Credit Card Fraud Detection

Krithanjay Raju, Winnie Zhang, Sanjana Surapaneni, Roland Saavedra, Nevin Gregory

View on GitHub

Introduction/Background

Credit card fraud is a significant issue in the financial sector, causing billions of dollars in annual financial losses. Common fraudulent activities, such as card cloning, identity theft, and unauthorized transactions are common challenges for credit card companies. To combat this, many institutions use machine learning to detect unusual transaction patterns and flag potential fraud.

Literature Review

Past Cases:

- At SRM Institute, assistant professor S.P. Maniraj built a very similar model to what we are aiming to do. His team's model was able to successfully identify new transactions as fraudulent or not with an accuracy of 99.7% [1].
- Safa and Ganga created a model that heavily utilized logistic regression to analyze distorted credit card data. The logistic regression method had a 99.07% accuracy [2].
- Tiwari et al. (2021) conducted a comprehensive review of machine learning techniques for credit card fraud detection, including Hidden Markov Models, Decision Trees, Logistic Regression, Neural Networks, and others. They analyzed the strengths and limitations of each method, offering insights into their effectiveness in identifying fraud [3].

Dataset Description:

We plan to use publicly available datasets like the Kaggle Credit Card Fraud Detection Dataset, which contains real credit card transactions with fraudulent transactions. This dataset contains transactions in the year 2023 by European cardholders. This data is spanned over the whole year, where there are over 550,000 records [4]. Features from V1-V28 are anonymized and class labels are given to represent not fraud (0) and fraud (1), with only 0.17% of transactions labeled as fraud.

Dataset Link

Problem Definition

Problem

The challenge is to detect fraudulent transactions using a supervised method (classification) to identify patterns that indicate fraud. The data itself is highly imbalanced, and transactions that are fraudulent are rare, making standard machine learning models prone to bias towards the majority class.

Motivation

The motivation behind creating a machine learning model to detect credit card fraud stems from the desire to protect consumers and financial institutions from significant financial losses. As online transactions and digital payments increase, so does the risk of fraud. This offers a more adaptive approach, learning from patterns in transactional data to detect suspicious activity in real time, reducing false positives, and enhancing security without compromising user experience.

Methods

Preprocessing Method:

- **Drop Unnecessary Columns:** We decided to remove the ID column as it is an identifier and does not contribute to predictive modeling.
- **Handling Missing Values:** We checked for missing values and removed any rows containing them to ensure data integrity.
- **Feature Scaling:** Although features V1-V28 are already anonymized and likely normalized, the *Amount* column was scaled using Standard Scaler to bring it in line with the other features.
- **Class Imbalance:** We applied SMOTE to balance the dataset because fraud transactions are much rarer than non-fraudulent ones. Balancing the classes ensures the supervised learning models can better detect fraud without being biased toward the majority class.

We split the preprocessed data into training (80%) and validation (20%) sets. This split provides a strong basis for evaluating model performance during supervised learning.

- **SMOTE**

  - Handles imbalance in datasets, especially in binary classification tasks like fraud detection and transactions where they are much rarer.
  - ML models tend to be more biased towards the majority class, which results in poor performance. In this situation, the model might classify most transactions as non-fraudulent, leading to high accuracy but low effectiveness in detection.

Supervised Method:

To solve the problem of credit card fraud detection, we focused on **supervised machine learning** methods. These models require labeled data, making them well-suited for detecting fraud when historical data includes both fraudulent (Class 1) and non-fraudulent transactions (Class 0). The supervised approach allows models to learn the patterns and characteristics associated with fraudulent behavior, enabling accurate predictions on new, unseen data. Three unique algorithms were selected for this task, each offering different strengths in terms of interpretability, computational efficiency, and ability to capture complex relationships in the data.

Algorithms/Models:

1. Logistic Regression

- **Supervised Learning Algorithm:** Logistic Regression is a linear classification model specifically designed for binary outcomes, making it a natural choice for fraud detection (fraudulent vs. non-fraudulent transactions).

- **Why Selected:**
  1. **Binary Classification:** It is particularly effective for problems where the outcome is binary, such as fraud detection.
  2. **Interpretability:** Logistic regression provides coefficients that explain the importance of each feature, enabling financial institutions to understand **which factors contribute most to fraudulent transactions**.
  3. **Efficiency:** It is computationally inexpensive, making it ideal for large datasets.
- **Expected Results:** Logistic Regression is expected to perform well in identifying linear relationships between features and fraud. However, due to its linear nature, it may struggle with non-linear patterns or complex interactions between features. This model provides a solid baseline for comparison with more sophisticated methods.

