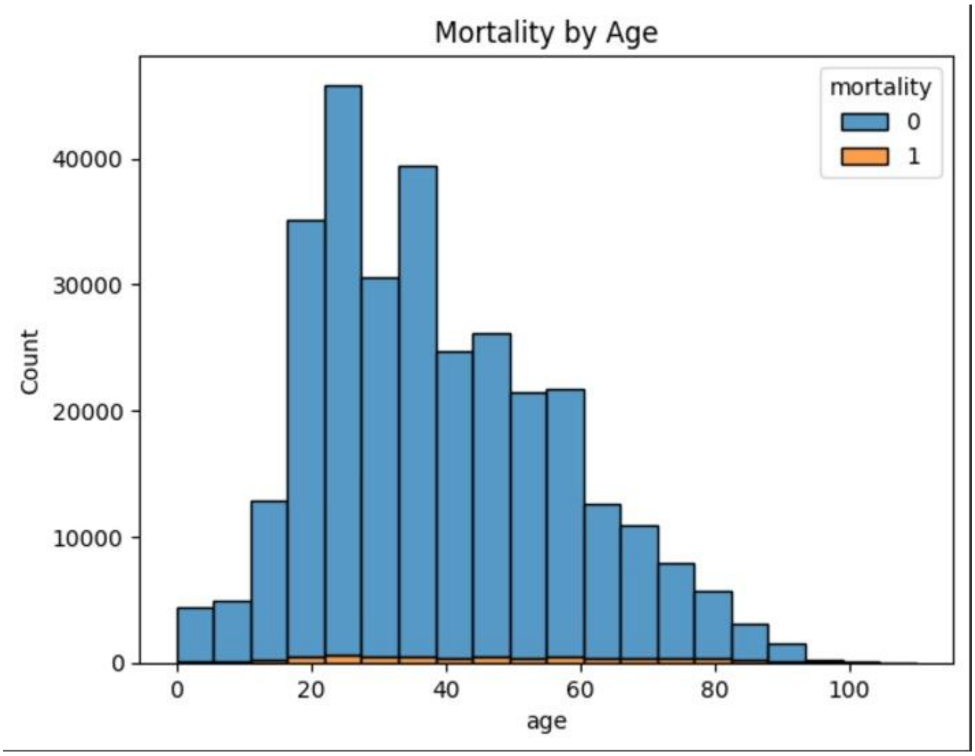
- **Mortality by Gender**:



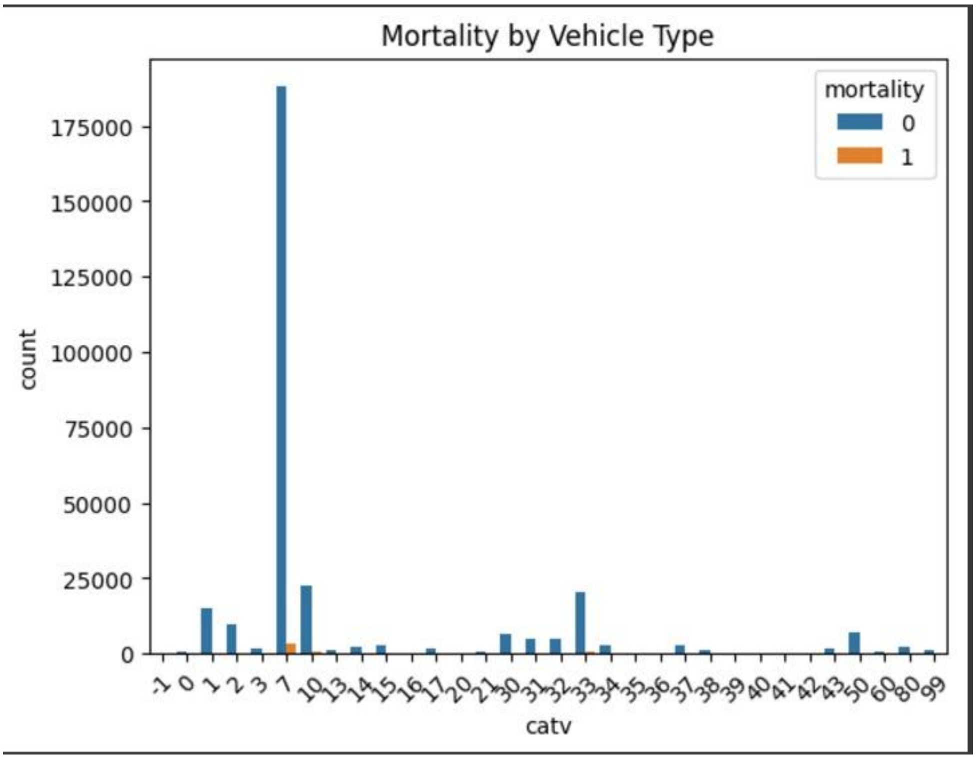
- **Mortality by Month**:



