**FINAL PROJECT REPORT:**

**FRAUD CREDIT CARD TRANSACTION DETECTION**

**ABSTRACT**

The goal of this project is to assist credit card companies in identifying and preventing fraudulent transactions before they are completed. In the payments industry, fraud typically involves the unauthorized use of a card or card number, often without the cardholder's knowledge. This can lead to disputes, investigations, and chargebacks, which not only affect the customer experience but also impact the financial institution’s credibility.

That’s why it's so important for credit card companies to proactively detect suspicious activity ensuring customers aren’t unfairly charged and can continue to trust their service provider.

Why Machine Learning?

Fraud detection is a classic case where machine learning can make a real difference. Predicting customer behavior and unusual patterns is a widely used application in domains like finance, marketing, and risk management. By building intelligent predictive models, organizations can make more informed decisions and better protect both themselves and their customers.

My Approach

To tackle this use case, I tested several machine learning algorithms:

Logistic Regression

Random Forest

Support Vector Machine (SVM)

XGBoost

Deep Learning using Keras

**CONTENTS**

**Abstract……………………………………………………………………………........ 2**

**Introduction……………………………………………………………………….…… 4**

**Research Question ……………………………………………………………….……. 4**

**Data Appraisal…………………………………………………………………….…… 5**

**Techniques………………………………………………………...……………….…... 7**

**Feature Engineering…………………………………………………………………... 9**

**Data Modelling and Evaluation……………………………………………………… 10**

**Machine Learning Model…………………………………………………………….. 11**

**Evaluation Metrics And Model Evaluation…………………………………………. 14**

**Conclusion……………………………………………………………………………... 17**

**INTRODUCTION**

It's crucial for credit card companies to recognize fraudulent transactions early, so that customers aren’t unfairly charged for purchases they didn’t make. Ensuring this not only protects customers financially but also helps maintain their trust and satisfaction with the company.

In this project, we explore how different features in the dataset relate to a customer’s transaction behavior, and whether certain patterns in credit card usage can help us identify fraudulent activity.

At its core, this is a classification problem — our goal is to predict whether a transaction is fraudulent (the dependent variable, or "Class") using information from the other features. Since many of the variables are either categorical or can be interpreted as such, a classification approach fits this problem well.

**Dataset description and details:**

The dataset used for this project contains credit card transactions made by European cardholders during a two-day period in September 2013. Out of 284,807 total transactions, only 492 were marked as fraudulent meaning that fraudulent cases made up just 0.172% of the data. This highlights a significant class imbalance, which posed a major challenge in training models to accurately detect fraud.

To protect confidentiality, most of the features have been transformed using Principal Component Analysis (PCA) and are labeled as V1 through V28. This transformation helps preserve the privacy of sensitive information while still allowing for meaningful analysis. Only two features ‘Time’ and ‘Amount’ were left in their original form.

The ‘Time’ feature represents the number of seconds elapsed since the first transaction in the dataset.

The ‘Amount’ feature reflects the transaction value and can be especially useful in cost-sensitive learning models.

The target variable, ‘Class’, indicates whether a transaction is fraudulent (1) or legitimate (0), making this a classic binary classification problem.

**RESEARCH QUESTION: “Can we predict the fraudulent credit card transaction using machine learning?”**

**Reasoning and Hypothesis:**

In this project, I explored several predictive models to evaluate how well they can distinguish between legitimate transactions and fraudulent ones. Although the dataset has been anonymized for privacy—meaning the feature names aren’t disclosed and most variables have been scaled using PCA—I was still able to analyze key patterns and insights from the data to guide model development and evaluation.

**My Goal:**

**1. Understanding the data distribution**

The first step in my project was to get a sense of the data. One of the first things I noticed was that fraud cases were extremely rare—only about 0.17% of all transactions—making the dataset highly imbalanced from the start This helped guide how I approached both training and evaluation going forward.

**2. Balancing the dataset with NearMiss**

To train more reliable models, I created a balanced dataset using the NearMiss algorithm. This technique undersamples the majority class (non-fraud) to match the number of fraud cases, giving me a 50/50 split between fraud and non-fraud transactions. This helped ensure the models didn’t just learn to always predict "non-fraud."

**3. Choosing and evaluating classifiers**

I tried out a few different classification models like Logistic Regression, Random Forest, and XGBoost to see which one could best detect fraudulent transactions. For each, I evaluated how well they could detect frauds, and compared their performance to find the most effective one.

