IML Course Project: Credit Card Fraud Detection

Submitted to: Dr. Munnendra Ojha

Name:Payal Meena

Roll No. IIT2021060

Problem Definition

The project's primary goal is to build a machine learning-based system for detecting credit card fraud. This system will analyze diverse features and patterns related to credit card transactions, including transaction amounts, locations, and frequency, to distinguish between legitimate and fraudulent activities. The importance of this project lies in its potential to strengthen financial security, safeguarding users from unauthorized transactions and potential financial losses. By developing an effective and precise detection system, the project aims to enhance the overall security of credit card transactions and ensure a more secure financial experience for users.

Literature Review

Purposed an unsupervised fraud detection method using autoencoder based clustering.The autoencoder is anauto associatorneural network they have used it to lower the dimensionality, extract the useful features, and increase the efficiency of learning in a neural network. They had used European dataset with 284807 transactions in which 0.17% is the fraud and trained there autoencoder based clustering with the following parameters

Number of iterations = 300

Number of clusters = 2

Clustering initialization = k-means++ Divergence tolerance = 0.001

Learning rate of the model = 0.1

Number of epochs = 200

Activation function = elu, Relu

As a result, they got their training loss as 0.024 and validation loss as 0.027 and the mean of not fraud data 75% less than the mean of reconstructive error that is 25% the design of there is model is context-free. In concern about the model predictions, the True positive are 56,257, Falsenegative is 607, False positive are 18, True negativesis 80 and the best preferred are (56,257 + 80 = 56,337). The right predictions made are 56,337 out of 284807.

CART-based random forest. They use different random forest algorithms to train the behavior features of normal and abnormal transactions and both of the algorithms are different in their base classifications and their performance. They applied both of the algorithms on the dataset e-commerce company in China. In which the fraud transaction in the subsets ratio is 1:1 to 10:1. As a result, accuracy from the random-treebased random forest is 91.96% whereas in CART-based random forest is 96.7%. Since the data used is from the B2C dataset many problems arrived such as unbalanced data. Hence, thealgorithm can be improved.

Data preprocessing

In the context of machine learning, the effectiveness of our model is highly dependent on the quality of the dataset we use. Prior to delving into the nuances of model development, a pivotal phase involves the preprocessing of your dataset. This encompasses addressing missing values, handling duplicates, and transforming variables into a format that is conducive for model comprehension. This chapter delves into the intricate steps of data preprocessing undertaken for a credit card fraud detection project.

Removal of Unnecessary Columns: In dataset, columns like 'Unnamed: 0' and those with a single unique value may not provide meaningful insights into fraudulent transaction. Conduct a thorough analysis of each column to identify those that don't contribute significantly to the fraud detection task. Remove columns with identifiers like 'Unnamed: 0' and those with a single unique value, as they are unlikely to aid in distinguishing fraudulent from non-fraudulent transactions.

Handling Missing Values: Missing values might occur due to various reasons such as errors in data collection or processing. Identify columns with missing values and decide on an appropriate strategy. Common techniques include imputation, where missing values are replaced with a calculated value (mean, median, or mode), or removal of rows or columns with missing values.

Duplicate Removal: Duplicates in the dataset can distort the model's understanding and lead to biased results.Check for duplicate entries based on unique identifiers like transaction IDs or combinations of features. Remove duplicates to ensure that each transaction is represented only once.

Scaling and Distributing: In this phase of our kernel, we will first scale the columns comprise of **Time** and **Amount** . Time and amount should be scaled as the other columns. On the other hand, we need to also create a sub sample of the dataframe in order to have an equal amount of Fraud and Non-Fraud cases, helping our algorithms better understand patterns that determines whether a transaction is a fraud or not.

Splitting Data: Before proceeding with the Random UnderSampling technique we have to separate the orginal dataframe. Why? for testing purposes, remember although we are splitting the data when implementing Random UnderSampling or OverSampling techniques, we want to test our models on the original testing set not on the testing set created by either of these techniques. The main goal is to fit the model either with the dataframes that were undersample and oversample (in order for our models to detect the patterns), and test it on the original testing set.