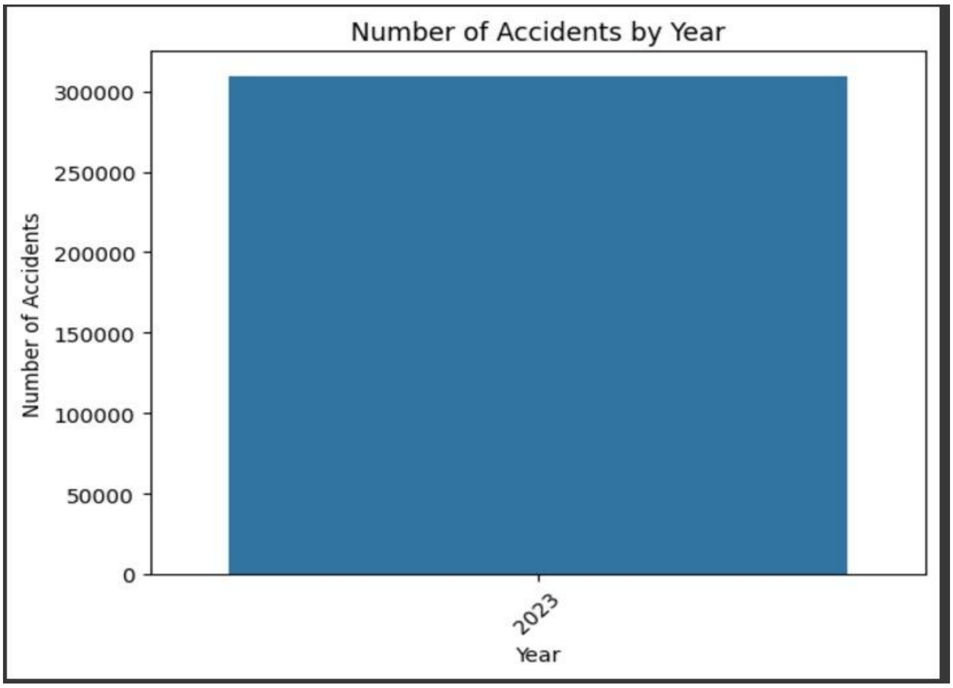
- **Mortality by Age**:



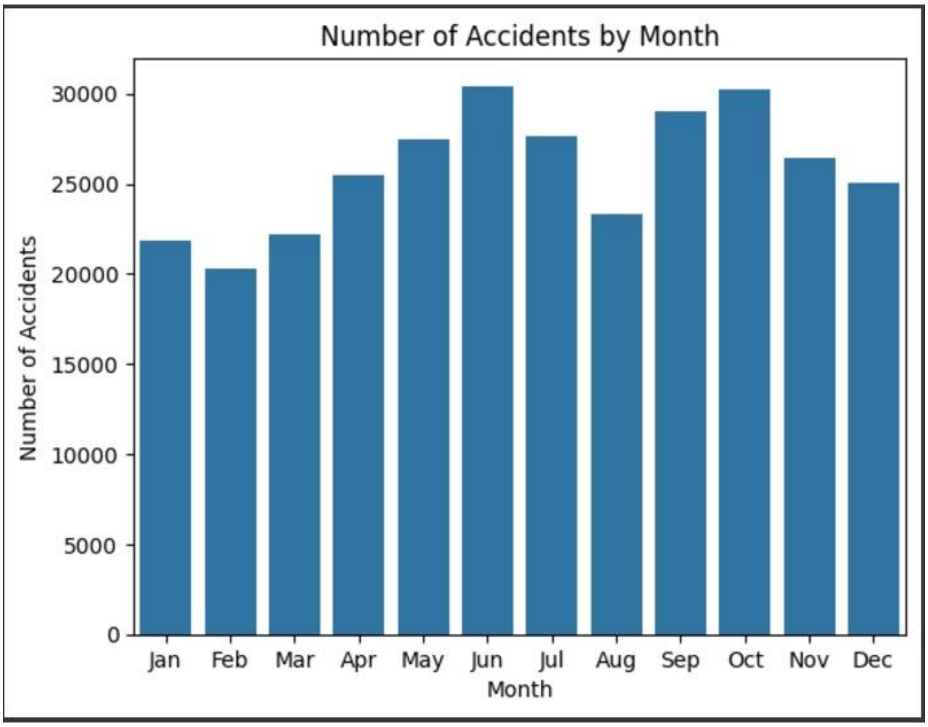
- **Mortality by Vehicle Type**:



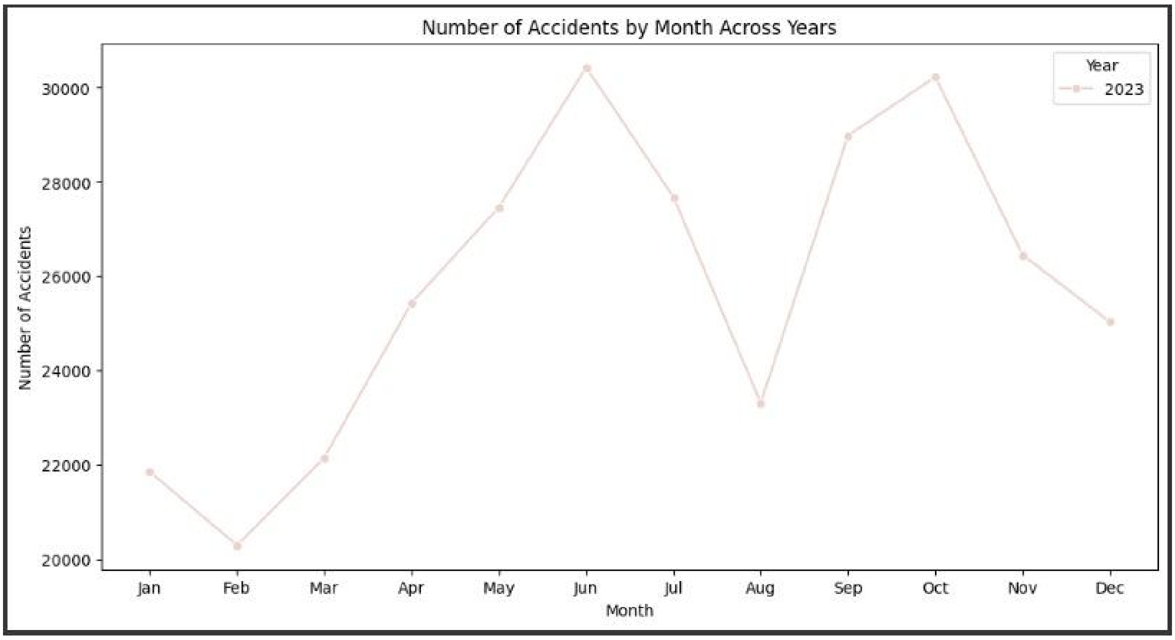
- **Accidents by Year**:



