Data Science Project Report

Credit Card Fraud Detection

Instructor : Dr Nouman Durrani

Group Members: K21-4878 Sarosh Irfan
K21-3261 Umer tariq
K21-4933 Ibrahim Jawaid

Introduction

In our data science project focused on detecting credit card fraudulence, we've undertaken the task of evaluating six distinct classification models: **Support Vector Classifier (SVC), K-Nearest Neighbors (KNN), Random Forest, Decision Tree, Naive Bayes, and Logistic Regression**. Each model has been meticulously fine-tuned through hyperparameter optimization to extract its maximum potential. We've ensured consistency by maintaining a fixed random state and size for the test-train split across all models. Now, the critical phase lies in determining the most effective model for our fraud detection task.

Methodology

- Initially, we cleaned and preprocessed the dataset by:
  - Checking for null values.
  - Ensuring consistent data types.
  - Dropping unnecessary columns to reduce dimensionality.
  - Removing duplicate rows.
  - Scaling and normalizing columns like the amount.
  - Identifying data imbalance.

- For each model, we implemented a systematic approach to hyperparameter tuning. This involved techniques such as grid search or random search to explore the hyperparameter space efficiently.

- We evaluated each model's performance using various metrics, including accuracy, recall, precision, and F1-score.

- We constructed confusion matrices to gain insights into the models' predictive capabilities, particularly in distinguishing between fraudulent and non-fraudulent transactions. These matrices provided a visual representation of true positives, false positives, true negatives, and false negatives, aiding in our model selection process.

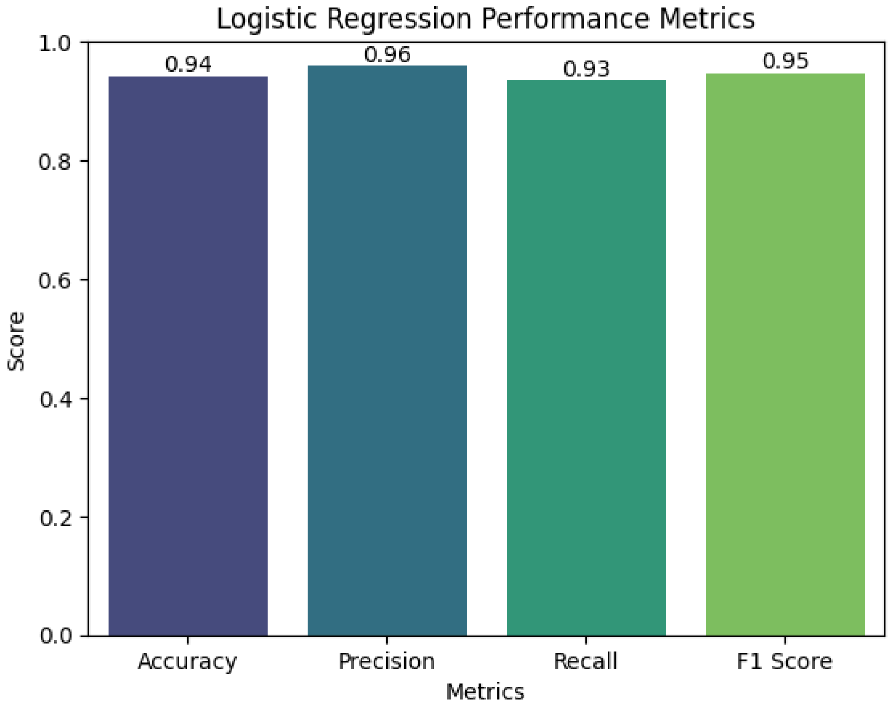
- The most challenging aspect was addressing the data imbalance, with 99% of the data representing legitimate transactions and only 1% indicating fraud. To tackle this issue, we employed the Synthetic Minority Over-sampling Technique (SMOTE)

