Credit Card Transactions Fraud Detection Dataset(project-2 report)

Abstract:
The Credit Card Transactions Fraud Detection Dataset is an extensive compilation strategically crafted to support research and innovation in the critical field of credit card fraud detection. It has been sourced from Kaggle. This dataset offers a realistic portrayal of credit card transactions, encompassing both legitimate and fraudulent activities. Featuring anonymized transactional details, timestamps, and merchant information, the dataset stands as a valuable asset for exploring and implementing advanced machine learning and statistical techniques with the goal of improving the precision and efficiency of fraud detection systems. This resource empowers researchers and practitioners to utilize the dataset for the development, refinement, and benchmarking of fraud detection models, thereby contributing to the continual enhancement of security measures in financial transactions.

Introduction:

The enduring challenge of credit card fraud in the financial industry underscores the ongoing need for innovative fraud detection methodologies. Presented as a foundational asset, the "Credit Card Transactions Fraud Detection Dataset" is introduced to advance the capabilities of fraud detection systems. In an era of escalating digital transactions, the demand for robust and adaptive fraud detection mechanisms has never been more imperative. This dataset encapsulates a diverse array of credit card transactions, capturing intricacies and patterns associated with both legitimate and fraudulent activities. The anonymized nature of the dataset prioritizes privacy and confidentiality while providing a robust data source for analytical purposes. The introduction of this dataset seeks to catalyze research and development initiatives in credit card fraud detection. By offering a realistic and representative collection of transactional data, researchers can tackle the challenges posed by sophisticated fraud techniques, ultimately contributing to the development of more resilient and effective fraud detection models. This resource aims to encourage collaboration among researchers, data scientists, and industry experts, collectively striving to reinforce the security of credit card transactions in an evolving digital financial landscape.

Dataset summary:

The dataset outlines a simulated credit card transaction dataset created by Sparkov Data Generation | Github tool, spanning from January 1, 2019, to December 31, 2020. This dataset encompasses both legitimate and fraudulent transactions and involves credit cards from 1000 customers engaging with 800 merchants. The data generation tool, attributed to Brandon Harris, was employed to simulate the transactions, and the resulting dataset has been converted into a standard format, comprising two csv files: a training set (fraudTrain.csv) and a test set (fraudTest.csv).
Here's a detailed summary of the dataset-

Dataset Characteristics:
Duration- Transactions cover the period from January 1, 2019, to December 31, 2020. –
Customers- 1000 customers' credit card transactions are included.
Merchants- Interactions with a pool of 800 merchants are simulated.

Dataset Files- Two files are present-one for training and one for testing.

Columns- In both trainset and test set there are 23 columns. The columns are Unnamed, trans_date_trans_time, cc_num, merchant_category, amt, first, last, gender, street , city, state, zip, lat, long , city_pop , job , dob , trans_num , unix_time , merch_lat, merch_long, is_fraud. These features likely serve as input variables, with "is_fraud" serving as the target variable indicating the presence of fraudulent transactions.

Data Types- In both datasets there are Float, Integer and Object types of data are present.

Fraud Examples in Train dataset- Train Set Size includes 1,296,675 rows and number of Fraud Examples are 7,506

This dataset includes diverse information such as transaction details, customer information, merchant details, and temporal aspects. It provides a valuable resource for developing and evaluating fraud detection models, particularly given the presence of labeled examples in the training set for supervised learning tasks.

Dataset collection:

The Simulated Credit Card Transactions dataset, generated using the Sparkov Data Generation tool, constitutes a synthetic depiction of credit card transactions accessible on the Kaggle platform. Recognized as the dataset's contributor, Kartik Shenoy publicly released it on August 5, 2020. The dataset, available at the URL https://www.kaggle.com/kartik2112/fraud-detection, adheres to the Creative Commons Zero v1.0 Universal license (https://creativecommons.org/publicdomain/zero/1.0/).
This artificially constructed dataset encompasses a range of variables, including transaction date, credit card number, merchant details, transaction category, transaction amount, individual names, street addresses, and gender information. Importantly, all these variables are synthetically generated using the Sparkov tool, providing a controlled environment for analytical exploration. Specifically designed to simulate "Card Not Present Transaction Fraud," a distinct category of fraudulent activities in credit card transactions, Kartik Shenoy, the dataset provider, underscores the intentionally synthetic nature of the data. The dataset, sourced from Kaggle with modifications from a pre-existing version, captures transactions involving 1000 customers interacting with a pool of 800 merchants over a six-month period. Both the training and test segments were directly acquired from the original data, with the test segment being subject to random down sampling. In collecting this dataset, the Kaggle platform served as the primary source, reflecting the collaborative nature of data sharing and exploration within the Kaggle community. This dataset, offering a comprehensive yet simulated setting, stands as a valuable resource for researchers and practitioners engaged in the development and exploration of fraud detection models within the credit card transaction domain.

Used Algorithms:
For training the credit card fraud detection model I have utilized a blend of machine learning algorithms, namely Decision Tree, Random Forest. These algorithms, commonly employed in classification tasks, are adept at discerning and classifying instances of fraudulent credit card transactions.

Decision Tree: Decision trees adopt a tree-like structure where each node makes decisions based on specific features. The algorithm iteratively divides the dataset based on the feature that optimally separates classes. Decision trees are interpretable, capable of capturing intricate relationships within the data.

Random Forest: Random Forest, an ensemble learning method, constructs multiple decision trees and consolidates their predictions. Introducing randomness during the tree-building process enhances generalization and mitigates overfitting. The robustness of Random Forest often yields higher accuracy compared to individual decision trees.

I also implemented K-Nearest Neighbors (KNN) and Logistic Regression model initially but did not get expected result so did not choose this two algorithms.

The approach, combining the strength of Random Forest, and the interpretability of Decision Trees, forms a robust framework for credit card fraud detection. This framework comprehensively captures diverse aspects of the data, contributing to enhanced overall model performance.

Experimental Setup:
To set up the experiment for detecting credit card fraud, I followed a step-by-step process to make sure our model is reliable and works well. First, I bring in the necessary data into our chosen environment or programming language. This step is like laying the groundwork for my analysis and makes it easy to explore the data later on.

Importing Data- The initial step in the experimental setup involves importing the relevant data into the chosen environment or programming language. This ensures that the dataset is accessible for subsequent analysis and manipulation.
Exploring Data: Data exploration entails a comprehensive examination of the dataset to gain insights into its structure, characteristics, and potential patterns. Descriptive statistics, visualizations, and summary metrics are employed to understand the distribution and relationships within the data.

Exploratory Analysis of Data- Building on data exploration, exploratory analysis involves a deeper investigation into specific aspects of the dataset. This includes identifying correlations between variables, detecting outliers, and gaining a nuanced understanding of the data's underlying patterns.

Cleaning Data- Data cleaning is a critical step where inconsistencies, missing values, and errors are addressed. This process ensures that the dataset is reliable and suitable for subsequent analysis, preventing potential biases or inaccuracies in the results.

Data Preprocessing- Data preprocessing involves several below key sub-steps

Feature Engineering- Feature engineering aims to create new, informative features from the existing ones or transform variables to enhance the model's performance. Here I have created few like transaction month, transaction year, age etc.

Split Train and Test Data- The dataset was already partitioned into training and testing sets. I set the target class. The training set is used to train the model, while the testing set evaluates its performance on unseen data.

Handle Data Imbalance- If there is a significant imbalance in the distribution of classes (e.g., fraudulent and non-fraudulent transactions), techniques such as oversampling, undersampling, or the use of specialized algorithms are applied to address this imbalance.

Standardize Data- Standardization involves scaling numerical features to a common mean and standard deviation. Here I have used StandardScaler. This ensures that different features are on a similar scale, preventing any particular feature from dominating the model training process.

Model training-In this phase of model training, machine learning models are selected and trained based on the preprocessed data. For credit card fraud detection, the chosen models include Decision Tree, Random Forest. This stage involves optimizing performance metrics, and evaluating the model's generalization on new, unseen data. This comprehensive experimental setup aims to develop a robust and effective credit card fraud detection model.