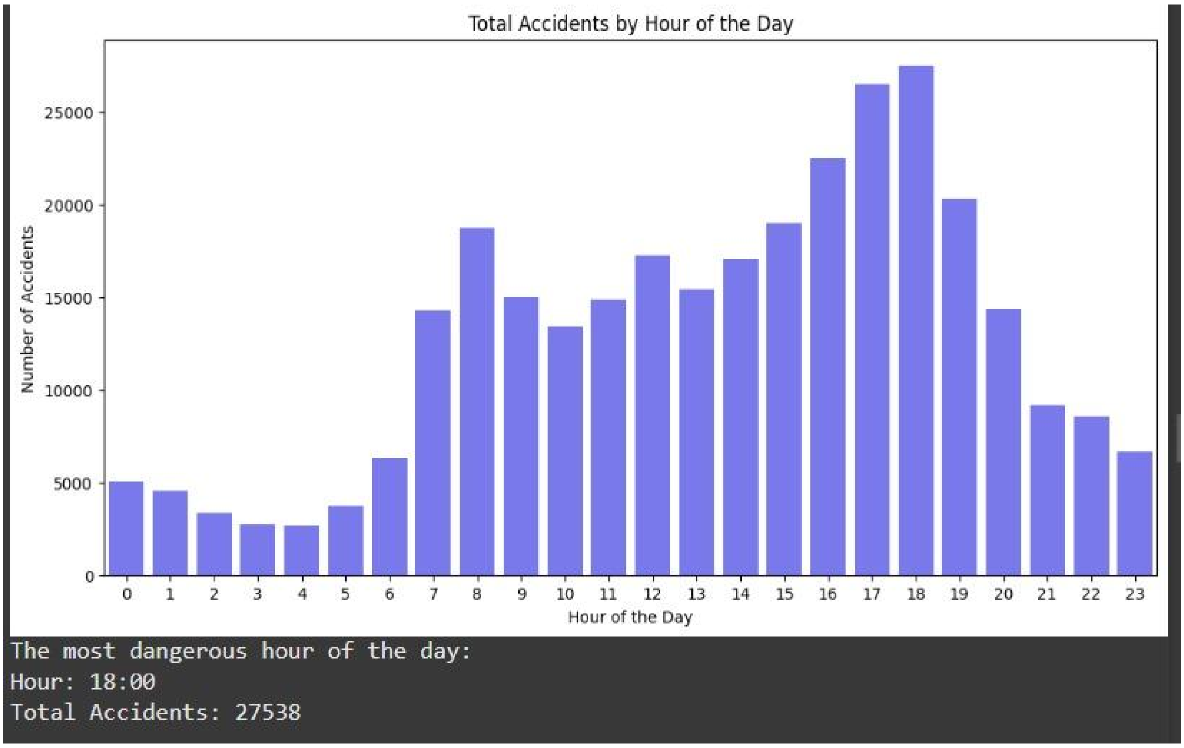
- **Accidents by Month:**



- **Number of Accidents by Month Across Years**:



- **Accidents by Hour**:



```
The most dangerous hour of the day:
Hour: 18:00
Total Accidents: 27538
```

Key Insights:

- Most accidents occurred during daylight conditions, but night conditions with street lighting also had high frequencies.
- The majority of the accidents were linked to male drivers.
- The distribution of accidents varied across different months and hours.

5. Feature Engineering

We engineered a new feature called mortality to indicate if an accident resulted in mortality (1 if grav == 2, else 0). The age of each user was also computed from their year of birth.

6. Machine Learning Models

We built three machine learning models to predict mortality in accidents.

- Logistic Regression
- Random Forest
- Neural Network

Training the model Original data vs Normalized data

Logistic Regression Performance:

- **Original Data**:
  - **Training Accuracy**: 0.9806
  - **Testing Accuracy**: 0.9802

  - **Confusion Matrix**:

    ```
    [[60642     0]
     [ 1227     0]]
    ```

  - **Classification Report**:
    - Precision for Class 0: 0.98
    - Precision for Class 1: 0.00 (indicates issues with class imbalance)
    - **F1 Score for Class 1**: 0.00

- **Normalized Data**:
  - **Training Accuracy**: 0.9806
  - **Testing Accuracy**: 0.9802
  - **Confusion Matrix**:

    ```
    [[60636     6]
     [ 1217    10]]
    ```

  - **Classification Report**:
    - Precision for Class 0: 0.98
    - Precision for Class 1: 0.62 (improved after normalization)
    - **F1 Score for Class 1**: 0.02

Summary:

- Normalizing the data slightly improved precision and recall for Class 1.

- Logistic Regression had difficulty in predicting the minority class in both original and normalized versions.

Random Forest Performance:

- **Original Data**:
    - **Training Accuracy**: 1.0000
    - **Testing Accuracy**: 1.0000
    - **Confusion Matrix**:

      ```
      [[60642     0]
       [    2  1225]]
      ```

    - **Classification Report**:
        - Precision for Class 0: 1.00
        - Precision for Class 1: 1.00
        - **F1 Score for Class 1**: 1.00
- **Normalized Data**:
    - **Training Accuracy**: 1.0000
    - **Testing Accuracy**: 1.0000
    - **Confusion Matrix**:

      ```
      [[60642     0]
       [    1  1226]]
      ```

    - **Classification Report**:
        - Precision for Class 0: 1.00

        - Precision for Class 1: 1.00
        - **F1 Score for Class 1**: 1.00

Summary:

- **Potential Overfitting**: Random Forest showed identical training and testing accuracy of 1.0000, suggesting overfitting.
- **Normalization Impact**: No significant difference between original and normalized data performances.

