DATA603– Principles of Machine Learning
Final Term Project Report

Credit Card Fraud Detection

Bala Swapnika Gopi

RESEARCH QUESTION

1. **Describe the research question**
   Which model will be best suited for classification of credit card fraud transactions?

   The most important goal of the research question was to compare and determine which one among the machine learning models proposed is the best model to detect fraudulent transactions in the dataset of credit card transactions. Fraud detection is a key problem in financial systems; it can lead to large financial losses due to fraudsters. Solving this problem can help ensure better security and trust in payment systems.

2. **What were you trying to find in the dataset?**
   The key aim behind the analysis of the dataset was first to identify a pattern or characteristic that differentiates fraudulent transactions from non-fraudulent. Precisely, we set out to:
   a. Understand which features are the most contributing to fraud identification.
   b. Evaluate the performance of machine learning models in effectively classifying fraudulent transactions.
   c. The performance metrics-accuracy, precision, recall, and F1-score-are measured to determine the best-suited model for fraud classification
   d. Address the challenge of class imbalance by implementing appropriate preprocessing techniques, like SMOTE.

3. **Why is this problem important?**
   The problem of credit card fraud detection is very critical because of several reasons:
   Fraud detection accuracy is one of the key factors that determine customer trust in a digital payment system. The timely detection of fraudulent transactions prevents unauthorized access to customers' funds and thus supports proactive fraud management. Credit cards and online payments are being increasingly used, and hence fraudulent activities are also increasing both in frequency and sophistication; therefore, automated detection systems are a must.

4. **How did you formulate the problem?**
   The problem was formulated as a binary classification problem where credit card transactions needed to be classified as fraudulent (Class 1) or non-fraudulent (Class 0). The following are the formulation steps: 1. Understanding the Dataset, 2: Defining the Objective, 3. Key Challenges, 4: Choice of Machine Learning Models, 5: Performance Metrics

5. **Which ML Task you used?**
   The ML task used in this project was binary classification. Binary classification is a supervised machine learning task where the goal is to categorize data into two distinct classes. In this case, the task was to classify credit card transactions into two categories: Class 0:(Non-Fraudulent Transactions), Class 1(Fraudulent Transactions). The classification models used for this task included:
   Support Vector Classifier (SVC), Decision Tree Classifier, Logistic Regression, K-Nearest Neighbors (KNN).

DATASET

1. **Describe the dataset you have used**
   This dataset consists of a set of real credit card transactions that were made by European cardholders. The data is extremely imbalanced, with a class distribution of: Non-fraudulent transactions (Class 0): 99.83% of the data; Fraudulent transactions (Class 1): 0.17% of the data.

2. **Modality, sample size, features, and labels**
   - **Modality:** Tabular data.
   - **Sample size:** 284,807 rows and 31 columns.
   - **Features:** 30 numerical features (V1 to V28), 'Time,' and 'Amount.'
   - **Labels:** The target variable 'Class' (0 = Non-fraudulent, 1 = Fraudulent).

3. **How is the dataset collected**
   The dataset was collected and made available by researchers from European cardholders, where the data was anonymized for privacy reasons. The transactions were collected over a period of two days, and the features (V1 to V28) were generated through Principal Component Analysis (PCA) to protect sensitive financial information.

4. **Why is this dataset important**
   The dataset is very realistic in nature, and the feature distribution of the real data does indeed reflect the characteristics typical of credit card transactions. The features are transaction amounts, time of transactions, and anonymized features of various natures representing a transaction, for example. The positive class (fraudulent transactions) represents a minority of the whole dataset, which is the usual case in real-world fraud scenarios. The creditcard.csv dataset anonymizes sensitive information like customer identity and transaction details, thus ensuring privacy while still availing rich data that can allow the detection of fraud patterns. This feature is important in training machine learning models without compromising privacy.

Machine Learning Methodology

1. **What are the methods used in this project**
   The following machine learning methods were used:
   Logistic Regression: A linear binary classification model that estimates probabilities using the logistic function.