Transformation: Credit card fraud detection often involves numerical features like transaction amount and time.Standardize or normalize numerical features to bring them to a similar scale. Standardization involves transforming the data to have a mean of 0 and a standard deviation of 1, while normalization scales the values to a range of 0 to 1. This ensures that all features contribute equally to the model, preventing certain features from dominating due to their scale. Additionally, transformation techniques like logarithmic scaling might be applied to skewed features for a more symmetric distribution.

Conclusion: By implementing these preprocessing techniques tailored to the specific characteristics of credit card transactions, we establish the groundwork for the model to discern intricate patterns associated with fraudulent activities. The optimized dataset not only ensures the model's accuracy but also contributes to bolstering the security measures against credit card fraud. As we move forward in model development, the comprehensive data preprocessing undertaken serves as a crucial enabler for creating a robust and efficient credit card fraud detection system, enhancing the overall security of financial transactions.

Feature Selection

Feature selection is a crucial phase in the machine learning , where the goal is to identify and retain the most important features for model training while discarding those that may introduce noise or redundancy. The process involves a comprehensive analysis of both numerical and categorical features.

In my code, explicit feature selection techniques are not applied. Instead, the focus is on outlier detection and removal, correlation analysis, and subsampling to address class imbalance. Let's discuss how feature-related aspects are handled in the code.

Correlation Analysis:

- Correlation matrices (`corr` and `sub_sample_corr`) are used to visualize the correlation between features and the target variable ('Class').
- Features with high positive or negative correlations with the target variable are identified.
- Boxplots are created to show the distribution of features with the highest negative and positive correlations.

Outlier Detection:

- Outlier removal is performed for features V14, V12, and V10 using the interquartile range (IQR) method.
- The reduction in extreme outliers is visualized using boxplots.

Subsampling:

- Random under-sampling is applied to balance the class distribution, ensuring an equal number of fraud and non-fraud instances.
- This subsample is then used for further analysis and model training.

While the code focuses on addressing imbalance and outliers, traditional feature selection techniques such as Recursive Feature Elimination (RFE) or feature importance from tree-based models are not explicitly employed.

ML Model Consideration

Choosing the right machine learning models is a crucial decision in establishing an effective Credit Card Fraud Detection system. In this chapter, we explore the rationale behind our model selection, emphasizing the unique strengths of each chosen algorithm and their collective contribution to the overall success of the project.

1. Model Selection:

Our model ensemble comprises two powerful classifiers: RandomForestClassifier, LogisticRegressionClassifier. Each of these models offers distinct advantages that synergize to enhance the robustness, predictive accuracy, and efficiency of our fraud detection system.

Random Forest: Random Forest, an ensemble learning method, is selected for its ability to handle complex, non-linear relationships within the data. Given the intricate patterns associated with credit card fraud, Random Forest's capacity to mitigate overfitting and provide high accuracy is invaluable.

Logistic Regression: Logistic Regression serves as an initial model due to its simplicity and interpretability. It is well-suited for scenarios where the relationship between features and the binary outcome (fraud or non-fraud) can be adequately captured using a linear decision boundary.

2.Feature Engineering:

In crafting an effective credit card fraud detection model, the feature set is meticulously curated to encompass both numeric and potentially relevant categorical attributes. Each feature's selection is grounded in its significance in aiding the model to discern patterns indicative of fraudulent credit card transactions.

3.Preprocessing:

The preparation of data for model training involves careful preprocessing steps tailored to the credit card fraud detection context. Numeric features are standardized using techniques such as StandardScaler, MinMaxScaler, and RobustScaler. This ensures a consistent scale across the dataset, facilitating the model's ability to capture patterns irrespective of the original scale. Furthermore, categorical features are transformed using OneHotEncoder to represent them in a format conducive to model training.