Neural Network Performance:

- **Original Data**:
    - **Training Accuracy**: 0.9806
    - **Testing Accuracy**: 0.9802
    - **Confusion Matrix**:

      ```
      [[60642     0]
       [ 1227     0]]
      ```

    - **Classification Report**:
        - Precision for Class 0: 0.98
        - Precision for Class 1: 0.00
        - **F1 Score for Class 1**: 0.00
- **Normalized Data**:
    - **Training Accuracy**: 1.0000
    - **Testing Accuracy**: 1.0000

- **Confusion Matrix**:

  ```
  [[60642     0]
   [    0  1227]]
  ```

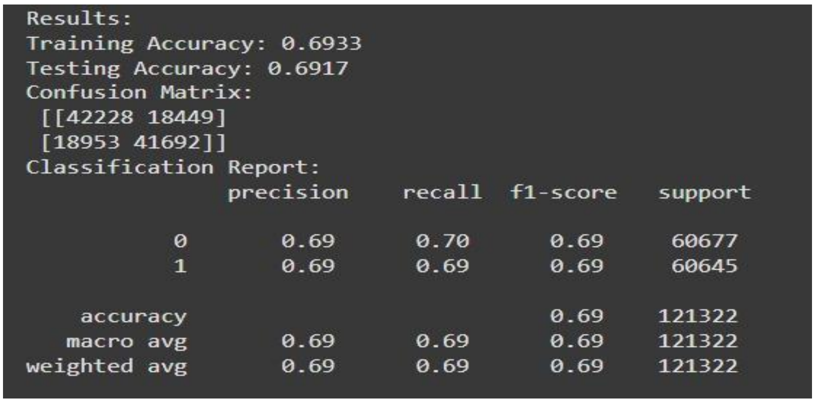
- **Classification Report**:
  - Precision for Class 0: 1.00
  - Precision for Class 1: 1.00
  - **F1 Score for Class 1**: 1.00

Summary:

- **Performance Improvement**: Normalizing the data significantly improved the precision and recall for the minority class (Class 1).
- **Overfitting**: Similar to Random Forest, Neural Network showed overfitting with identical training and testing accuracies of 1.0000.

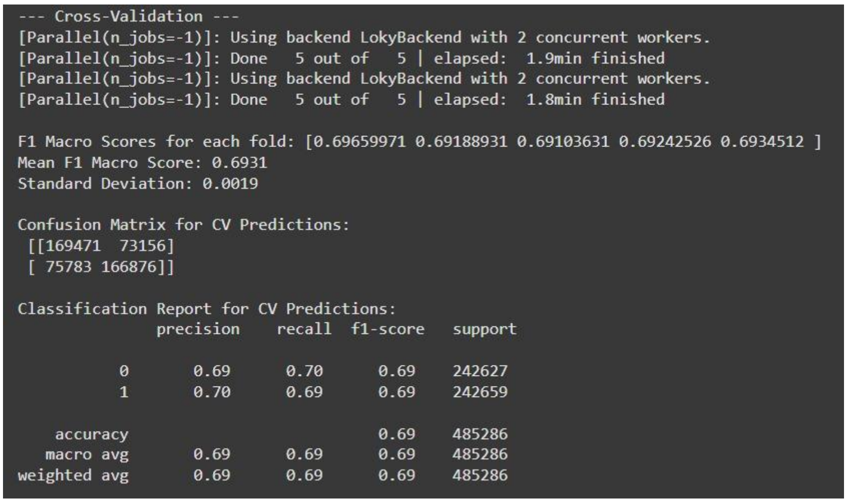
A. Logistic Regression

- **SMOTE Applied Data**:
  - Training Accuracy: 0.6933 Testing Accuracy: 0.6917



```
Results:
Training Accuracy: 0.6933
Testing Accuracy: 0.6917
Confusion Matrix:
 [[42228 18449]
 [18953 41692]]
Classification Report:
              precision    recall  f1-score   support

           0       0.69      0.70      0.69     60677
           1       0.69      0.69      0.69     60645

    accuracy                           0.69    121322
   macro avg       0.69      0.69      0.69    121322
weighted avg       0.69      0.69      0.69    121322
```

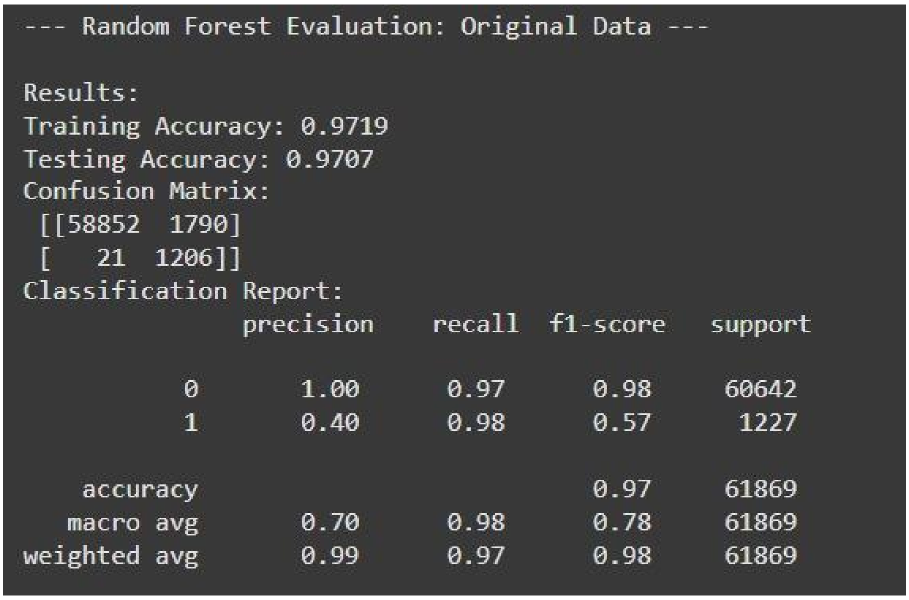
- **Cross Validation**:



```
--- Cross-Validation ---
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 2 concurrent workers.
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:  1.9min finished
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 2 concurrent workers.
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:  1.8min finished

F1 Macro Scores for each fold: [0.69659971 0.69188931 0.69103631 0.69242526 0.6934512 ]
Mean F1 Macro Score: 0.6931
Standard Deviation: 0.0019

Confusion Matrix for CV Predictions:
 [[169471  73156]
 [ 75783 166876]]

Classification Report for CV Predictions:
              precision    recall  f1-score   support

           0       0.69      0.70      0.69    242627
           1       0.70      0.69      0.69    242659

    accuracy                           0.69    485286
   macro avg       0.69      0.69      0.69    485286
weighted avg       0.69      0.69      0.69    485286
```