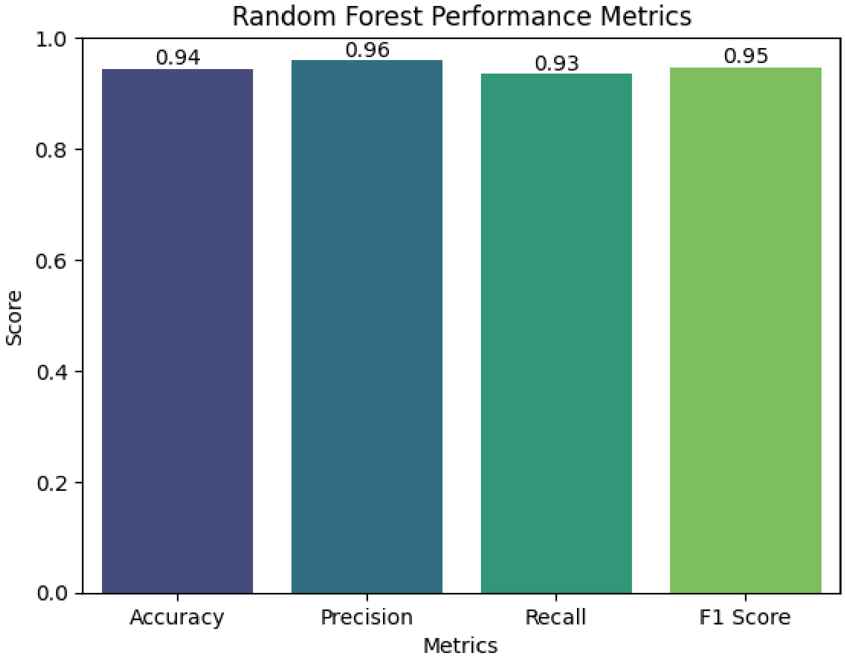
- SMOTE works by generating synthetic samples in the minority class (fraudulent transactions) to balance the dataset, thereby improving the model's ability to learn from both classes effectively.

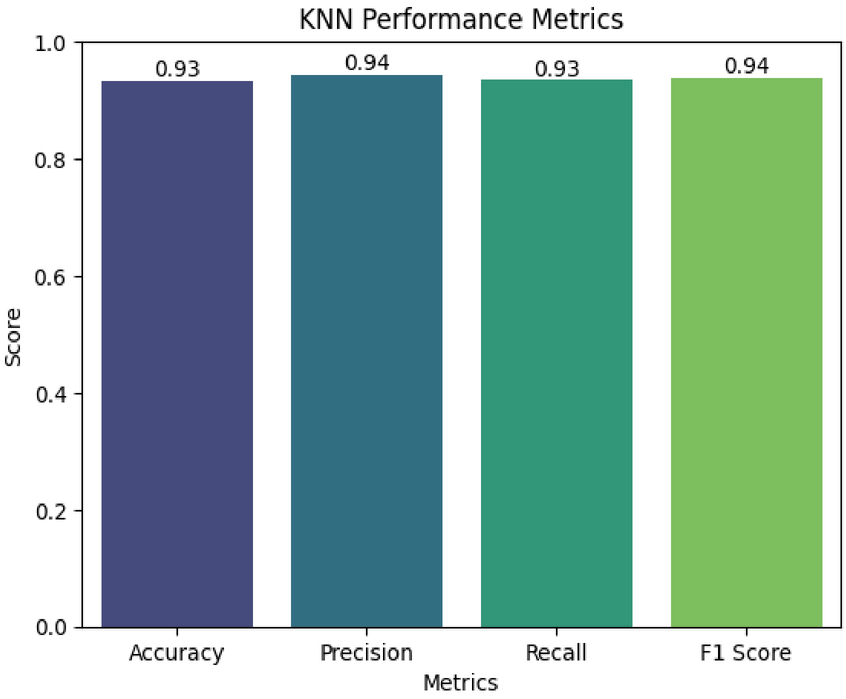
- We set `random_state=42` and `np.random.seed(42)` to keep the test conditions consistent for all runs.