2. K-Nearest Neighbors (KNN)

- **Supervised Learning Algorithm:** KNN is a non-parametric algorithm that predicts the class of a transaction based on the majority class of its nearest neighbors in the feature space.

- **Why Selected:**
  1. **Non-Linear Patterns:** KNN captures local relationships between data points, making it effective for detecting fraud patterns that are not strictly linear.
  2. **Instance-Based Learning:** It does not require a pre-defined model, allowing it to adapt to the data dynamically.
  3. **Ease of Understanding:** The concept of “neighborhoods” is simple to visualize, which can aid in understanding how predictions are made.
- **Expected Results:** KNN is expected to achieve high accuracy, especially in identifying clusters of similar fraudulent transactions. However, it may overfit the data, particularly in cases of class imbalance or high-dimensional data. The computational cost is also a challenge, as KNN must compute distances for each query point, making it less suitable for large-scale datasets without optimization.

3. Decision Tree Classifier

- **Supervised Learning Algorithm:** Decision Trees create a flowchart-like structure that splits data based on feature thresholds, progressively segmenting it into homogeneous classes.

- **Why Selected:**
  1. **Handling Non-Linear Data:** Decision trees can model complex, non-linear relationships between features and the target variable.
  2. **Interpretable Rules:** The tree structure provides clear decision rules, making the predictions interpretable for stakeholders.
  3. **Feature Importance:** Decision trees inherently identify the most important features during the splitting process, providing insights into the most predictive attributes for fraud detection.
- **Expected Results:** The Decision Tree model is expected to excel at capturing non-linear interactions between features and identifying rules that distinguish fraudulent from non-fraudulent transactions. However, it may overfit the training data, especially with imbalanced datasets, requiring regularization (e.g., limiting tree depth or minimum samples per leaf) to enhance generalization.

Results and Discussion

Quantitative Metrics

Training Score:

- **Logistic Regression:** 96.0% accuracy score
- **K-Nearest Neighbors (KNN):** 100.0% accuracy score
- **Decision Tree Classifier:** 100.0% accuracy score

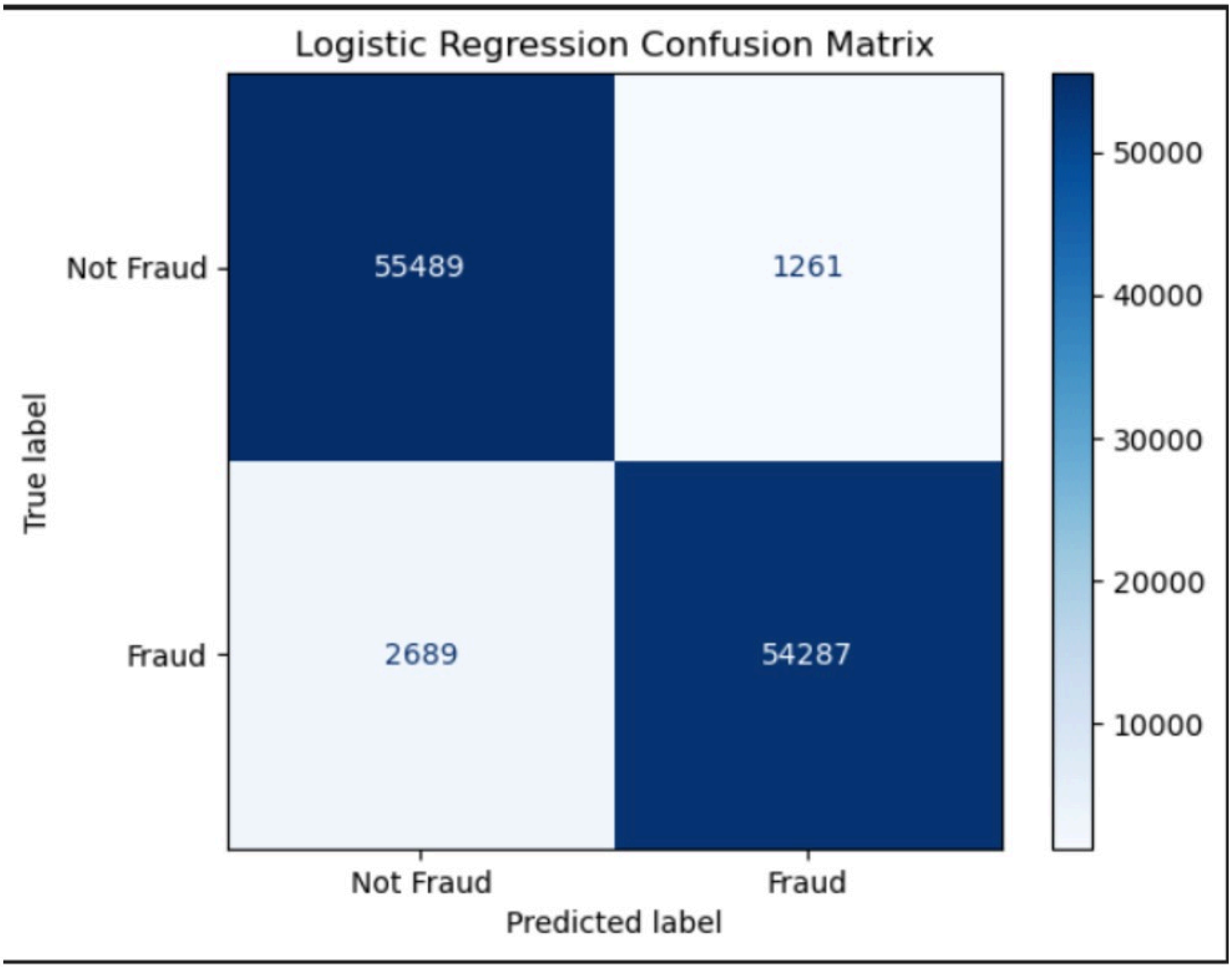
Validation Score:

- **Logistic Regression Cross Validation Score:** 96.49%.
- **K-Nearest Neighbors Cross Validation Score:** 99.9%.
- **Decision Tree Classifier Cross Validation Score:** 94.71%.

Visualizations:

1. Confusion Matrix for Logistic Regression:

Confusion matrices are essential for understanding the performance of binary classifiers like Logistic Regression. This will show how well the model distinguishes between fraudulent and non-fraudulent transactions.



Visualization: Confusion Matrix

What It Tells Us:

- The confusion matrix highlights the model's ability to correctly classify both non-fraudulent (Class 0) and fraudulent (Class 1) transactions.
- The **true positives (54,287) and true negatives (55,489)** show a strong ability to predict both classes accurately.
- There are some **false positives (1,261)** and **false negatives (2,689)**, suggesting room for improvement, especially in catching fraudulent cases.

Strengths:

- Logistic Regression provides interpretable results, and its relatively low false positives ensure fewer disruptions for legitimate transactions.
- It balances performance well across both classes with a **96.49% cross-validation score.**

Limitations:

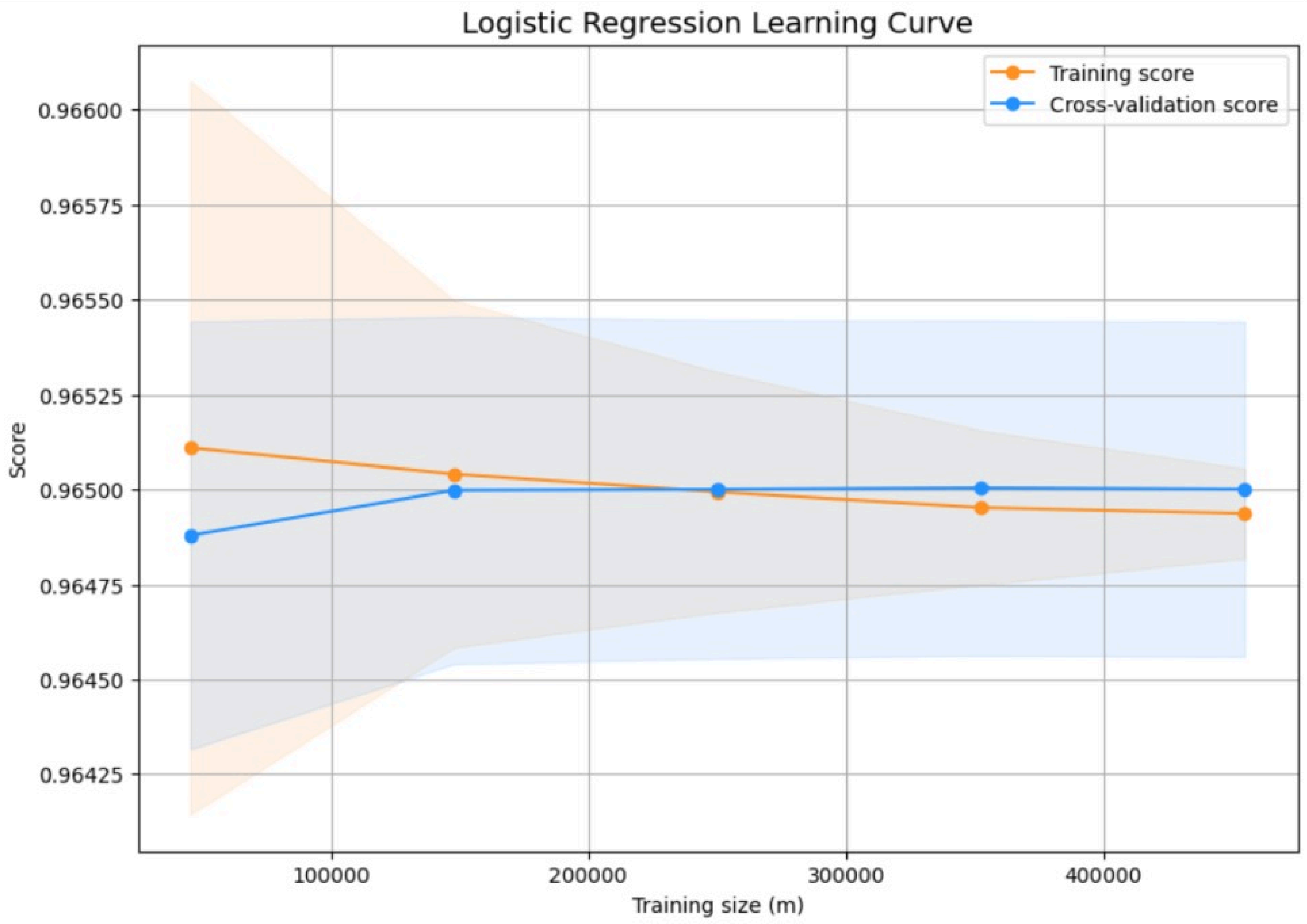
- It struggles to handle non-linear relationships in data, leading to false negatives where fraud patterns may be more complex.

Trade Offs:

- The model is computationally efficient, making it a good baseline, but it sacrifices some accuracy for simplicity.

Comparison:

- Logistic Regression underperforms compared to KNN in terms of cross-validation accuracy and ability to catch more fraudulent cases (as seen in false negatives). However, it outperforms KNN in terms of interpretability and computational efficiency.



Visualization: Logistic Regression Learning Curve

What It Tells Us:

- The learning curve shows the relationship between training size and model performance (both training and cross-validation scores).
- The training score remains stable as more data is added, indicating that the model is not overfitting.
- The cross-validation score closely follows the training score with a small gap, suggesting that the model generalizes well to unseen data.

Strengths:

- Logistic Regression performs consistently across different dataset sizes, as the cross-validation score stabilizes quickly.
- The small gap between training and validation scores indicates a low bias and low variance model, making it robust for real-world deployment.

Limitations:

- The flat curve suggests that adding more data may not significantly improve performance, as the model has likely learned the linear relationships present in the dataset.
- Logistic Regression may fail to capture non-linear interactions, which could slightly limit its fraud detection accuracy.

Trade Offs:

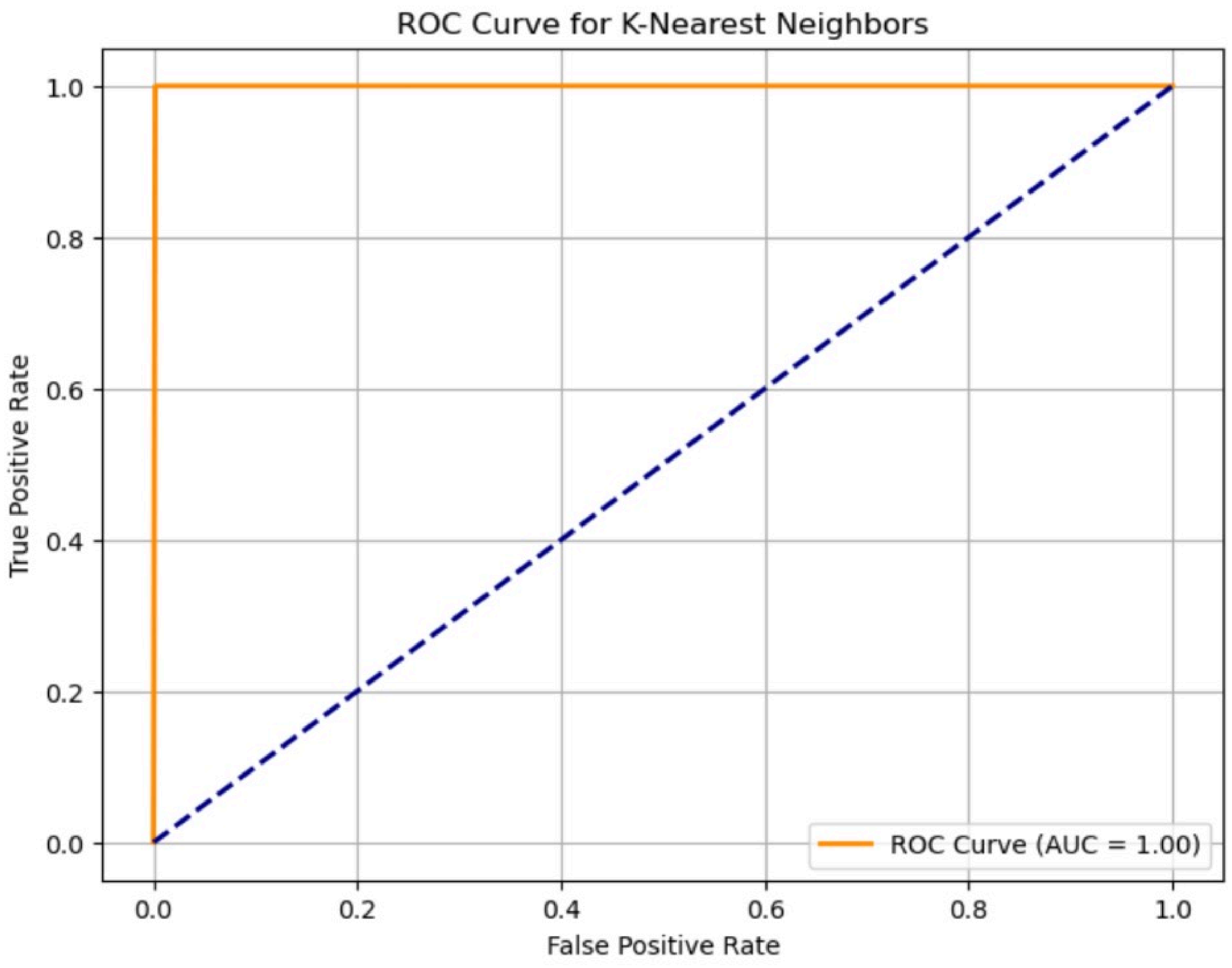
- While Logistic Regression is computationally efficient and performs well with smaller datasets, its linear nature means it might miss complex fraud patterns captured by non-linear models like Decision Trees or KNN.

Comparison:

- Compared to the ROC curve for KNN, Logistic Regression's learning curve emphasizes its ability to generalize better without overfitting.
- Unlike Decision Trees, which heavily rely on certain features, Logistic Regression shows consistent performance as data volume increases, making it more reliable for larger datasets.

2. ROC Curve for K-Nearest Neighbors:

The ROC (Receiver Operating Characteristic) curve evaluates how well the model discriminates between classes across different thresholds. This visualization is particularly helpful for KNN, where distance-based decisions can be visualized in terms of sensitivity and specificity.



Visualization: ROC Curve

What It Tells Us:

- The ROC curve for KNN shows a perfect **AUC = 1.0**, suggesting the model distinguishes between fraudulent and non-fraudulent cases flawlessly across thresholds.
- This result is likely due to overfitting on the training data, where KNN memorizes the dataset instead of generalizing.

Strengths:

- KNN captures non-linear patterns and works well for local clusters of similar data points.
- High sensitivity ensures that fraudulent cases are almost always flagged.