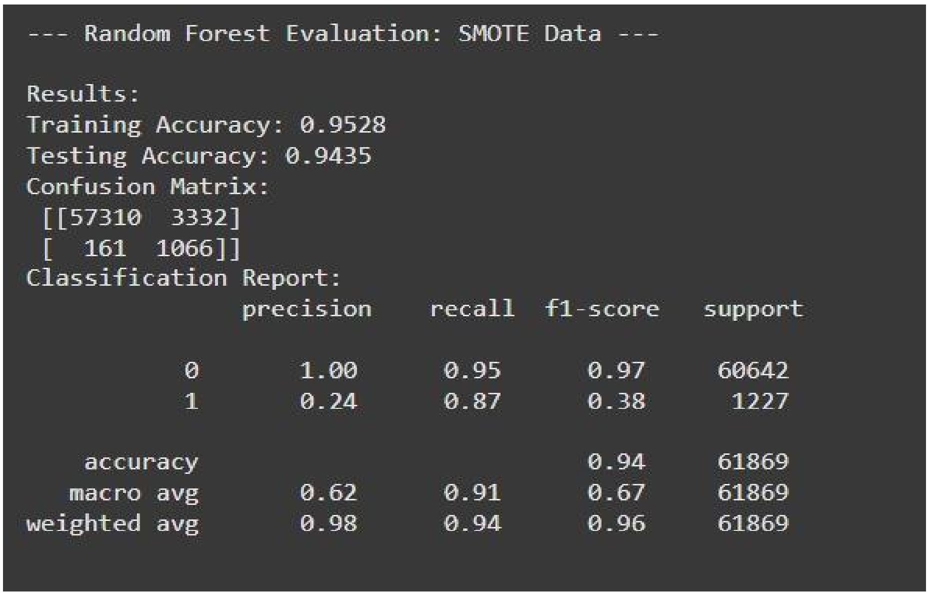
B. Random Forest

- **Original Data**:
  - Training Accuracy: 0.9719 Testing Accuracy: 0.9707



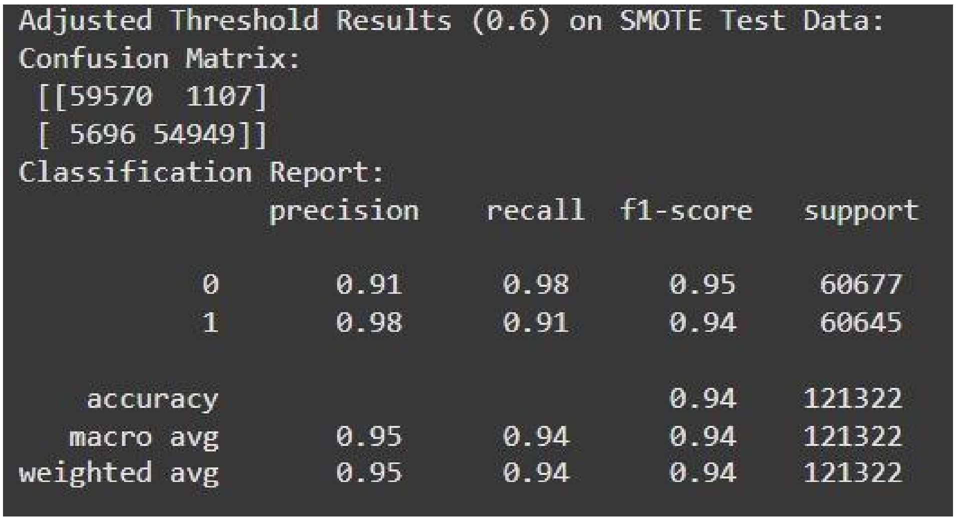
```
--- Random Forest Evaluation: Original Data ---

Results:
Training Accuracy: 0.9719
Testing Accuracy: 0.9707
Confusion Matrix:
 [[58852  1790]
 [   21  1206]]
Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.97      0.98     60642
           1       0.40      0.98      0.57      1227

    accuracy                           0.97     61869
   macro avg       0.70      0.98      0.78     61869
weighted avg       0.99      0.97      0.98     61869
```

