

Credit Card Fraud Detection Model

Using Logistic Regression

Final Project

Najia Jahan

Professor Erik K. Grimmelmann

CSC 44700

Date: 18 December 2023

Table of Contents

Abstract.....	2
Introduction.....	2
Logistic Regression	3
Mathematical Equation	3
Graphical Representation.....	4
Key Characteristics	4
Dataset.....	5
Code Analysis.....	6
Steps of the Process	6
Output and Matrices	8
Correlation Matrix.....	9
ROC Curve	10
Precision Recall Curve	11
Confusion Matrix.....	11
Challenges.....	12
Alternate Model Approach	13
Conclusion	15
References	16

Abstract

This report introduces a credit card fraud detection model utilizing logistic regression. Using a comprehensive credit card transaction dataset, we evaluate the model's performance in identifying fraudulent transactions. Results, measured through accuracy, precision, recall, F1-score, and Matthews Correlation Coefficient, demonstrate the effectiveness of logistic regression. Visualizations such as correlation matrices and confusion matrices offer insights into the model's behavior. This report not only emphasizes the applicability of logistic regression for credit card fraud detection but also delves into the challenges faced by the model. Additionally, it offers a comparison of the model's effectiveness with other machine learning approaches.

Introduction

Background and Significance: Credit card fraud is a growing problem in the digital age, posing a serious challenge to the financial industry. As fraudulent tactics become more sophisticated, the need for effective detection methods has never been more crucial. With technological advancements occurring daily, addressing this issue requires innovative solutions, and machine learning has emerged as a powerful tool in this endeavor. Leveraging the capabilities of machine learning algorithms becomes essential to stay ahead of evolving fraud techniques and ensure the security of financial transactions in our increasingly digital world.

Objectives of the Project: The main goals of this project are outlined to tackle the challenges in credit card fraud detection. By using logistic regression, a widely-used machine learning algorithm, our aim is to check how well it can tell apart real and fake transactions. The project aims to provide useful insights to help create more secure and flexible fraud detection systems.

Overview of Logistic Regression in Fraud Detection: This subsection offers a brief introduction to logistic regression, highlighting its relevance in the context of fraud detection. Logistic regression's ability to handle binary classification problems makes it an attractive choice for distinguishing between normal and fraudulent credit card transactions. An overview of its underlying principles sets the stage for the subsequent sections, where logistic regression will be applied and evaluated in the specific context of credit card fraud detection.

Logistic Regression

Logistic Regression is a powerful statistical method commonly employed for binary classification tasks, such as predicting whether an email is spam or not, or, as in this project, determining whether a credit card transaction is fraudulent. Unlike linear regression, which predicts continuous outcomes, logistic regression models the probability that a given instance belongs to a particular category.

Mathematical Equation¹:

The logistic regression model employs the logistic function (also known as the sigmoid function) to transform the linear combination of input features into a probability score between 0 and 1.

The logistic function is represented by the equation:

$$P(Y=1) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n)}}$$

Here,

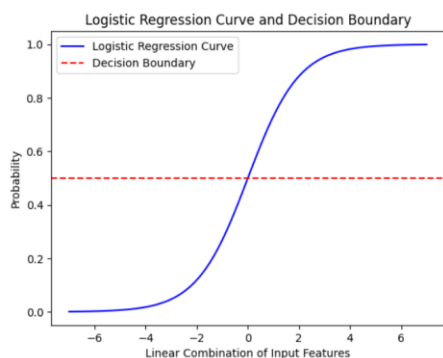
- $P(Y=1)$ is the probability of the instance belonging to Class 1 (fraudulent in our case).
- e is the base of the natural logarithm.

¹ "Sigmoid Function." Wikipedia, November 27, 2023. https://en.wikipedia.org/wiki/Sigmoid_function.

- β_0 is the intercept.
- $\beta_1, \beta_2, \dots, \beta_n$ are the coefficients associated with the input features X_1, X_2, \dots, X_n .