Limitations:

- The model likely overfits due to the high dimensionality and large dataset size, as reflected in the perfect ROC curve and **99.9% cross-validation score**.
- KNN is computationally expensive because it calculates distances for every prediction, making it unsuitable for large datasets like this one.

Trade Offs:

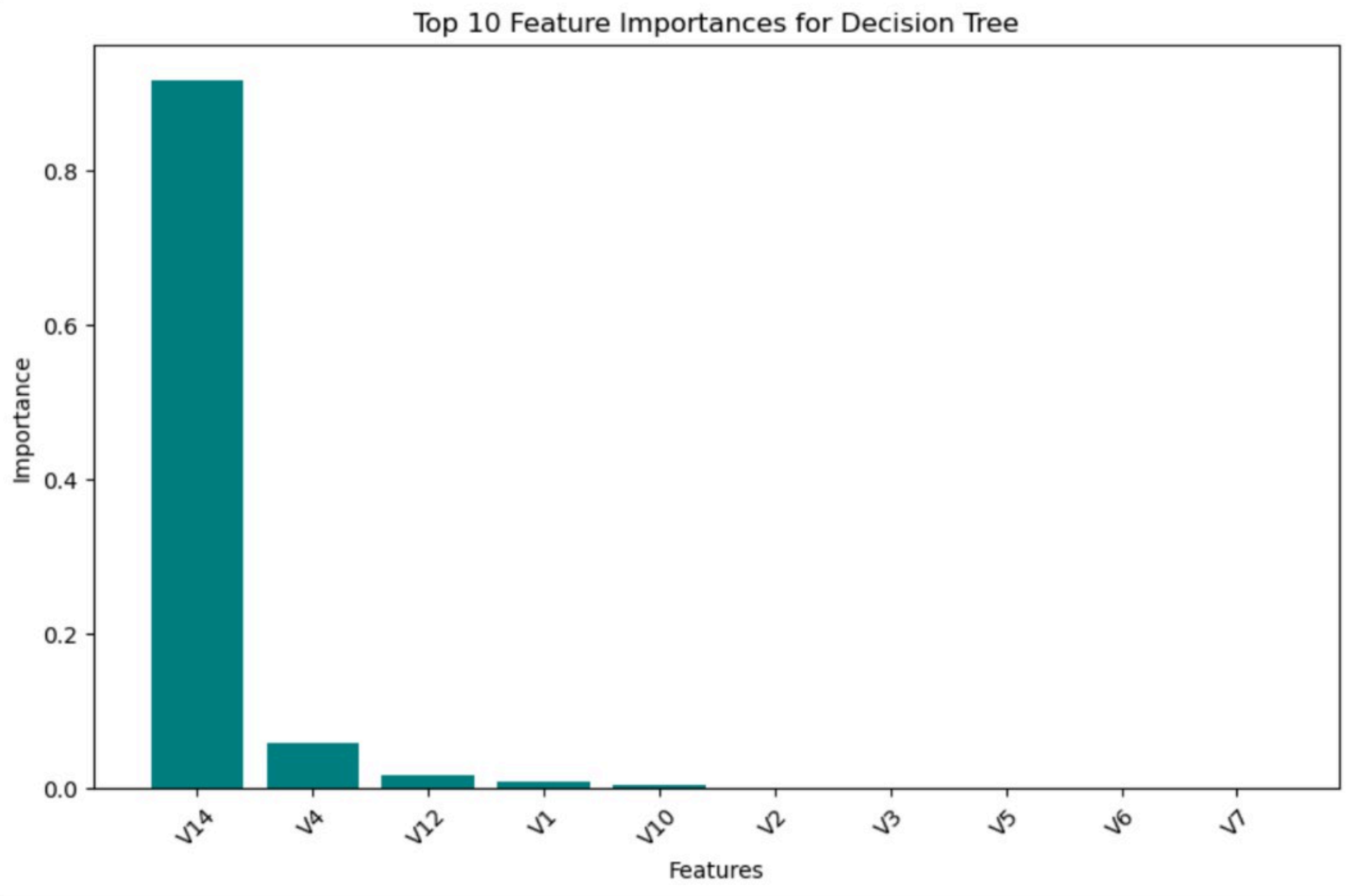
- The model achieves near-perfect accuracy but at the expense of computational efficiency and generalizability.