- **SMOTE Data**:
  - Training Accuracy: 0.9528 Testing Accuracy: 0.9435



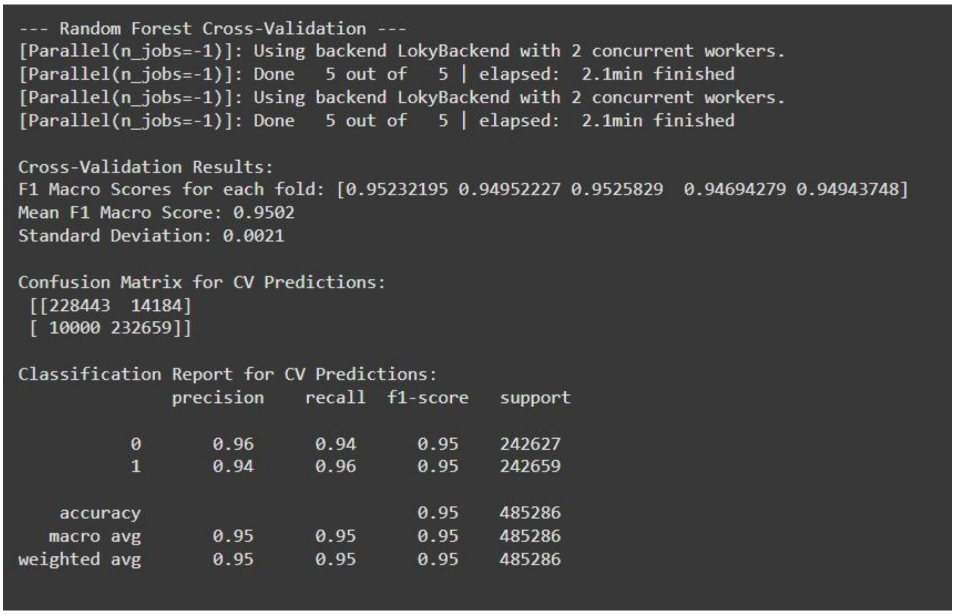
```
--- Random Forest Evaluation: SMOTE Data ---

Results:
Training Accuracy: 0.9528
Testing Accuracy: 0.9435
Confusion Matrix:
 [[57310  3332]
 [  161  1066]]
Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.95      0.97     60642
           1       0.24      0.87      0.38      1227

    accuracy                           0.94     61869
   macro avg       0.62      0.91      0.67     61869
weighted avg       0.98      0.94      0.96     61869
```

- **Adjusted Threshold**:
    - Threshold adjusted to 0.6 for better precision-recall balance.



```
Adjusted Threshold Results (0.6) on SMOTE Test Data:
Confusion Matrix:
 [[59570  1107]
 [ 5696 54949]]
Classification Report:
              precision    recall  f1-score   support

           0       0.91      0.98      0.95     60677
           1       0.98      0.91      0.94     60645

    accuracy                           0.94    121322
   macro avg       0.95      0.94      0.94    121322
weighted avg       0.95      0.94      0.94    121322
```

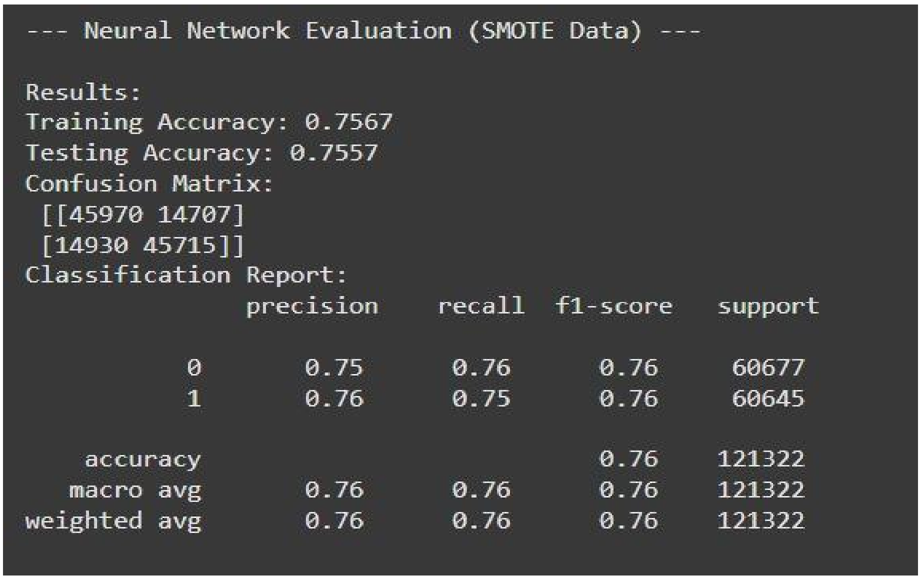
- **Cross Validation**:



```
--- Random Forest Cross-Validation ---
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 2 concurrent workers.
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:  2.1min finished
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 2 concurrent workers.
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:  2.1min finished

Cross-Validation Results:
F1 Macro Scores for each fold: [0.95232195 0.94952227 0.9525829  0.94694279 0.94943748]
Mean F1 Macro Score: 0.9502
Standard Deviation: 0.0021

Confusion Matrix for CV Predictions:
 [[228443  14184]
 [ 10000 232659]]

Classification Report for CV Predictions:
              precision    recall  f1-score   support

           0       0.96      0.94      0.95    242627
           1       0.94      0.96      0.95    242659

    accuracy                           0.95    485286
   macro avg       0.95      0.95      0.95    485286
weighted avg       0.95      0.95      0.95    485286
```