Graphical Representation

The sigmoid function generates an S-shaped curve, mapping any real-valued number to the range $[0, 1]$. This curve is essential for logistic regression, as it allows the model to output probabilities. The logistic regression decision boundary is set where the sigmoid function equals 0.5. Instances falling on one side of this boundary are predicted as Class 1, while those on the other side are predicted as Class 0.



In the graph, the x-axis represents the linear combination of input features, and the y-axis represents the predicted probability. As the linear combination increases, the probability approaches 1, and as it decreases, the probability approaches 0. The decision boundary (shown as a vertical line) determines the threshold for classification.

Key Characteristics




- Logistic regression is interpretable, providing coefficients that indicate the impact of each feature on the log-odds of the response variable.

- The model is particularly useful for datasets with a binary outcome and is well-suited for problems like credit card fraud detection.

In the context of this project, logistic regression serves as the predictive model, utilizing the characteristics of the dataset to discern between normal and fraudulent credit card transactions.

Dataset

The dataset utilized in this project provides a snapshot of credit card transactions conducted by European users in September 2013, spanning a concise two-day period. A total of 284,807 transactions are included, with only 492 identified as fraudulent, comprising a mere 0.172% of the dataset. To address confidentiality concerns, the dataset primarily includes features resulting from Principal Component Analysis (PCA) transformation, with 'Time' (representing seconds since the first transaction) and 'Amount' (indicating the transaction sum) being the exceptions. The crucial 'Class' feature serves as the label, denoting whether a transaction is fraudulent (Class 1) or non-fraudulent (Class 0). Below are two small snippets of the dataset used.

creditcard.csv (150.83 MB)																  	
Detail Compact Column																31 of 31 columns	
# Time	# V1	# V2	# V3	# V4	# V5	# V6	# V7	# V8	# V9	# V10	# V11						
0	-1.3598871336738	-0.0727811733098497	2.53634673796914	1.37815522427443	-0.338328769942518	0.462387777762292	0.239598554861257	0.8986979812610587	0.363786969611213	0.8987941719789316	-0.551599533268813						
0	1.19185711131486	0.26615871285963	0.16648811335321	0.448154878468911	0.0688176492822243	-0.0823688888155687	-0.0788829833323113	0.8851816549148184	-0.255425128189186	-0.1669744148804614	1.61272666105479						
1	-1.35835486159823	-1.34816387473689	1.77328934263119	0.379779593834328	-0.583198133318193	1.80049938879263	0.791468956458422	0.247675786588991	-1.51465432268583	0.287642865216696	0.624581459424895						
1	-0.966271711572887	-0.185226888882898	1.79299333957872	-0.863291275836453	-0.8183888796838823	1.24728316752486	0.23768893977178	0.377435874652262	-1.38782486278197	-0.8549519224713749	-0.226487263835481						
2	-1.15823389349523	0.877736754848451	1.54871784651121	0.483833939355121	-0.487193377311653	0.8959214624684256	0.592948745385545	-0.278532677192282	0.817739388235294	0.753874431976354	-0.822842877946363						
2	-0.425965884412454	0.968523844882985	1.14118934232219	-0.168252879768382	0.42899688877219	-0.8297275516639742	0.476288948728027	0.268314333874874	-0.56867137571251	-0.371487196834471	1.34126198801957						
4	1.22965763458793	0.141883587049326	0.8453787735899449	1.28261273673594	0.191888988597645	0.272788122899898	-0.08515980288258983	0.8812129398838894	0.464959994783886	-0.8992543211289237	-1.41698724314928						
7	-0.644269442348146	-1.41796354547385	1.874388376355615	-0.492199818495015	0.948934894764157	0.428118462833889	1.12863135838353	-3.88786423873589	0.615374738667827	1.24937617815176	-0.619467796121913						
7	-0.89428688228282	0.286157196276544	-0.113192212729871	-0.271526138888684	2.66959865959867	3.72181886112751	0.378145127676916	0.851884443280895	-0.392847586798684	-0.418438432848439	-0.785116586646536						
9	-0.33826175242575	1.11959337641566	1.84436655157316	-0.222187276738296	0.49936888649727	-0.24676118861991	0.651583286489972	0.8695385865186387	-0.736727316364189	-0.366845639286541	1.81761446783262						

creditcard.csv (150.83 MB)