Decision Trees: Non-linear classification model-based procedures that split the data, depending on threshold values coming from the characteristics. End.
Support Vector Machine (SVM): This involves separating classes by a hyperplane to maximally increase the margin between the two classes.
K-Nearest Neighbors: This is a distance-based algorithm that classifies all points by the majority among its neighbors.

2. **Data Usage in this project:**
   The dataset was utilized in the following ways:
   Exploratory Data Analysis: Analysed feature distributions, correlations, and identified class imbalance.
   Feature Scaling: The Time and Amount features were standardized to ensure models performed optimally.
   Class Balancing: The extreme imbalance in the dataset was addressed using SMOTE to oversample the minority class fraudulent transactions.
   Model Training and Testing: The data was split into training and testing sets for model development and evaluation in a 70-30 split.
   Performance Evaluation: Employed Precision, Recall, F1-Score, and Accuracy to evaluate model predictions on the test data.
   Handling Missing Data: There was no missing data in the dataset, so there was no need to rectify the dataset.

3. **ML Frameworks and Libraries Used:**
   The following frameworks and libraries were used in the implementation and evaluation of models:
   Pandas: For manipulation and preprocessing of data; cleaning, analyzing, and transforming data in the dataset.
   NumPy: Numerical operations and handling of data arrays.
   Matplotlib and Seaborn: Visualization, including histograms, box plots, and correlation heatmaps for exploratory data analysis.
   Scikit-learn: Used for the implementation of machine learning models, splitting of data, and model performance evaluation (Logistic Regression, Decision Tree, SVM, KNN).
   Imbalanced-learn: Used only for handling imbalanced data using the SMOTE technique.
   StandardScaler from Scikit-learn: The numerical features will be normalized to ensure efficient convergence of the models.

RESULTS

1. **Describe the results**

   The results for each model are summarized below, showcasing their performance across multiple metrics:

Model	Accuracy	Precision	Recall	F1 Score
SVM (with SMOTE)	0.997460	0.389776	0.824324	0.529284
SVM (without SMOTE)	0.999380	0.899160	0.722973	0.801498
Decision Tree (with SMOTE)	0.997367	0.372093	0.756757	0.498886
Decision Tree (without SMOTE)	0.999251	0.804348	0.750000	0.776224
Logistic Regression (with SMOTE)	0.975972	0.061262	0.898649	0.114705
Logistic Regression (without SMOTE)	0.999146	0.864078	0.601351	0.709163
KNN (with SMOTE)	0.998361	0.516949	0.824324	0.635417
KNN (without SMOTE)	0.999462	0.947368	0.729730	0.824427

Table 1: Performance metrics for CNN, RNN, and Transformer models.