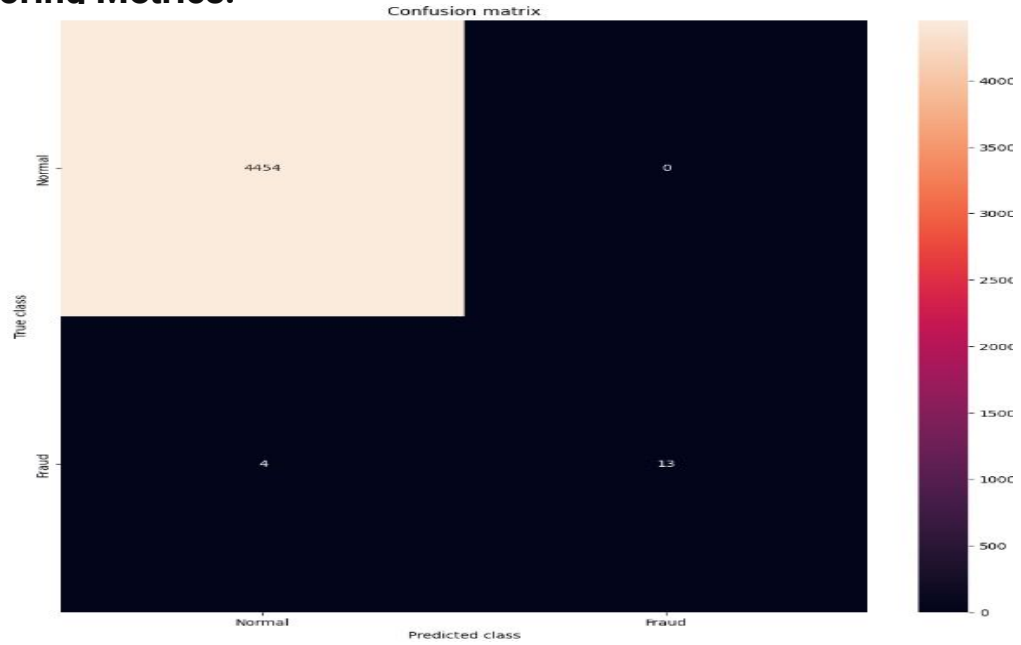
4. Handling Imbalanced Data:

Dealing with imbalanced datasets, a common challenge in credit card fraud detection, requires specialized techniques. To address this issue, Synthetic Minority Over-sampling Technique (SMOTE) is integrated into the preprocessing pipeline. By generating synthetic instances of the minority class (fraudulent transactions), SMOTE effectively mitigates imbalances in the dataset. This proactive step prevents the model from developing a bias toward the majority class (non-fraudulent transactions), enhancing its ability to accurately identify and classify instances of credit card fraud.

5.Model Evaluation and Optimization:

- To assess the performance of our credit card fraud detection model, we employ Stratified 5-fold Cross-Validation. This method is chosen for its ability to maintain the distribution of the target variable, ensuring that each fold of the cross-validation process represents a balanced proportion of fraudulent and non-fraudulent transactions. This approach enhances the reliability of the model's generalization performance.

6.Confusion Scoring Metrics:



Model creation

With a comprehensive understanding of the selected models and the features crucial for our credit card fraud detection system, this chapter delves into the creation of the machine learning models.

1.Logistic Regression Classifier:

Initialization:

- In the provided code, the LogisticRegression model is instantiated with default parameters using `LogisticRegression()` .

Parameters:

Logistic Regression has several important parameters:

- penalty: This parameter determines the regularization type, and the code considers both 'l1' and 'l2' penalties.
- C: The inverse of regularization strength. Smaller values specify stronger regularization.

2.Random Forest Classifier:

Initialization:

- The RandomForestClassifier is instantiated with default parameters using RandomForestClassifier(random_state=42).
- This creates an ensemble of decision trees for the purpose of credit card fraud detection. Each tree is constructed from a random subset of features and a random subset of the training data.
- The random_state=42 ensures reproducibility, allowing consistent results across different runs.

Parameters:

- n_estimators: The number of trees in the forest.
- max_depth: The maximum depth of the trees.
- min_samples_split: The minimum number of samples required to split an internal node.
- min_samples_leaf: The minimum number of samples required to be at a leaf node.
- max_features: The number of features to consider when looking for the best split.

3. Data Splitting:

- Prior to model training, the dataset for credit card fraud detection is split into training and testing sets. The `train_test_split` function is utilized, allocating 80% of the data for training and reserving 20% for testing.
- To ensure a balanced representation of both normal and fraudulent transactions in both sets, the `stratify` parameter is set, preserving the distribution of classes and maintaining the integrity of the dataset. This is crucial for accurate evaluation and validation of the model's performance on both normal and fraudulent transactions.

