KDD Methodology for Fraud Detection

Puneet Bajaj
1st November, 2024

Abstract
This paper applard transaction data and detect fraudulent transactions using machine learning models. The dataset's significanies the Knowledge Discovery in Databases (KDD) methodology to analyze credit ct class imbalance presented a unique challenge, addressed through the application of Synthetic Minority Over-sampling Technique (SMOTE). Three models were explored: Decision Tree, Random Forest, and Logistic Regression, with Logistic Regression showing the highest predictive accuracy. We evaluated model performance using metrics such as precision, recall, F1-score, and ROC-AUC. Visualization tools, including confusion matrices, ROC, and precision-recall curves, further demonstrated Logistic Regression's suitability for fraud detection.

Introduction
Fraud detection remains a high priority in financial systems, where every missed fraudulent transaction poses financial and reputational risks. The primary challenge lies in the high imbalance of transaction data, where fraudulent transactions make up a tiny fraction of the total, necessitating advanced techniques to avoid skewed predictions. Using the KDD methodology, this study aims to uncover patterns within the dataset and identify effective models for accurate fraud detection. The three models applied in this study—Decision Tree, Random Forest, and Logistic Regression—represent different strengths in interpretability and predictive power, with the results demonstrating Logistic Regression's ability to handle imbalanced data effectively.

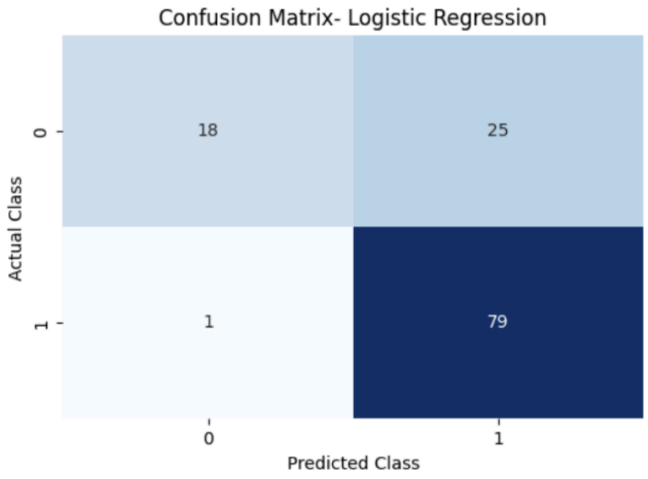
KDD Process

The KDD process comprises five essential phases: Data Selection, Preprocessing, Transformation, Data Mining, and Evaluation. Each phase contributes to systematically analyzing and structuring the data to build a robust model.

1. **Data Selection**: The credit card transaction dataset, sourced from Kaggle, includes 284,807 records with 492 labeled as fraudulent transactions (about 0.17% of the data). Key features are anonymized using Principal Component Analysis (PCA), with 'Amount' representing the transaction's monetary value and 'Time' denoting the time interval between transactions. Selecting this dataset highlights the challenges associated with real-world fraud detection, including high dimensionality, imbalanced classes, and anonymized features.
2. **Data Preprocessing**: Preprocessing involved several steps to ensure data quality and compatibility with machine learning algorithms:

- **Class Balancing**: Given the small percentage of fraudulent transactions, SMOTE was applied to oversample the minority class, creating a balanced dataset that better represents the real-world scenario.
- **Scaling**: To standardize the dataset, the `Amount` and `Time` features were scaled using StandardScaler. This transformation was critical to ensure all features contributed equally in distance-based algorithms and that model convergence was stable during training.