Comparison:

- While KNN achieves the highest performance on metrics like the ROC AUC, it suffers from overfitting compared to Logistic Regression and Decision Trees. Its computational cost makes it less practical for real-time fraud detection.

3. Feature Importance Barplot for Decision Tree:

Decision Trees provide inherent feature importance scores, which can be visualized to show the most influential features in fraud detection.



Visualization: Feature Importance Bar Plot

What It Tells Us:

- The bar plot reveals that *V14* is the most significant feature for fraud detection, followed by *V4*, *V12*, and *V10*.
- The tree relies heavily on a few key features, indicating that these features strongly influence fraud classification.

Strengths:

- Decision Trees handle non-linear patterns effectively, as seen in their ability to identify highly predictive features.
- The model is interpretable, with clear decision paths and feature importance rankings.

Limitations:

- Decision Trees are prone to overfitting, as shown in the **94.71% cross-validation score**, which is lower than Logistic Regression and KNN.
- Heavily relying on a few features may make the model less robust to variations in the data.

Trade Offs:

- The interpretability of Decision Trees comes at the cost of reduced generalization, as they overfit without proper regularization.

Comparison:

- Decision Trees outperform Logistic Regression in capturing non-linear relationships but are less robust than KNN in overall accuracy. However, they are computationally cheaper than KNN and provide more feature-level insights.

Next Steps:

1. **Address Overfitting in KNN and Decision Tree:**
    - Regularize Decision Trees by limiting *max_depth* or increasing *min_samples_split* to reduce overfitting.
    - For KNN, apply dimensionality reduction (e.g., PCA) to handle high-dimensional data and improve generalization.
2. **Focus on Evaluation Metrics Beyond Accuracy:**
    - Prioritize **recall** to minimize false negatives, as detecting fraudulent transactions is critical.

   - Use metrics like **precision**, **recall**, and **F1-score** for a more comprehensive assessment of model performance.
3. **Implement Ensemble Models:**
   - Combine Logistic Regression, KNN, and Decision Tree in a **Voting Classifier** or **Stacking Ensemble** to leverage their strengths and reduce individual weaknesses, improving overall robustness.

Gantt Chart

Gantt Chart Link

Contributions Table

Name	Contributions
Krithu Raju	Model design and selection, Data cleaning and visualization, Feature Reduction, Implementation & Coding, Model Research, Results & Discussion Writeup, Github Repo
Sanjana Surapaneni	Model selection, Model Research, Data cleaning, Final Report Writeup, Gantt chart
Winnie Zhang	Model selection, Model Research, Github Page, Final Report Writeup, Final Submission
Roland Saavedra	Model design and selection, Data cleaning and visualization, Feature Reduction, Implementation & Coding, Model Research
Nevin Gregory	Implementation & Coding, Model Research, Github Repo
All Members	Video Presentation, Final Report, Results Evaluation and Discussion, Keeping up with the repo

Video Presentation

Video Presentation Link

View Github

Github Repo Link

References:

1. Maniraj, S. P., Saini, A., Ahmed, S., & Sarkar, S. D. (2019). “Credit card fraud detection using machine learning and Data Science.” *International Journal of Engineering Research & Technology*, vol. 8, no. 9, pp. 110-115, 2019. [Online]. Available: https://doi.org/10.17577/ijertv8is090031
2. Safa, S., & Ganga, G., “Credit Card Fraud Detection Using Logistic Regression,” *Master’s Thesis*, Rochester Institute of Technology, Rochester, NY, USA, 2020. [Online]. Available: https://repository.rit.edu/cgi/viewcontent.cgi?article=12455&context=theses
3. Tiwari, P. M., Mehta, S., Sakhuja, N., Kumar, J., & Singh, A. K. (2021). “Credit Card Fraud Detection using Machine Learning: A Study.” arXiv preprint arXiv:2108.10005. [Online]. Available: https://arxiv.org/abs/2108.10005
4. “Credit Card Fraud Detection Dataset 2023.” Kaggle. [Online]. Available: https://www.kaggle.com/datasets/nelgiriyewithana/credit-card-fraud-detection-dataset-2023 (accessed Oct. 24, 2024).
5. “Credit Card Fraud Detection Model.” Kaggle. [Online]. Available: https://www.kaggle.com/models/evanschinonso/credit-card-fraud-detection (accessed Oct. 20, 2024).