Results:

Decision Tree Results and evaluation

The Decision Tree model exhibits a high accuracy of 97%, primarily driven by the accurate classification of non-fraudulent transactions (class 0). However, the model struggles significantly in identifying fraudulent transactions (class 1), as indicated by the low recall of 47%. This means that a substantial number of actual fraud cases are being misclassified as non-fraudulent. The precision for the fraudulent class is only 6%, suggesting a high number of false positives. This implies that many transactions predicted as fraudulent are, in fact, legitimate. The result is showing below-

	precision	recall	f1-score	support
0	0.00	0.00	0.00	553574
1	0.00	1.00	0.01	2145
accuracy			0.00	555719
macro avg	0.00	0.50	0.00	555719
weighted avg	0.00	0.00	0.00	555719

Random Forest Results and evaluation

The Random Forest model achieves perfect accuracy on the training set, emphasizing its ability to accurately classify both non-fraudulent and fraudulent transactions. Similar to the Decision Tree, the Random Forest faces challenges in correctly identifying fraudulent transactions, reflected in the recall of 45%. The precision for the fraudulent class is also 45%. While the model's overall accuracy is high, the balanced accuracy metrics (precision and recall) for the minority class suggest limitations in fraud detection performance. The result is showing below-

	precision	recall	f1-score	support
0	1.00	0.97	0.98	553574
1	0.06	0.47	0.10	2145
accuracy			0.97	555719
macro avg	0.53	0.72	0.54	555719
weighted avg	0.99	0.97	0.98	555719

Discussion

Imbalance Impact- The class imbalance in the dataset heavily influences model performance. Both models achieve high accuracy primarily due to the dominance of non-fraudulent transactions. However, this mask the challenges in accurately identifying fraudulent cases.

Trade-off Between Precision and Recall- The low precision and recall for the fraudulent class indicate a trade-off between minimizing false positives and false negatives. Striking the right balance is crucial in fraud detection, as false positives may inconvenience customers, while false negatives may result in financial losses.

Random Forest Overfitting- The perfect accuracy of the Random Forest on the training set raises concerns about overfitting. The model might be too tailored to the training data, potentially impacting its generalization to new, unseen data.

Improvement Opportunities- Further exploration is needed to enhance the models' ability to detect fraud, possibly through hyperparameter tuning, feature engineering, or alternative algorithms better suited to imbalanced datasets.

Consideration of Business Impact- Evaluating the business impact of misclassifications is essential. Determining the cost of false positives and false negatives can guide model optimization to align with business objectives.

Discussion of the strengths and weaknesses of my approach:

In the context of the "Credit Card Transactions Fraud Detection Dataset," the chosen approaches and algorithms bring distinct strengths and weaknesses to the table.

The Decision Tree algorithm, known for its interpretability, offers a clear understanding of the decision-making process, which is crucial in the intricate domain of credit card fraud detection. However, its susceptibility to overfitting may be a concern given the dataset's potential noise or outliers.

The Random Forest algorithm, leveraging ensemble learning, addresses overfitting concerns and achieves high accuracy, making it well-suited for identifying fraudulent transactions. However, its computational complexity could be challenging, necessitating careful consideration within the resource constraints outlined in the experimental setup.

In consideration of the dataset's characteristics, Random Forest's robustness to outliers and ability to handle imbalances, combined with Decision Trees' interpretability, seem well-suited for the experimental setup's objectives. The computational considerations, such as the potential complexity of Random Forest, should be balanced against the need for efficient processing. The choice of algorithm ultimately hinges on a nuanced understanding of the dataset, the experimental context, and the specific priorities in credit card fraud detection.

Conclusion

While both models demonstrate high accuracy, their limitations in detecting fraudulent transactions, especially in an imbalanced dataset, necessitate further refinement. Addressing these challenges requires a thoughtful approach, considering the business context and trade-offs between precision and recall in credit card fraud detection.