Detail Compact Column 31 of 31 columns

# V19	# V20	# V21	# V22	# V23	# V24	# V25	# V26	# V27	# V28	# Amount	# Class
0.483992968255733	0.251412898239785	-0.018386777944153	0.277837575558899	-0.118473918188767	0.0669288749146731	0.128539358273528	-0.189114843888824	0.133558376748387	-0.0218538534538215	149.62	0
-0.145783841325259	-0.0698831352238283	-0.225775248833138	-0.638671952771851	0.181288821253234	-0.339846475529127	0.167178484418143	0.125894532368176	-0.08898399914322813	0.0147241691924927	2.69	0
-2.26185789538414	0.524979725224484	0.247998153469754	0.771679481917229	0.989412262347719	-0.689288956498685	-0.327641833735251	-0.139896571514147	-0.8553527948384261	-0.0597518485929284	378.66	0
-1.2326219788892	-0.288837781160366	-0.108388452835545	0.08527359678253453	-0.198328518742841	-1.17557533186321	0.647376834682838	0.221928844458487	0.062722848729383	0.0614576285806353	123.5	0
0.883486924968175	0.488542368392758	-0.08943869713232919	0.79827849458971	-0.137458879619863	0.141266983824769	-0.286889587619756	-0.582292224181569	0.219422229513348	0.215153147499286	69.99	0
-0.0331937877876282	0.0849676728682849	-0.288253514656728	-0.559824796253248	-0.0263976679795373	-0.371426583174346	-0.232793816737834	0.185914779097957	0.253844224739337	0.0818882569229443	3.67	0
-0.0455758446637976	-0.21963255278686	-0.167716265815783	-0.278789726172363	-0.154183786889385	-0.788055415884671	0.75813693588659	-0.257236845917139	0.0345874297438413	0.08516776898624916	4.99	0
0.324584731321494	-0.156741852488285	1.94346533978412	-1.01545478979971	0.057583529867291	-0.649789805559993	-0.415266566234811	-0.8516342969262494	-1.28692188894258	-1.08533918832377	48.8	0
0.57832816746536	0.0527356691149697	-0.0734251081059225	-0.268891632235551	-0.284232669947878	1.0115918818785	0.373284688146282	-0.384157387782294	0.0117473564581996	0.14240432992147	93.2	0
0.451772964394125	0.283711454727929	-0.24691393691888	-0.633752642486113	-0.12879488488185	-0.385849925313426	-0.0697338468416923	0.0941988339514961	0.246219384619926	0.8838756493473326	3.68	0

2

Code Analysis

Steps of the Processes

- 1. Import Libraries & Data Exploration:** Import necessary libraries such as NumPy, Pandas, Seaborn, Matplotlib, and scikit-learn modules for logistic regression and model evaluation metrics. In the initial phase of the project, a comprehensive credit card transaction dataset comprising 284,807 transactions was loaded. This was achieved using the following code:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, accuracy_score, precision_score, recall_score, f1_score, matthews_corrcoef

# Load the dataset
data = pd.read_csv("creditcard.csv")

# Understanding the Data
print(data.shape)
print(data.describe())
```

² Machine Learning Group - ULB, "Credit Card Fraud Detection," Kaggle, March 23, 2018, <https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud/data>.

The dataset's statistical summaries and class distribution were then investigated, providing valuable insights into the distribution of fraudulent and valid transactions.

2. **Data Preprocessing:** To prepare the data for model training, a series of preprocessing steps were undertaken. A heatmap was generated to visualize correlations among the dataset features:

```
# Plot the Correlation Matrix
corrmat = data.corr()
fig = plt.figure(figsize=(12, 9))
sns.heatmap(corrmat, vmax=.8, square=True)
plt.show()
```

This visualization aided in identifying potential relationships among features.

Subsequently, the dataset was split into features (X) and the target variable (Y), forming the basis for subsequent logistic regression modeling:

```
# Separating the X and Y values
X = data.drop(['Class'], axis=1)
Y = data['Class']
print(X.shape)
print(Y.shape)
xData = X.values
yData = Y.values
```

3. **Model Building:** The logistic regression algorithm was chosen as the predictive model for credit card fraud detection:

```
# Building the Logistic Regression Model
log_reg = LogisticRegression(max_iter=1000) # Handling potential convergence issues
log_reg.fit(xTrain, yTrain)
```

During the model-building phase, attention was given to potential convergence issues by increasing the iteration limit, ensuring the model's stability and effectiveness.

4. **Evaluation Metrics:** The effectiveness of the logistic regression model was evaluated using a set of key metrics. The following code segment assesses model performance and prints key metrics:


```
# Evaluating the classifier
n_outliers = len(fraud)
n_errors = (yPred != yTest).sum()
print("The model used is Logistic Regression")

acc = accuracy_score(yTest, yPred)
print("The accuracy is {}".format(acc))

prec = precision_score(yTest, yPred)
print("The precision is {}".format(prec))

rec = recall_score(yTest, yPred)
print("The recall is {}".format(rec))

f1 = f1_score(yTest, yPred)
print("The F1-Score is {}".format(f1))