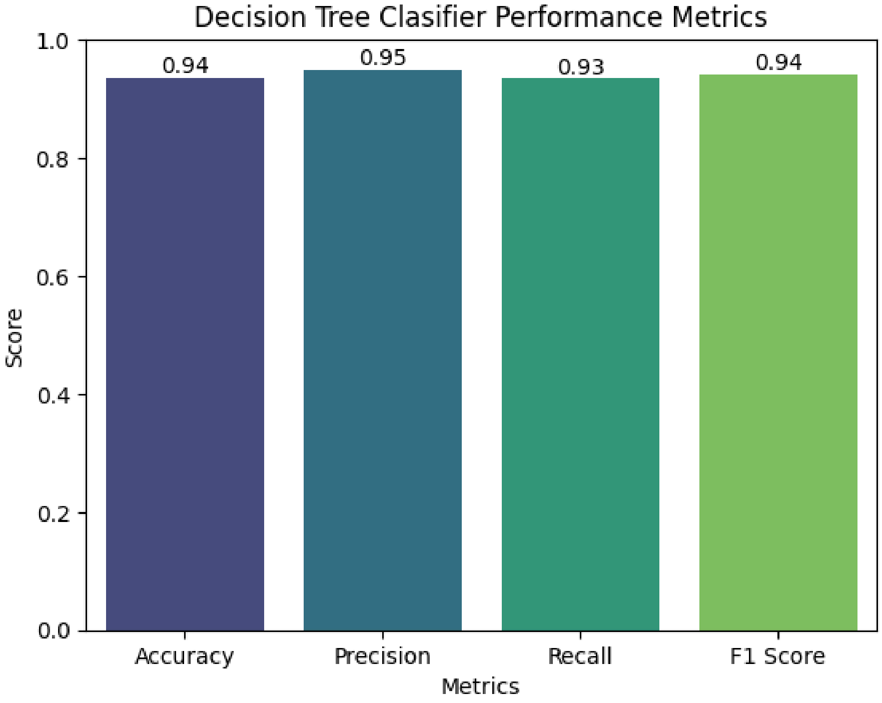
- To choose the best model, we considered two factors:
    - High and Consistent Performance:
        - We evaluated models based on their F1 score, accuracy, recall, and precision. A high and consistent score across these metrics indicates a reliable model that performs well overall.
    - Minimization of False Positives and False Negatives:
        - We also considered the number of false positives and false negatives produced by each model. Minimizing false positives reduces the risk of falsely accusing innocent individuals, while minimizing false negatives reduces the risk of guilty individuals being freed.