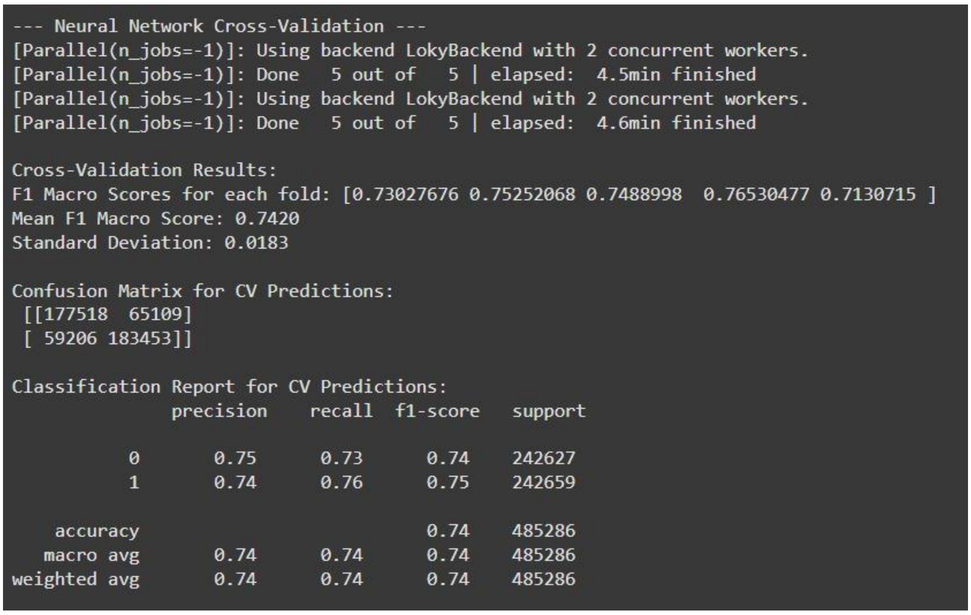
C. Neural Network

- **SMOTE Applied Data**:
  - Training Accuracy: 0.7567 Testing Accuracy: 0.7557



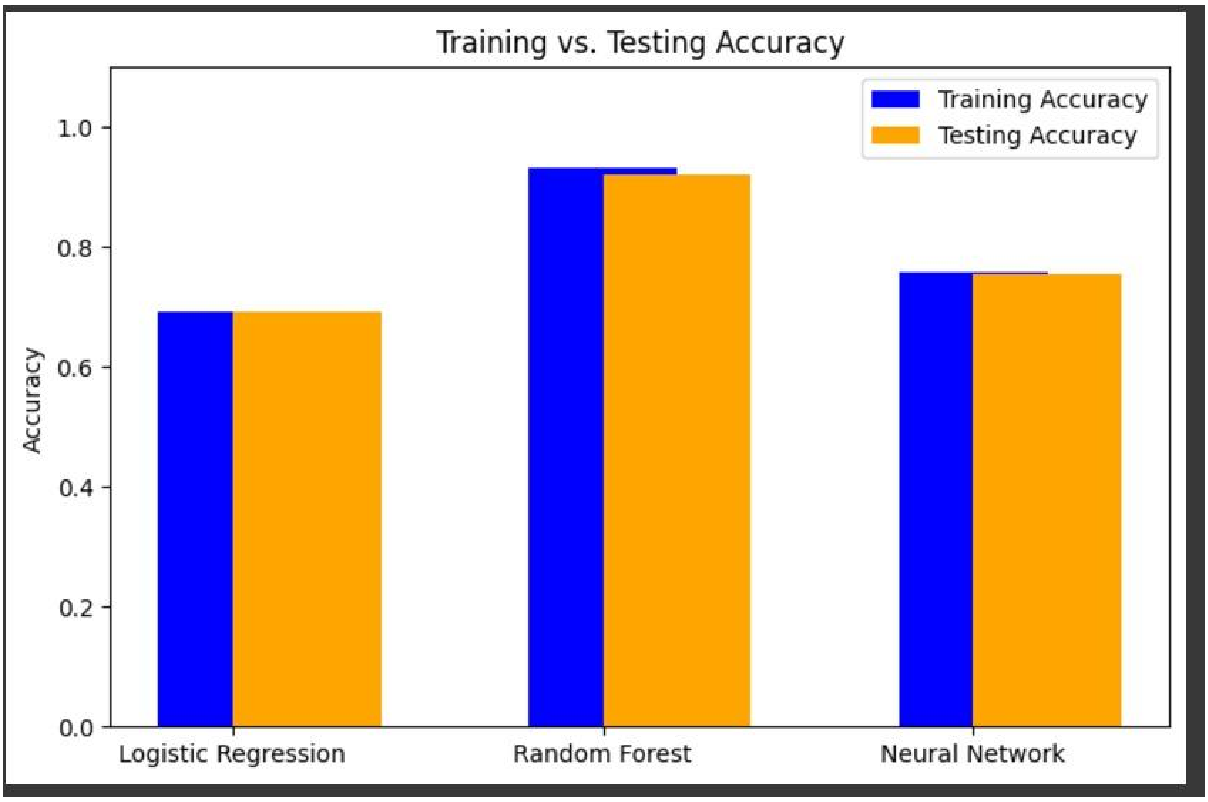
```
--- Neural Network Evaluation (SMOTE Data) ---

Results:
Training Accuracy: 0.7567
Testing Accuracy: 0.7557
Confusion Matrix:
 [[45970 14707]
 [14930 45715]]
Classification Report:
              precision    recall  f1-score   support

           0       0.75      0.76      0.76     60677
           1       0.76      0.75      0.76     60645

    accuracy                           0.76    121322
   macro avg       0.76      0.76      0.76    121322
weighted avg       0.76      0.76      0.76    121322
```

- **Cross Validation**:



```
--- Neural Network Cross-Validation ---
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 2 concurrent workers.
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:  4.5min finished
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 2 concurrent workers.
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:  4.6min finished

Cross-Validation Results:
F1 Macro Scores for each fold: [0.73027676 0.75252068 0.7488998  0.76530477 0.7130715 ]
Mean F1 Macro Score: 0.7420
Standard Deviation: 0.0183

Confusion Matrix for CV Predictions:
 [[177518  65109]
 [ 59206 183453]]

Classification Report for CV Predictions:
              precision    recall  f1-score   support

           0       0.75      0.73      0.74    242627
           1       0.74      0.76      0.75    242659

    accuracy                           0.74    485286
   macro avg       0.74      0.74      0.74    485286
weighted avg       0.74      0.74      0.74    485286
```

7.Comparison between 3 models



8. Comparison of Cross-Validation Results

We performed cross-validation on all three models to evaluate the stability of model performance.