2. Figures and Analysis:

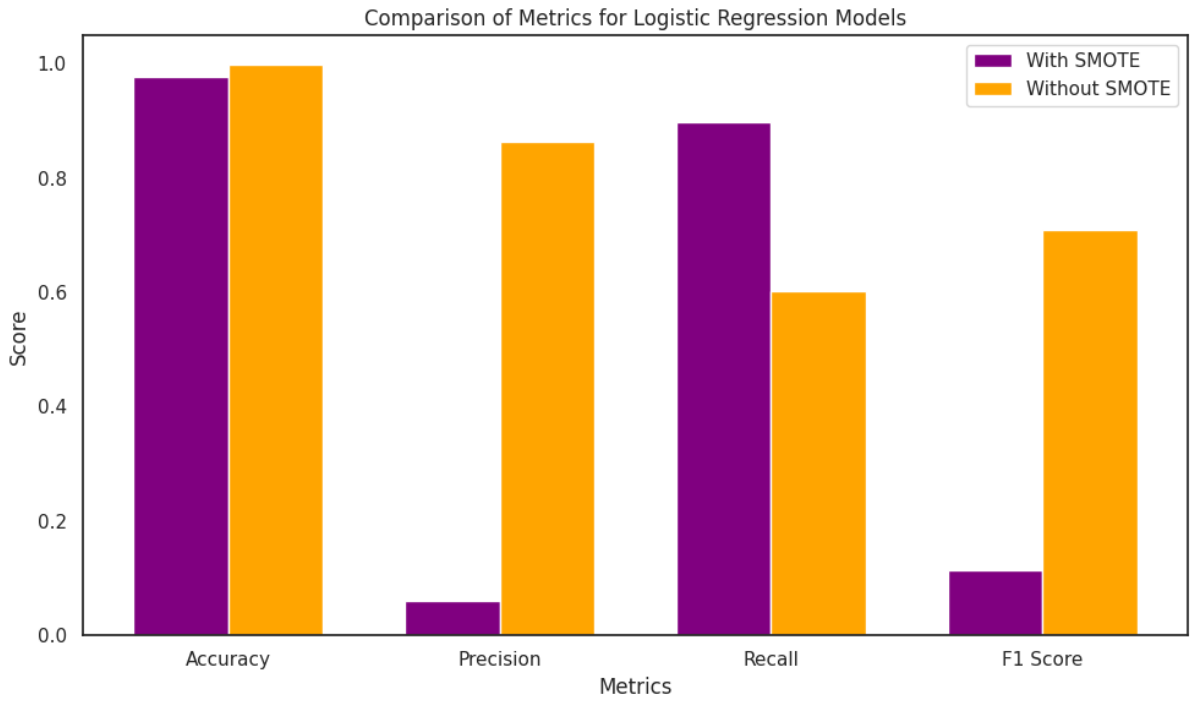

Fig 1: Comparison of metrics for Logistic Regression Models

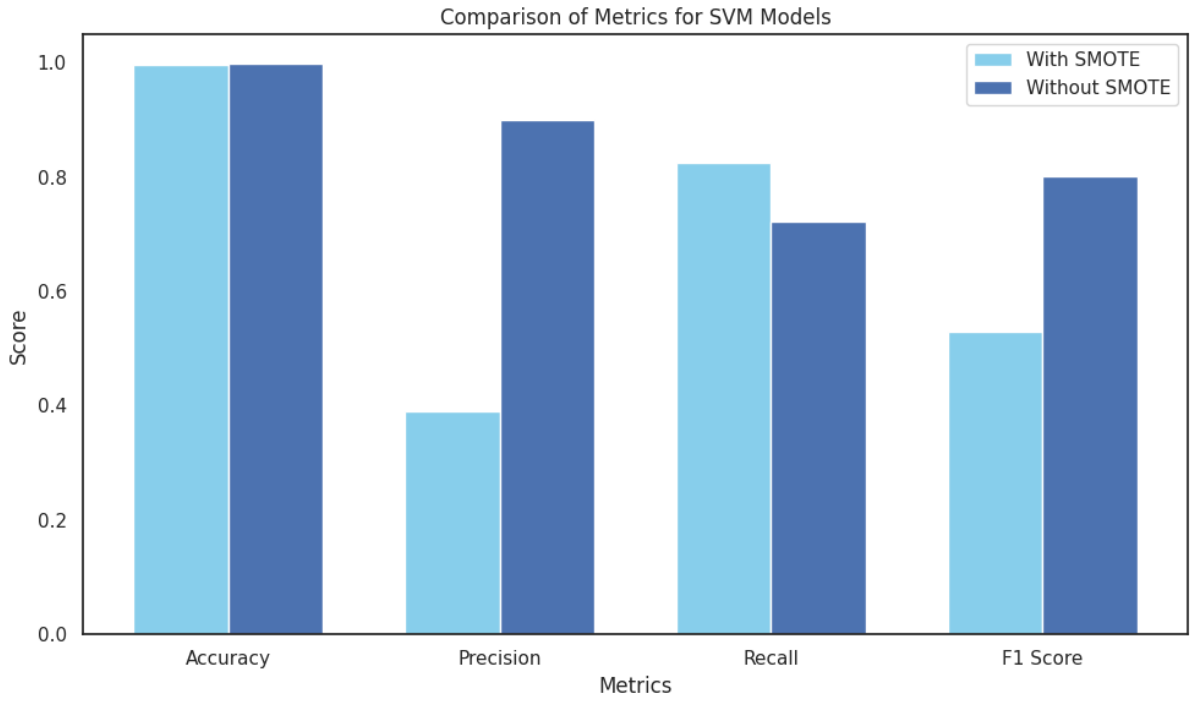

Fig 2: Comparison of metrics for SVM Models

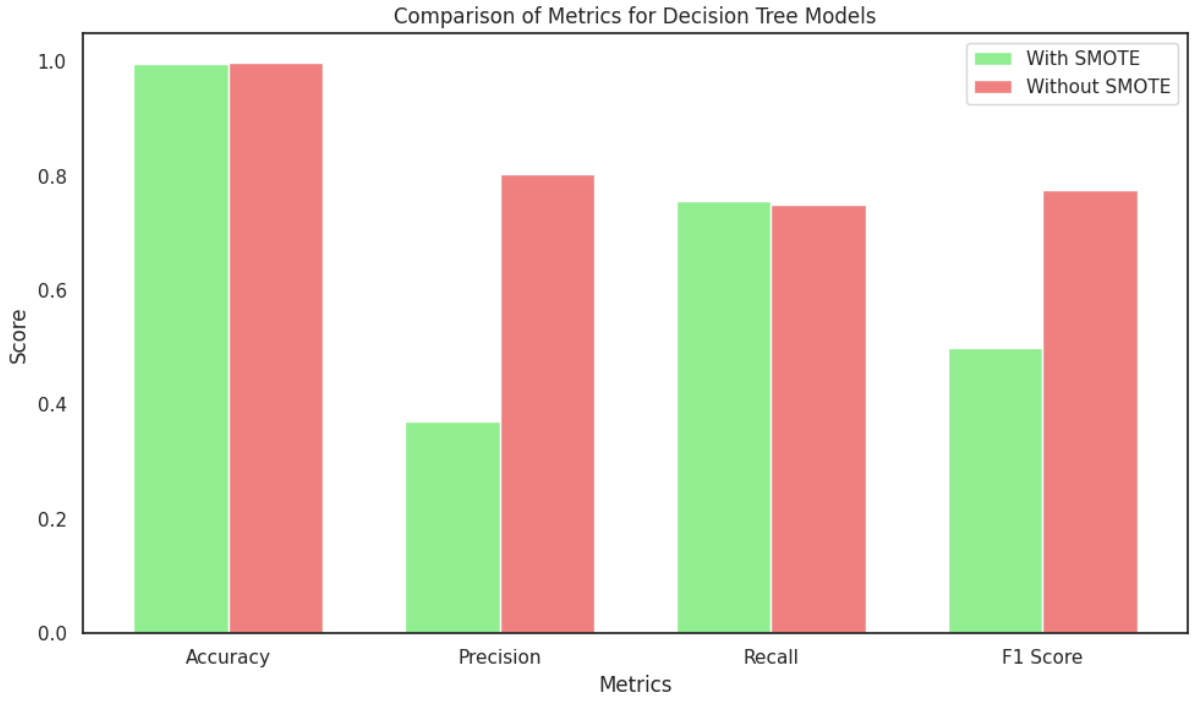

Fig 3: Comparison of metrics for Decision Tree Models

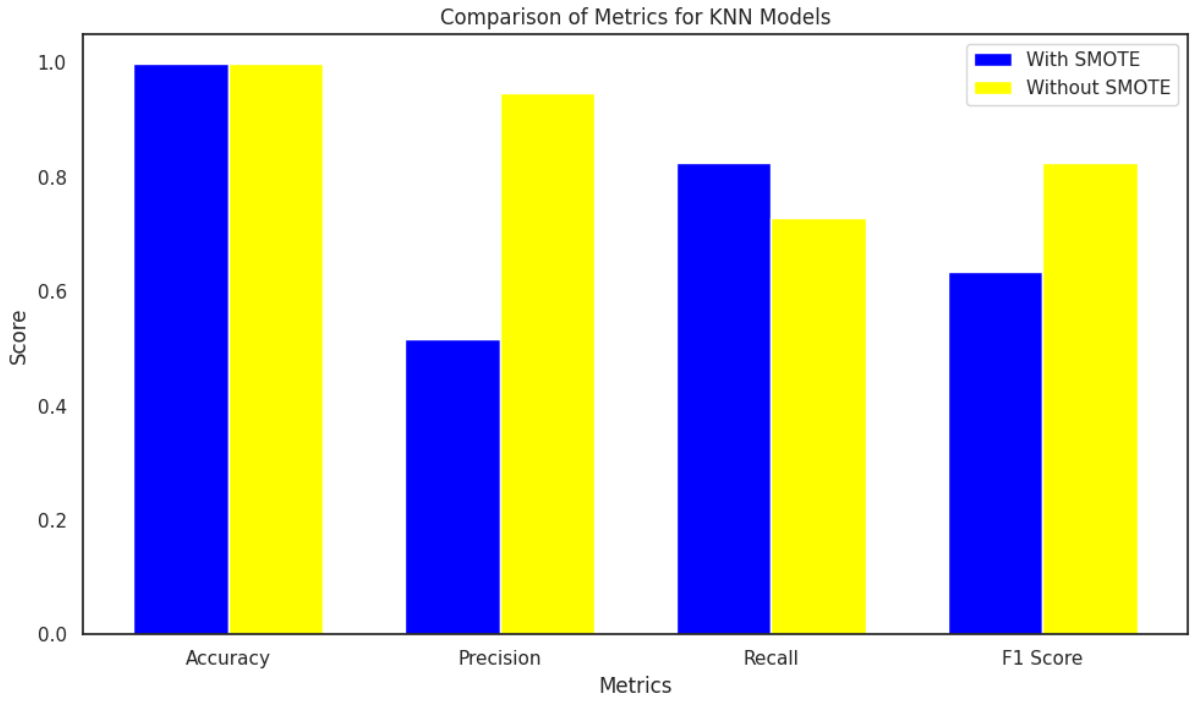

Fig 4: Comparison of metrics for KNN Models

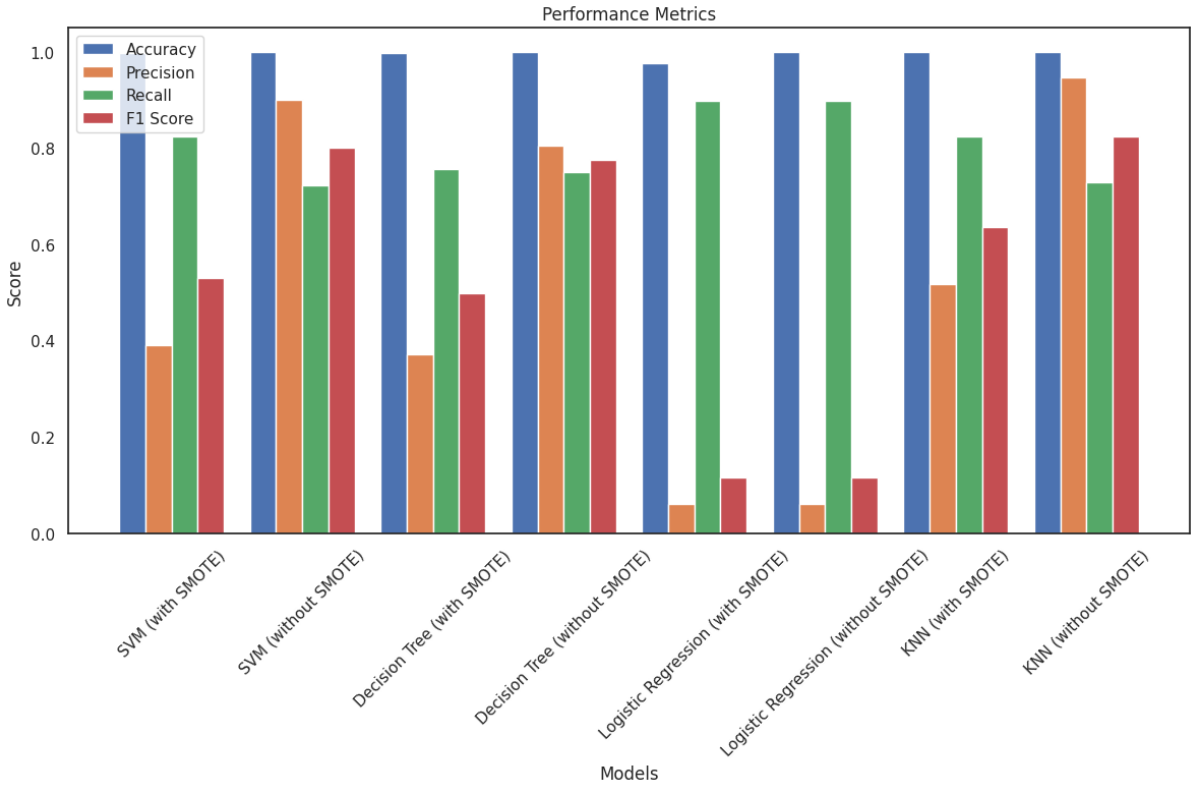

Fig 5: Comparison of metrics for all the models

3. **Comment on the best-performing algorithm**
   Among the evaluated models, KNN (without SMOTE) emerges as the leading choice due to its superior F1 score, indicating an effective balance between precision and recall. This model excels in minimizing false positives while reliably identifying a significant portion of fraudulent transactions. Other models

like SVM (without SMOTE) and Decision Tree (without SMOTE) also demonstrate commendable performance but with slightly lower F1 scores compared to KNN**.