Training of ML Model:

Training Set Splitting:

- Prior to model training for credit card fraud detection, the dataset is divided into training and testing sets. The `train_test_split` function is employed, dedicating 80% of the data to training and reserving 20% for testing.
- To maintain the distribution of normal and fraudulent transactions in both sets, the `stratify` parameter is utilized, preserving the integrity of the dataset and ensuring a balanced representation.

Feature Engineering and Scaling:

- Relevant columns for one-hot encoding and scaling are identified. Categorical features, such as 'Transaction Type,' 'Merchant ID,' and 'Card Type,' undergo one-hot encoding, while numerical features are scaled to standardize the data.
- This step is crucial for preparing the data in a format suitable for model training, ensuring that both categorical and numerical features contribute effectively to the model's ability to discern fraudulent patterns.

Testing of the ML model:

Introduction of Dummy Classifier:

- To establish a context for evaluating the credit card fraud detection model's performance, a Dummy Classifier with the 'stratified' strategy is introduced.
- The 'stratified' strategy ensures that the distribution of classes in the testing set is similar to that in the training set. This classifier serves as a benchmark against which the performance of the trained model can be compared.

Preprocessing Test Data:

- The preprocessor, derived from the optimized pipeline during training, is applied to preprocess the `X_test` data. This ensures that the testing data undergoes consistent preprocessing steps as the training data.
- Consistency in preprocessing is crucial for fair and meaningful performance comparison between the Dummy Classifier and the trained model.

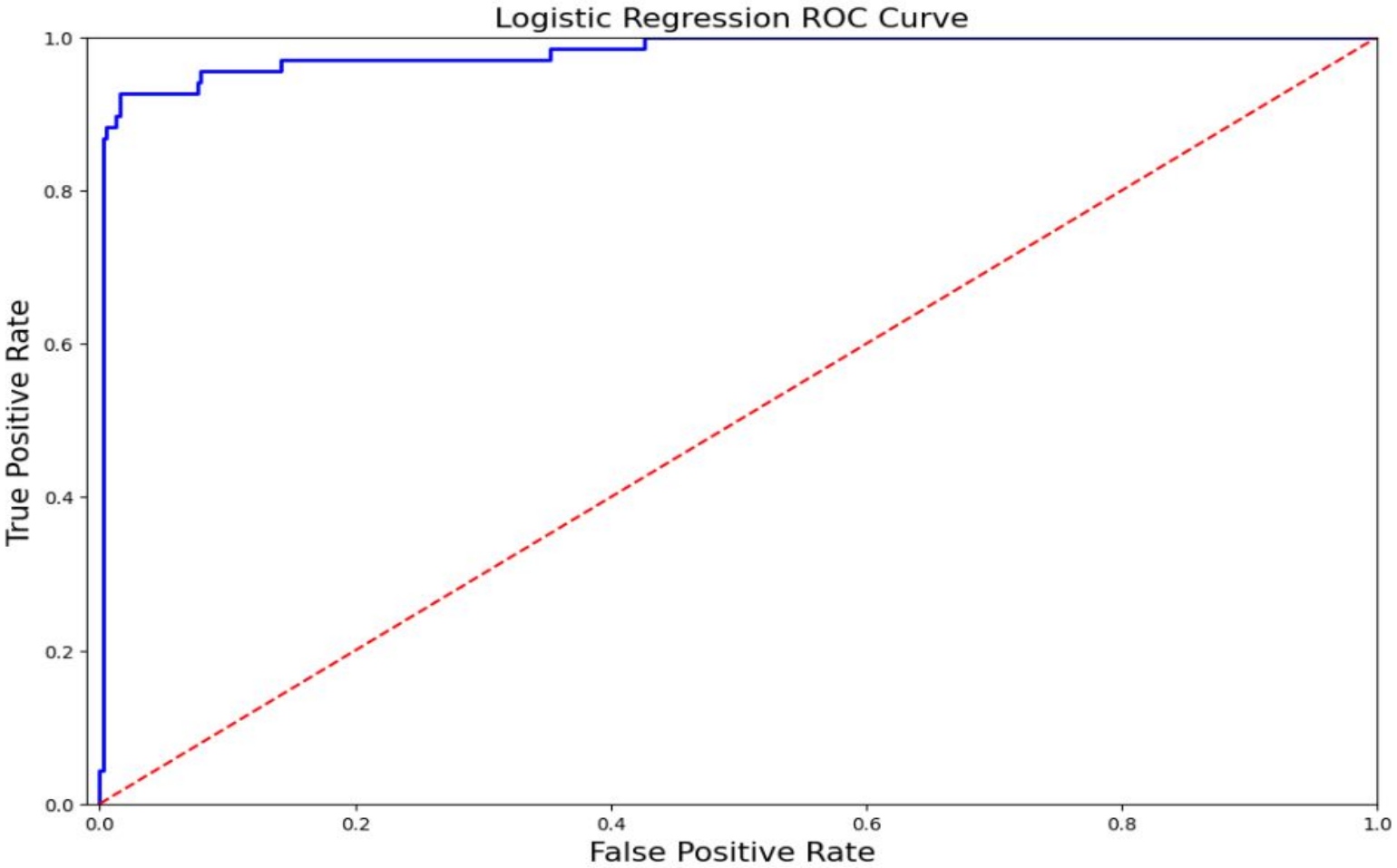
F1 Score Calculation:

The F1 scores for both classifiers are presented in a DataFrame (e.g., `f1_test_scores_credit_card_fraud_df`) for clear and concise comparison. Insights drawn from this testing chapter guide the interpretation of the model's real-world performance.

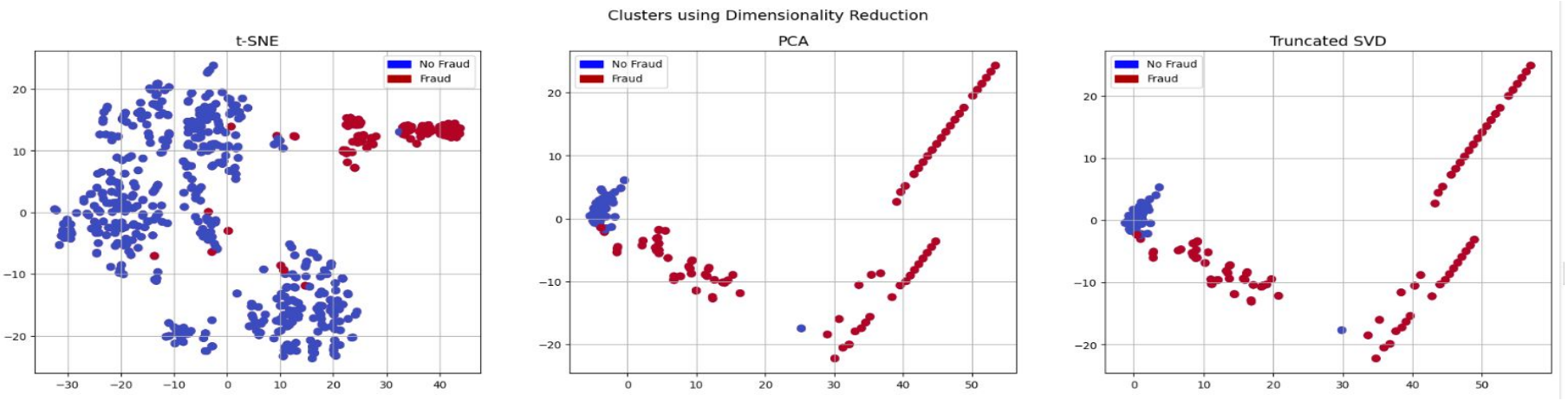
Reporting Results

1.Confusion matrices: In the evaluation of the credit card fraud detection model, a confusion matrix is employed to provide a detailed understanding of the model's performance. This matrix categorizes the predicted outcomes into four distinct scenarios: true positives (correctly identified fraud cases), true negatives (correctly identified non-fraud cases), false positives (non-fraud cases misclassified as fraud), and false negatives (fraud cases overlooked by the model). By visualizing these metrics, the confusion matrix offers a comprehensive overview of the model's ability to discriminate between fraudulent and legitimate transactions. This nuanced analysis is essential for assessing the model's strengths and weaknesses, enabling stakeholders to make informed decisions about the credit card fraud detection system's real-world applicability.