A. Logistic Regression

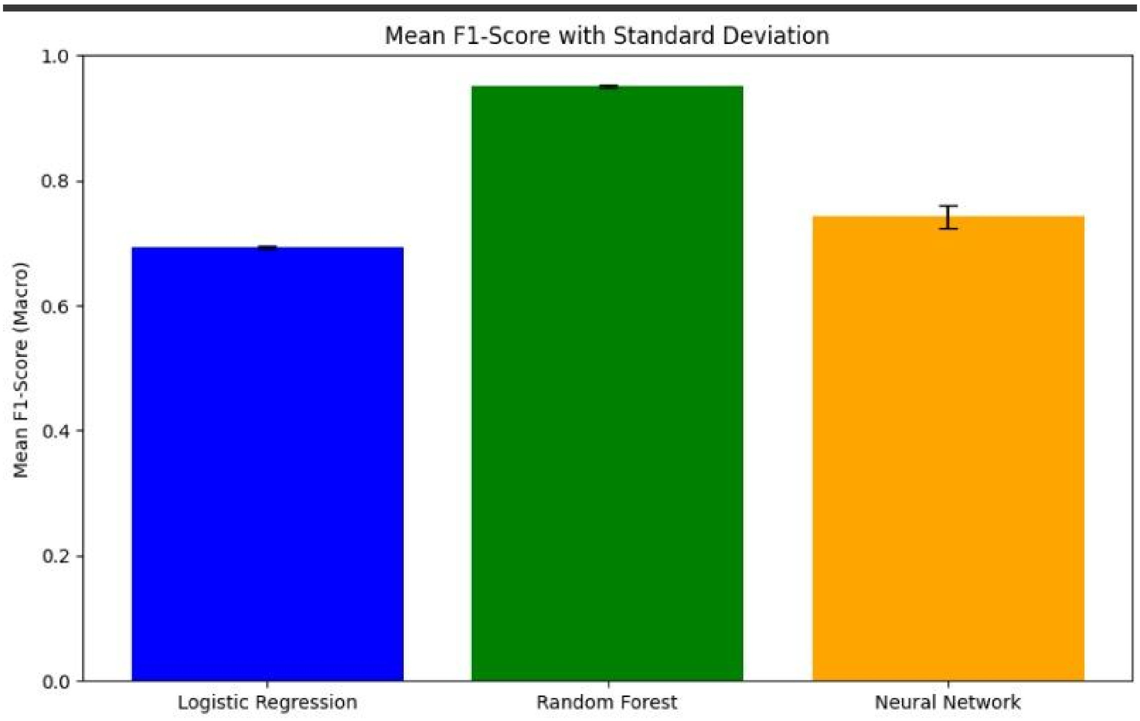
- **Mean F1 Macro Score**: 0.6931

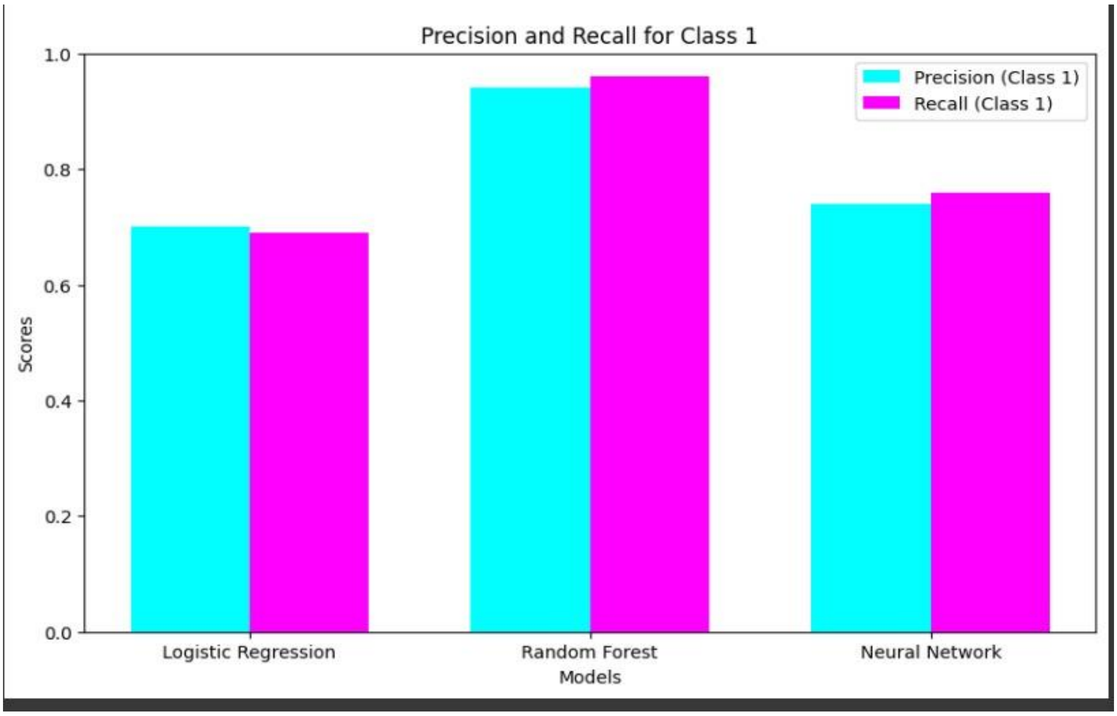
B. Random Forest

- **Mean F1 Macro Score**: 0.9502

C. Neural Network

- **Mean F1 Macro Score**: 0.7420


Mean F1-Score with Standard Deviation
Mean F1-Score (Macro)
1.0
0.8
0.6
0.4
0.2
0.0
Logistic Regression
Random Forest
Neural Network


Precision and Recall for Class 1
Precision (Class 1)
Recall (Class 1)
Scores
1.0
0.8
0.6
0.4
0.2
0.0
Logistic Regression
Random Forest
Neural Network
Models

9. Conclusion

- The **Random Forest model** showed the best performance, especially after hyperparameter tuning.
- There are indications of overfitting in some cases, particularly with the Random Forest model's high training accuracy and testing accuracy being very close to 1.0.
- **SMOTE** helped balance the classes and improve model robustness.

Recommendations:

- Further tuning and potentially more complex models, like **Gradient Boosting** or **XGBoost**, might help to avoid overfitting while maintaining high predictive power.
- Adjusting class weights and using techniques like **early stopping** could improve the Neural Network's performance.