MCC = matthews_corrcoef(yTest, yPred)
print("The Matthews correlation coefficient is {}".format(MCC))
```

These metrics, including accuracy, precision, recall, F1-score, and the Matthews correlation coefficient, provided a comprehensive assessment of the model's performance in credit card fraud detection.

Output and Matrices

Model Output:

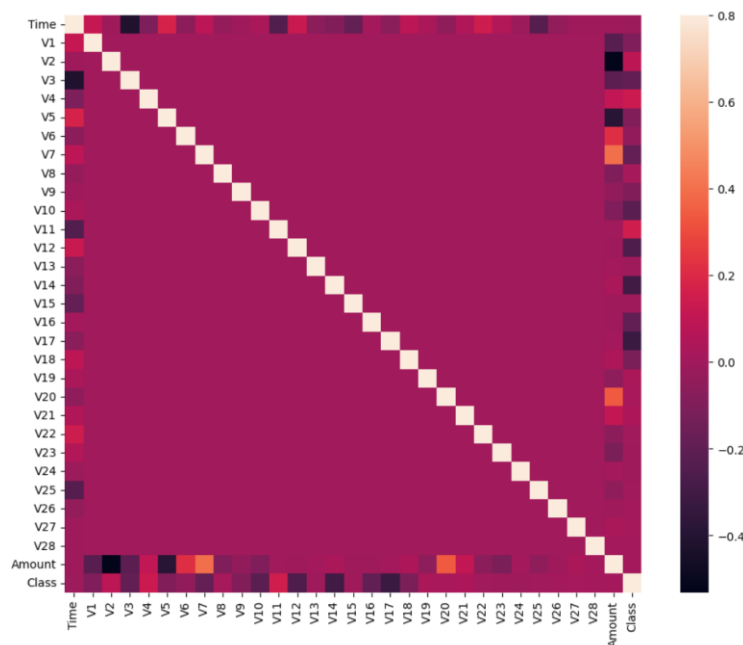
```
The model used is Logistic Regression
The accuracy is 0.9989291106351603
The precision is 0.8135593220338984
The recall is 0.4897959183673469
The F1-Score is 0.6114649681528662
The Matthews correlation coefficient is 0.630785580957534
```

- **Accuracy:** Logistic Regression achieves an impressive 99.89% accuracy, showing strong correctness in classifying transactions.
- **Precision:** With 81.36% precision, the model accurately identifies fraud, minimizing false alarms.
- **Recall:** Recall, also known as sensitivity or true positive rate, indicates the model's ability to capture actual instances of fraud. A recall of 48.98% means that the model

successfully identified almost half of the total fraudulent transactions present in the dataset.

- **F1-Score:** Balancing precision and recall, the F1-Score is 61.15%.
- **MCC:** The Matthews Correlation Coefficient is 63.08%, indicating the model's effectiveness in handling imbalanced data.
- **Efficiency:** Consideration of running time and computational efficiency is crucial. If the model maintains high accuracy and proves to be efficient, it's well-suited for real-time credit card fraud detection systems.

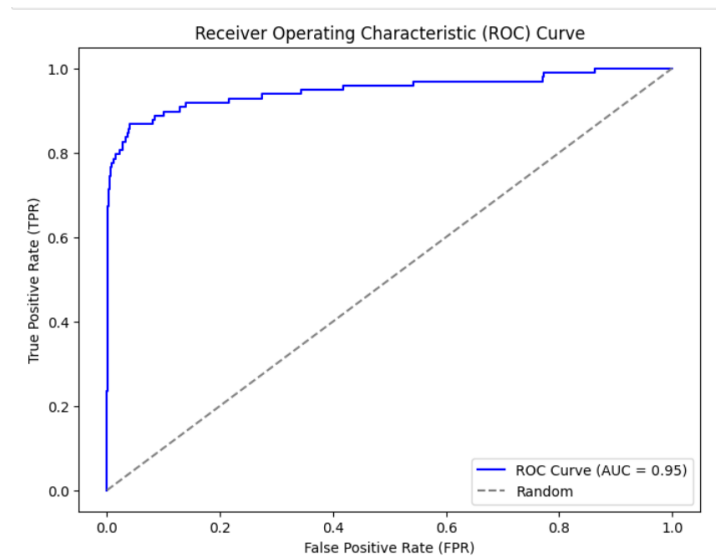
Correlation Matrix (Heatmap):



This correlation matrix is like a map that shows how different features in our fraud detection data relate to each other. Each entry in the matrix tells us the strength and direction of the relationship between two features. If the number is close to 1, it means they move together in the same direction, and if it's close to -1, it means they move in opposite directions. The correlation of 0

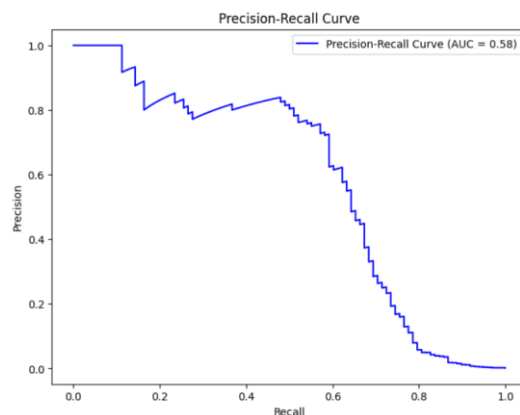
suggests no linear relationship. By looking at this matrix, we can spot patterns and understand which features might be influencing each other. Although most of the features have very less correlation, but some of the features are correlated such as feature V2 has a negative correlation with feature Amount.

Receiver Operating Characteristics ROC Curve:



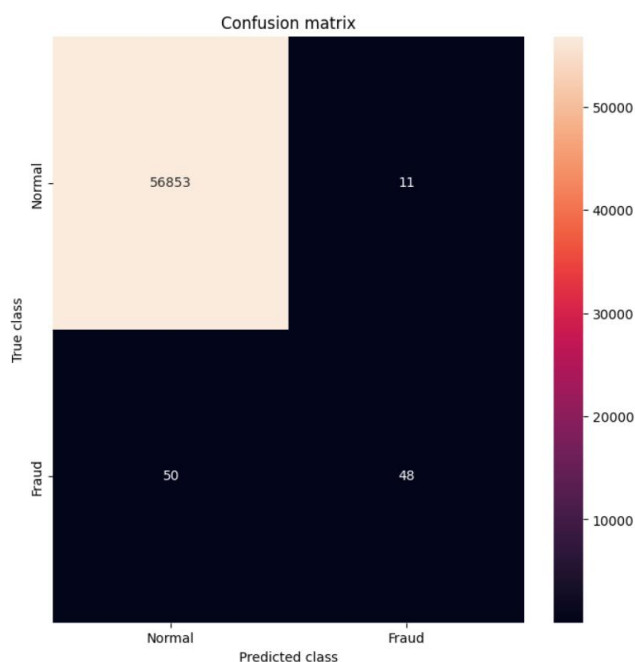
With an AUC of 0.95 for the ROC curve, our fraud detection model demonstrates a strong ability to distinguish between genuine and fraudulent transactions. The high AUC value, close to 1, indicates that the model effectively balances correctly identifying fraud (true positives) while minimizing false alarms (false positives). This means that, overall, the model is proficient at ranking and classifying transactions, making it a robust tool for detecting potential instances of fraud.

Precision Recall Curve:



With a Precision-Recall curve AUC of 0.58, our fraud detection model is not as balanced in catching all fraud while avoiding mistakes. The line coming down from the top corner (1,1) shows that, as we try to catch more fraud, we start to make more errors by flagging normal transactions as fraud. It's like a trade-off – catching more fraud might mean accepting more false alarms. The steeper the line, the more this trade-off matters, indicating how sensitive the model is to changes in how it classifies transactions.

Confusion Matrix:



The confusion matrix highlights the model's effectiveness in practical terms. It correctly identified 56,853 normal transactions and 48 fraudulent transactions, showcasing its proficiency. However, 11 false positives and 50 false negatives suggest areas for improvement. Evaluating these values provides insight into the model's overall performance and areas that may need refinement for enhanced effectiveness.

Challenges

Some of the key challenges of building this project are:

- **Imbalanced Data:** The dataset is heavily imbalanced, with only 0.2% being fraudulent and 99.8% non-fraudulent transactions. This creates a big problem for logistic regression, as it tends to lean towards predicting the majority class. It struggles to accurately catch the rare fraud cases, impacting how well it can find fraud and how often it gets it right.³
- **Interpretability vs. Complexity:** Balancing the model's interpretability with its complexity posed a crucial challenge, ensuring both understanding and accuracy.
- **Threshold Setting:** Determining the right threshold for classifying transactions was essential, impacting precision, recall, and overall model performance.
- **Ethical and Regulatory Compliance:** Navigating ethical and regulatory considerations in handling sensitive financial data added complexity to model development.
- **Adaptability to Fraud Tactics:** Building a model capable of adapting to evolving fraud tactics required continuous monitoring and updates for sustained effectiveness.

³ Jason Brownlee, "Logistic Regression for Machine Learning," MachineLearningMastery.com, December 5, 2023, <https://machinelearningmastery.com/logistic-regression-for-machine-learning/>.

Alternate Model Approach

While we chose logistic regression for its simplicity and speed, it's crucial to acknowledge other models with unique strengths. Our decision balanced interpretability and efficiency. To optimize further, we're exploring alternative approaches for credit card fraud detection, each with distinct attributes. Some of the other effective machine learning models that could be considered for this project are:

- **Decision Trees:**
 - **Strengths:** Decision Trees are known for their interpretability and the ability to handle non-linear patterns in data. However, they may be prone to overfitting, where the model captures noise in the training data.
 - **Comparison to Logistic Regression:** Logistic Regression is simpler, less prone to overfitting, and remains interpretable. The choice between Decision Trees and Logistic Regression depends on the trade-off between complexity and interpretability.
- **Random Forest:**
 - **Strengths:** Random Forests can be considered as advanced versions of Decision Trees. They offer improved performance by constructing multiple trees and averaging the results, which helps mitigate overfitting.
 - **Comparison to Logistic Regression:** While Logistic Regression is simpler, Random Forests may outperform it on more intricate patterns, but at the cost of increased complexity.
- **Neural Networks:**

- **Strengths:** Neural Networks, as deep learners, excel at handling complex tasks and extracting intricate patterns from data.⁴
- **Comparison to Logistic Regression:** Logistic Regression is computationally simpler and more interpretable. Neural Networks, on the other hand, are computationally intensive and less interpretable. The choice depends on the desired trade-off between complexity and performance.
- **Support Vector Machines (SVM):**
 - **Strengths:** SVMs are powerful for capturing complex relationships in data and are effective in high-dimensional spaces.⁵