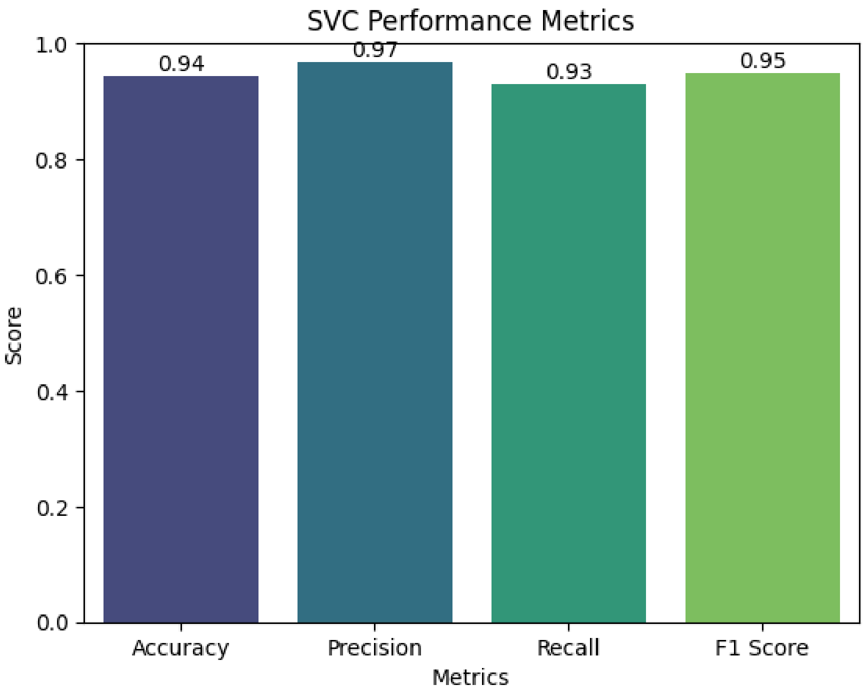
Graphical Results of All Models

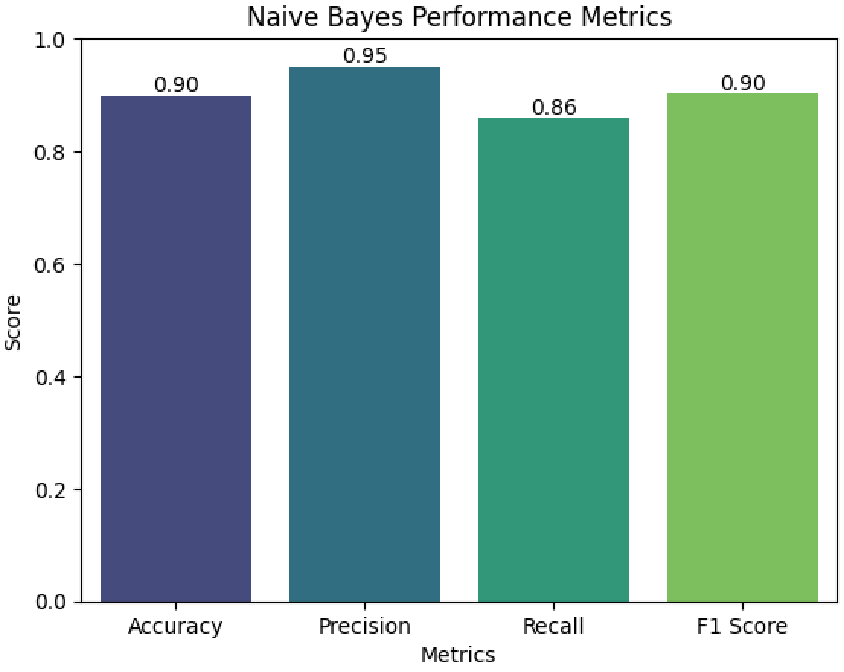


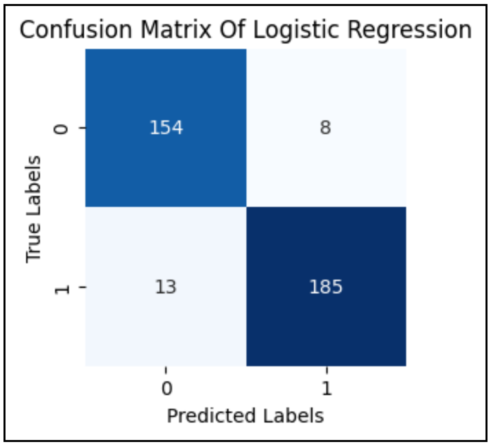


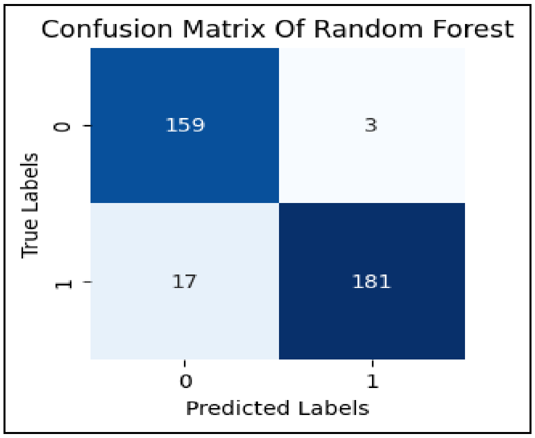

KNN Performance Metrics
1.0
0.8
0.6
0.4
0.2
0.0
Score
0.93
0.94
0.93
0.94
Accuracy
Precision
Recall
F1 Score
Metrics