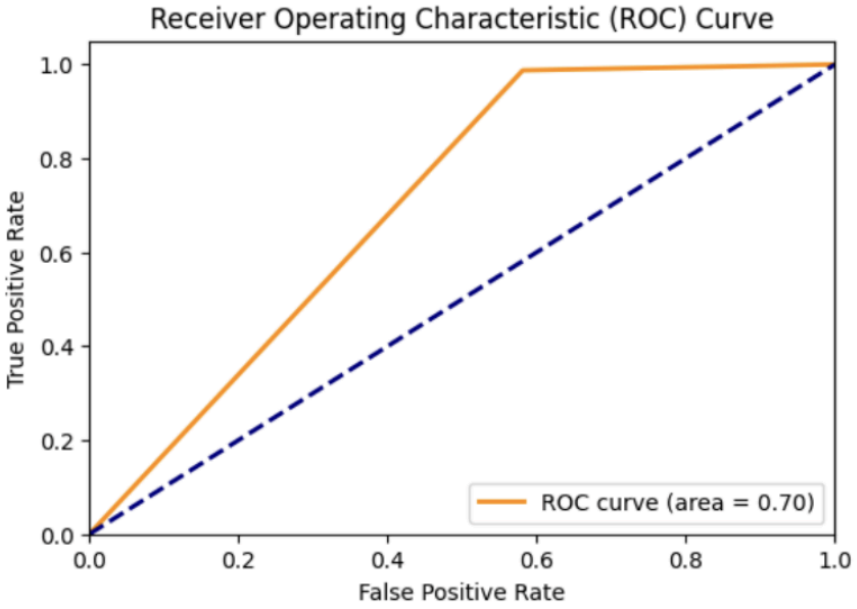
3. By preparing the data in this way, we reduced potential bias introduced by the imbalance and scaled features, setting a strong foundation for model building.
4. **Transformation**: Principal Component Analysis (PCA) was applied to reduce the dimensionality of anonymized features, simplifying the dataset while retaining key predictive components. This step is especially useful in fraud detection, where anonymized features lack intuitive interpretability. PCA transformations not only reduced computational load but also helped avoid overfitting by reducing noise, enabling more accurate and generalizable models.
5. **Data Mining**: We trained three models on the preprocessed and transformed dataset:
   - **Decision Tree**: As an initial model, the Decision Tree provided a baseline accuracy. While interpretable and effective on smaller datasets, it struggled with the highly imbalanced data, frequently misclassifying legitimate transactions as fraudulent.
   - **Random Forest**: This ensemble method combines multiple decision trees to improve accuracy and robustness. Random Forest handled non-linear relationships more effectively than the single decision tree, achieving improved accuracy and reduced overfitting.
   - **Logistic Regression**: As a linear model, Logistic Regression achieved the best performance overall, with high accuracy in fraud detection and strong handling of class imbalance. Its probabilistic approach enabled it to differentiate legitimate from fraudulent transactions with higher precision, particularly in this dataset's structure.
6. **Evaluation**: The effectiveness of each model was evaluated using several metrics:
   - **Confusion Matrix**: The confusion matrix illustrated each model's accuracy in fraud vs. non-fraud classifications, highlighting Logistic Regression's strength in accurately predicting true positives while minimizing false positives.



Confusion Matrix for Logistic Regression

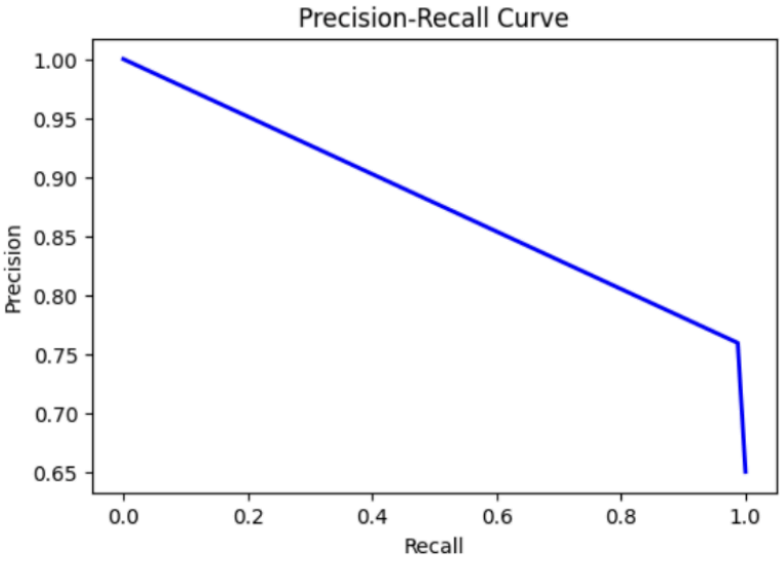
   - **ROC-AUC**: The Receiver Operating Characteristic (ROC) curve provided a graphical representation of each model's trade-off between sensitivity (true

positive rate) and specificity (false positive rate). Logistic Regression achieved the highest area under the curve (AUC), indicating its strong ability to distinguish between classes.



ROC Curve for Logistic Regression

- **Precision-Recall Curve**: Given the imbalanced nature of the dataset, the precision-recall curve offered insights into the trade-offs between maintaining high precision while correctly identifying fraud cases. Logistic Regression showed the best balance, maintaining high precision with minimal sacrifice in recall, which is particularly useful in real-world applications where false positives carry a high cost.



Precision-Recall curve for Logistic Regression

Conclusion

Through the KDD methodology, we identified Logistic Regression as the most effective model for fraud detection within this dataset's constraints. The methodology provided a structured, phase-by-phase approach to handling high-dimensional, imbalanced datasets in a way that maximized predictive power and minimized the risks associated with high-stakes false positives. Future research could focus on deploying this model in real-time environments and exploring ensemble methods or deep learning models, such as autoencoders or recurrent neural networks, to enhance predictive performance. This research underscores the KDD framework's efficacy in guiding complex, high-impact data mining projects in finance and fraud detection.