 - **Comparison to Logistic Regression:** Logistic Regression is computationally simpler and easier to interpret. SVMs might be considered overkill for simpler datasets, but they can outperform Logistic Regression in capturing intricate relationships.
- **K-Nearest Neighbors (KNN):**
 - **Strengths:** KNN is an intuitive algorithm that classifies data points based on the majority class among their neighbors.⁶
 - **Comparison to Logistic Regression:** Logistic Regression is more robust with cleaner, well-structured data and provides clear probabilities. KNN can be sensitive to noisy data and may require careful preprocessing.
- **XGBoost (Extreme Gradient Boosting):**

⁴ How to improve fraud detection with machine learning. Accessed December 15, 2023. <https://datadome.co/learning-center/fraud-detection-machine-learning/>.

⁵ Ms. Madhuri B. Thorat and Purva D. Netke2, International Journal of Research Publication and Reviews 04, no. 01 (2022): 1806–12, <https://doi.org/10.55248/gengpi.2023.4149>.

⁶ Yiyang Dong et al., A Machine Learning Model for Product Fraud Detection Based On SVM, accessed December 15, 2023, <https://ieeexplore.ieee.org/document/9479632/>.

- **Strengths:** XGBoost is a robust performer, particularly useful for handling imbalanced data and achieving high accuracy.
- **Comparison to Logistic Regression:** While Logistic Regression is simpler, XGBoost offers better accuracy, especially in scenarios with imbalanced classes, but at the cost of increased complexity.⁷

In summary, the choice of the model depends on factors like the complexity of patterns in the data, interpretability requirements, computational resources, and the nature of the dataset.

Conclusion

In conclusion, the credit card fraud detection model built on logistic regression has demonstrated high effectiveness, achieving an impressive 99.89% accuracy with balanced precision and recall. Despite facing challenges related to imbalanced data and ethical considerations, logistic regression proves its relevance in addressing fraud detection.

However, upon closer examination of the confusion matrix, there are indications of room for improvement, suggesting that alternative models like random forest or neural networks might offer enhanced efficacy in capturing fraudulent transactions. The exploration of alternative models highlighted trade-offs in interpretability, complexity, and computational intensity, emphasizing the importance of selecting models tailored to specific project requirements.

This project not only provides valuable insights for transaction security enhancement but also underscores the continuous need for adaptation and vigilance in combating evolving fraud tactics.

⁷ "What Is XGBoost?," NVIDIA Data Science Glossary, accessed December 18, 2023, <https://www.nvidia.com/en-us/glossary/data-science/xgboost/>.

Works Cited

“Sigmoid Function.” Wikipedia, November 27, 2023.

https://en.wikipedia.org/wiki/Sigmoid_function.

Machine Learning Group - ULB, “Credit Card Fraud Detection,” Kaggle, March 23, 2018,

<https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud/data>.

Jason Brownlee, “Logistic Regression for Machine Learning,” MachineLearningMastery.com,

December 5, 2023, <https://machinelearningmastery.com/logistic-regression-for-machine-learning/>.

Yiyang Dong et al., A Machine Learning Model for Product Fraud Detection Based On SVM,

accessed December 15, 2023, <https://ieeexplore.ieee.org/document/9479632/>.

How to improve fraud detection with machine learning. Accessed December 15, 2023.

<https://datadome.co/learning-center/fraud-detection-machine-learning/>.

“What Is XGBoost?” NVIDIA Data Science Glossary. Accessed December 18, 2023.

<https://www.nvidia.com/en-us/glossary/data-science/xgboost/>.