**4. Building a Neural Network**

To push the results further, I also built a simple neural network model. Next, I compared its performance with the best-performing traditional machine learning model to understand whether deep learning actually offered any added benefit for this particular problem.

**5. Addressing common pitfalls in imbalanced data**

I was careful to avoid common mistakes like relying on overall accuracy, which can be misleading when one class dominates. Instead, I focused on metrics that matter in fraud detection: precision, recall, and especially the Area Under the Precision-Recall Curve (AUPRC). This gave a more realistic picture of how well the models handled rare but critical fraud cases.

**DATA APPRAISAL**

**Attribute Characteristics:** Numerical

**Missing value:** No

**Associated Task:** Classification

**Understanding of data:**

The data set is quite big with the Size of 153MB,

No. of entries: 284807

No. of features with Target column: 31

Graphical user interface, text, application

Description automatically generated

**Graphical user interface

Description automatically generated with low confidence**

Graphical user interface, application, table, Excel

Description automatically generated

Graphical user interface, text, application

Description automatically generated with medium confidence

A picture containing diagram

Description automatically generated

**Plotting a Bar chart to show the count of both the classes.**

Chart

Description automatically generated

Graphical user interface

Description automatically generated

**TECHNIQUES**

**Correlation Matrix-** Plotting the correlation matrix to check how the features are correlated with each other.

**Checking the distribution of data in the negative classes.**

Chart, box and whisker chart

Description automatically generated

**Checking the distribution of data in the positive classes.**

Chart, box and whisker chart

Description automatically generated

**DATA UNDERSTANDING**

**Below is the observation I found in the dataset.**

**immediate use.**

**Highly Imbalanced Data:** Fraudulent transactions make up only about 0.17% of the total data, which creates a significant imbalance—roughly a 99:1 ratio of normal to fraud cases.

**PCA-Transformed Features:** Most of the features have been anonymized and transformed using Principal Component Analysis (PCA) for confidentiality reasons.

**Negative Correlations:** Features like V17, V14, V12, and V10 show a negative correlation with fraud. In other words, lower values in these features tend to be linked with a higher likelihood of fraudulent activity.

**Positive Correlations:** On the other hand, features such as V2, V4, V11, and V19 are positively correlated with fraud—higher values in these features are more commonly associated with fraudulent transactions.

**FEATURE ENGINEERING**

**Text, application

Description automatically generated**

**Text

Description automatically generated with medium confidence**

**Text

Description automatically generated**

**Table

Description automatically generated with medium confidence**

**RANDOM SAMPLING:**

**Text

Description automatically generated**

**Graphical user interface

Description automatically generated**

**MACHINE LEARNING MODEL**

**Graphical user interface, text, application, email

Description automatically generated**

**PRELIMINARY RESULTS:**

**A picture containing text

Description automatically generated**

**Text

Description automatically generated**

**Text

Description automatically generated**

**Graphical user interface, chart

Description automatically generated**

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

**EVALUATION METRICS AND MODEL EVALUATION**

Table

Description automatically generated

Graphical user interface, application, treemap chart, PowerPoint

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

A picture containing text, receipt

Description automatically generatedA screenshot of a computer

Description automatically generated with low confidence

**DEEP LEARNING:**

Text

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Table

Description automatically generated with medium confidence

Table

Description automatically generated with low confidence

**CONCLUSION**

After applying SMOTE to address the imbalance in our dataset where legitimate transactions far outnumber fraudulent ones we saw some improvement. However, it's important to note that our neural network model on the oversampled data sometimes performed worse at catching fraudulent transactions compared to the model trained on the under-sampled data.

One key difference is that we applied outlier removal only to the under-sampled dataset, not the oversampled one. Interestingly, while the under-sampled model did better at detecting fraud, it struggled to correctly identify non-fraudulent transactions. This means it occasionally flagged regular purchases as fraudulent.

Imagine a customer’s normal purchase getting incorrectly flagged and their card being blocke that would lead to frustration, increased complaints, and potential dissatisfaction with the financial institution.

To improve our model’s performance, the next logical step is to apply outlier removal to the oversampled dataset too. By doing this, we aim to see whether cleaning the data in this way helps the model make more accurate predictions especially by reducing false alarms on genuine transactions while still catching fraud effectively. The goal is to strike a better balance in identifying both fraud and non-fraud cases with higher confidence.