** In fraud detection systems, where false positives can cause significant disruptions and false negatives may allow fraudulent activities to go unnoticed, it's crucial to select a model that effectively manages both types of errors. The KNN (without SMOTE) model, with its high precision and robust recall, is the optimal choice based on the data analysis.

4. **Computational Complexity**
   Compared with Decision Tree and Logistic Regression, SVM and KNN are computationally more expensive, due to their longer compilation times.

5. **Are there any trade-offs?**
   Trade-offs Between Precision and Recall:
   Although Logistic Regression had a lower precision, it showed robustness in recall, which is important in fraud detection to avoid false negatives. Its F1-score, however, is very low, which means there is a very low balance between precision and recall.

LESSONS LEARNED

1. **What did you learn?**
   We obtained practical experience in processing real-world imbalanced datasets, performing machine learning pipelines, and analysing model trade-offs. Besides, we have learned how important it is to preprocess and balance data to get the best results from classification models. Imbalanced data preprocessing with methods like SMOTE enhances model performance significantly. In fraud detection, recall (the ability to detect fraudulent transactions) is more important than precision.

2. **What was important in this problem?**
   The critical part of this problem was to minimize the number of false negatives, as missing fraudulent transactions could lead to huge financial losses. Ensuring models were robust in recall, while maintaining reasonable precision, was essential for practical fraud detection systems.

3. **What were the challenges?**
   Severe class imbalance was a challenge to handle and optimize the models for the minority class of fraud transactions without overfitting. Computational complexity with SVM and KNN models involved some trade-offs with runtime efficiency.

4. **What should the reader get out of reading your project?**
   Your project on credit card fraud detection should leave the reader with the following takeaways:
   Research Significance: Fraud detection is one of the most urgent problems in the financial systems, having important implications for security, preventing financial loss, and earning customer trust in digital payment systems. This project seeks to identify effective machine learning techniques for fraud detection in transactions.

Practical Lessons: It highlights the importance of preprocessing and balancing imbalanced datasets to show how techniques like SMOTE can enhance model performance. Furthermore, it illustrates why recall is considered much more important in fraud detection-to minimize false negatives and prevent financial losses.

Actionable Recommendations: The project wraps up with very clear guidance on the selection of machine learning models for fraud detection, favouring solutions that balance efficiency and accuracy while highlighting computational costs and trade-offs.

References

- https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud
- https://journalofbigdata.springeropen.com/articles/10.1186/s40537-022-00573-8
- https://www.sciencedirect.com/science/article/pii/S1877050923002314
- https://www.kaggle.com/code/janiobachmann/credit-fraud-dealing-with-imbalanced-datasets
- https://medium.com/@corymaklin/synthetic-minority-over-sampling-technique-smote-7d419696b88