2.Precision-Recall Curve: In the context of credit card fraud detection using the provided code, the Precision-Recall Curve is a valuable tool for assessing the model's performance across various thresholds. The curve is constructed by plotting precision (positive predictive value) against recall (sensitivity or true positive rate) for different classification thresholds. This visualization offers insights into the trade-off between precision and recall, especially important in imbalanced datasets where the focus is often on correctly identifying the positive class (fraud) while minimizing false positives. The curve is generated by varying the decision threshold for classifying instances and computing precision and recall at each point. A higher area under the Precision-Recall Curve (AUC-PR) indicates a better-performing model. Analyzing this curve provides a nuanced understanding of the model's ability to identify fraud while maintaining precision, crucial for practical deployment where minimizing false positives is often a priority.


Logistic Regression ROC Curve
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0

3.Box Plots: In the given credit card fraud detection, scatter plots play a crucial role in visually representing the distribution of data points, specifically those generated through dimensionality reduction techniques such as T-SNE, PCA, and Truncated SVD. These scatter plots provide insights into the separation and clustering of fraud and non-fraud instances in a reduced feature space. Each point on the plot represents an observation, and the color distinguishes between fraud and non-fraud instances. T-SNE, PCA, and Truncated SVD are employed to visualize the data in two dimensions, offering a condensed representation of the original high-dimensional feature space. By examining these scatter plots, one can discern patterns, clusters, or separations between the two classes, aiding in the interpretation of the model's ability to capture meaningful distinctions between fraudulent and legitimate transactions.



4.Feature Importance: In the context of the credit card fraud detection code, The RandomForestClassifier inherently provides a feature importance score for each attribute in the dataset. These scores quantify the impact of each feature on the model's predictive performance. Higher importance scores indicate a more influential role in the model's decision-making process. The analysis involves extracting and ranking these importance scores, allowing for the identification of the most significant features contributing to fraud detection. This information is valuable for understanding the factors that heavily influence the model's predictions, aiding in interpretability and potentially guiding further feature engineering or data preprocessing steps.

Comparison of All Models

1. Classifier Performance: The below screenshot of my code of F1 scores along with accuracy,precision and Recall for different combinations of classifiers.Here in our case Random Forest is Best classifier in terms of Accuracy.

In Random Forest Classifier:

```
The model used is Random Forest classifier
The accuracy is 0.9991053455602773
The precision is 1.0
The recall is 0.7647058823529411
The F1-Score is 0.8666666666666666
```

In Logistic Regression:

```
--------------------------
Overfitting:

Recall Score: 0.94
Precision Score: 0.67
F1 Score: 0.78
Accuracy Score: 0.92
--------------------------
--------------------------
How it should be:

Accuracy Score: 0.89
Precision Score: 0.00
Recall Score: 0.17
F1 Score: 0.01
--------------------------
```

2.Model Comparison:

Logistic Regression:

- Working Principle: Logistic Regression is a binary classification algorithm that models the probability of a given instance belonging to a particular class. In credit card fraud detection, it predicts whether a transaction is fraudulent (class 1) or not (class 0).
- Key Features:
    - Sigmoid Function: Logistic Regression uses the sigmoid function to map the output of the linear combination of features to a probability score between 0 and 1.
    - Decision Threshold: A decision threshold is set (usually 0.5). If the predicted probability is above this threshold, the instance is classified as class 1; otherwise, it's classified as class 0.
- Interpretability: Logistic Regression provides a straightforward interpretation of the relationship between input features and the likelihood of fraud. It's particularly useful when understanding the impact of individual features on the prediction.

Random Forest Classifier:

- Working Principle: Random Forest is an ensemble learning algorithm that builds multiple decision trees during training. Each tree in the forest is trained on a random subset of features and a random subset of the training data. For credit card fraud detection, the ensemble of trees collectively makes a prediction.
- Key Features:
- Decision Trees: Each decision tree in the forest independently classifies instances. The final prediction is often determined by a majority vote or averaging the predictions of all trees.