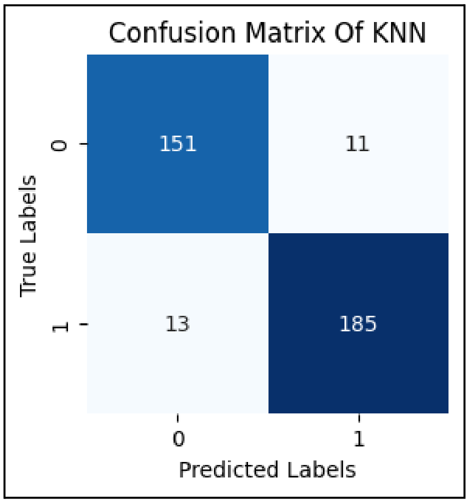

Decision Tree Clasifier Performance Metrics
1.0
0.8
0.6
0.4
0.2
0.0
Score
0.94
0.95
0.93
0.94
Accuracy
Precision
Recall
F1 Score
Metrics

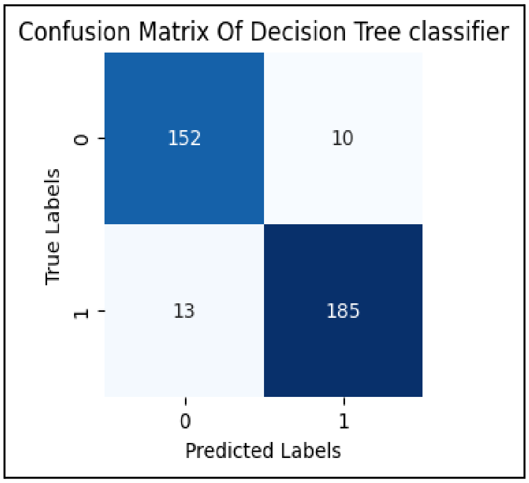

SVC Performance Metrics
1.0
0.8
0.6
0.4
0.2
0.0
Score
0.94
0.97
0.93
0.95
Accuracy
Precision
Recall
F1 Score
Metrics

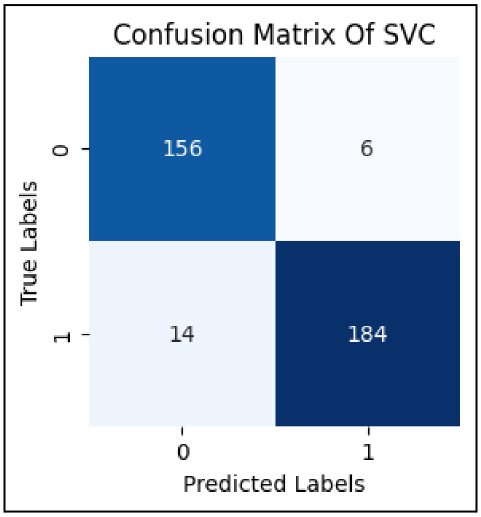

Naive Bayes Performance Metrics
1.0
0.8
0.6
0.4
0.2
0.0
Score
0.90
0.95
0.86
0.90
Accuracy
Precision
Recall
F1 Score
Metrics

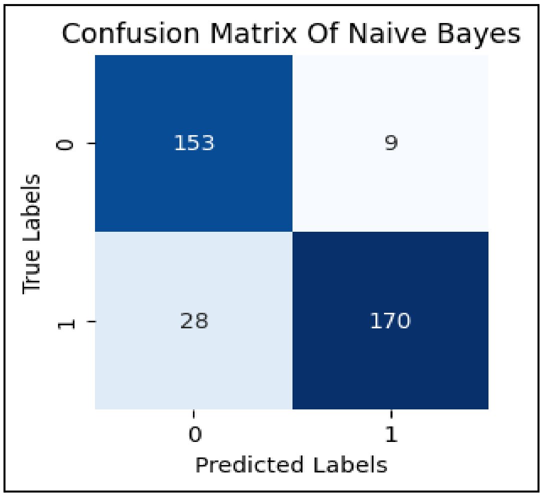

Confusion Matrix Of Logistic Regression


Confusion Matrix Of Random Forest


Confusion Matrix Of KNN


Confusion Matrix Of Decision Tree classifier


Confusion Matrix Of SVC


Confusion Matrix Of Naive Bayes