- Feature Importance: Random Forest provides a measure of feature importance, indicating which features contribute more to the model's predictive power.
- Robustness: Random Forest is robust to overfitting, thanks to the combination of multiple trees. It can handle a large number of features and capture complex relationships within the data.

3. Analysis and Conclusion:

Analysis:

Logistic Regression:

- *Interpretability:* Logistic Regression provides a clear interpretation of the impact of individual features on fraud prediction. This makes it valuable for understanding the drivers behind suspicious transactions.
- *Decision Boundary:* The linear decision boundary of Logistic Regression is effective for scenarios where fraud patterns exhibit a more straightforward relationship with features.

Random Forest Classifier:

- *Ensemble Power:* Random Forest excels in capturing complex, non-linear relationships within the data. Its ensemble nature allows it to handle diverse and intricate fraud patterns that may not be easily captured by a single linear model.
- *Feature Importance:* The feature importance analysis from Random Forest helps identify key features contributing to fraud detection. This information is valuable for prioritizing focus on specific aspects of transactions.

Conclusion:

- **Complementary Strengths:** Both Logistic Regression and Random Forest bring unique strengths to the credit card fraud detection task. Logistic Regression's simplicity and interpretability make it a valuable tool for understanding basic patterns, while Random Forest's ensemble approach enhances the model's ability to discern intricate fraud schemes.
- **Model Selection Considerations:** The choice between Logistic Regression and Random Forest should be guided by the nature of the dataset. For datasets with clear linear separability, Logistic Regression may suffice. In contrast, when dealing with complex fraud scenarios involving various factors, Random Forest can offer superior predictive performance.
- **Hybrid Approaches:** Depending on the characteristics of the dataset, a hybrid approach leveraging both models could be explored. Logistic Regression could serve as an initial filter, and Random Forest could provide a secondary layer of analysis to capture nuanced fraud patterns.
- **Continuous Monitoring and Adaptation:** Given the evolving nature of fraud, continuous monitoring and adaptation of models are crucial. Regular updates to account for new fraud patterns and adjusting model parameters ensure the sustained effectiveness of the credit card fraud detection system.

Conclusion

In the realm of credit card fraud detection, the utilization of both Logistic Regression and Random Forest Classifier proves to be a judicious strategy. Logistic Regression, with its simplicity and interpretability, offers insights into straightforward fraud patterns, facilitating a clear understanding of feature impacts. On the other hand, the Random Forest Classifier's ensemble nature excels in capturing intricate, non-linear relationships within the data, enhancing its ability to discern complex fraud schemes. The hybrid deployment of these models provides a comprehensive approach, leveraging their complementary strengths. Logistic Regression serves as an effective initial filter, while Random Forest offers a secondary layer of analysis for nuanced fraud detection. This symbiotic integration results in a robust and adaptable credit card fraud detection system, well-equipped to tackle diverse and evolving fraud scenarios.

Future Work and Scope:

Real-time Fraud Detection: The integration of real-time data processing capabilities could be pivotal for promptly identifying and responding to emerging fraud trends. Both models could be adapted to operate in a streaming environment, ensuring swift detection and prevention of fraudulent transactions.

Explainability and Compliance: Enhancements in the interpretability of models, especially for Random Forest, could be pursued to meet regulatory compliance requirements. Developing methods to provide clear, human-understandable explanations for model decisions can instill confidence and facilitate regulatory adherence.

Integration with Advanced Technologies: Exploring the integration of emerging technologies like blockchain or federated learning could offer innovative solutions to secure sensitive credit card transactions while preserving privacy and decentralization.

Global Collaboration: Collaboration between financial institutions, cybersecurity experts, and data scientists on a global scale can lead to the development of a shared intelligence network. This collaborative approach can enhance the models' effectiveness by leveraging a broader and more diverse dataset.

References:

https://www.kaggle.com/code/janiobachmann/credit-fraud-dealing-with-imbalanced-datasets

https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud?resource=download

https://www.geeksforgeeks.org/ml-credit-card-fraud-detection/

Thank You