Deployment of Optimized Model for Fraud Detection

A Course Project report submitted

In partial fulfillment of requirement for the award of degree

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

By

A. Jasper Enoch	2203A51535
G. Varshitha	2203A51675
P. Laxmi Prasanna	2203A51423
K. Amrutha	2203A51747

Under the guidance of

Dr. Soumik Podder

Assistant Professor, School of CS&AI.

SR University

Ananthasagar, Warangal.

CERTIFICATE

This is to certify that this project entitled **"Deployment of Optimized Model for Fraud Detectionon"** is the bonafide work carried out by **A. Jasper Enoch, G. Varshitha, P. Laxmi Prasanna, K. Amrutha** as a Course Project for the partial fulfillment to award the degree **BACHELOR OF TECHNOLOGY** in **School of Computer Science and Artificial Intelligence** during the academic year 2023-2024 under our guidance and Supervision.

Dr. Soumik Podder
Assistant Professor,
School of CS&AI,
SR University
Anathasagar, Warangal

Prof. Sheshikala Martha
Professor & Head,
School of CS&AI,
SR University
Ananthasagar, Warangal.

Reviewer-1
Name:
Designation:
Signature:

Reviewer-2
Name:
Designation:
Signature:

ACKNOWLEDGEMENT

We owe an enormous debt of gratitude to our Course project guide DR**. SOUMIK PODDER, Assistant Professor** as well as Head of the School of CS&AI, **Prof. Sheshikala Martha,** for guiding us from the beginning through the end of the Course Project with their intellectual advices and insightful suggestions. We truly value their consistent feedback on our progress, which was always constructive and encouraging and ultimately drove us to the right direction.

We express our thanks to AI & ML course coordinator, Dr. Arpita Baronia, for her encouragement and support.

Finally, we express our gratitude and sincere thanks to all the teaching and non-teaching staff of the School of Computer Science & Artificial Intelligence, for their suggestions and timely support.

TABLE OF CONTENTS

ABSTRACT

Financial fraud poses a significant threat to businesses and individuals alike, necessitating the development of effective detection mechanisms. This project aims to leverage machine learning techniques to build a robust model capable of accurately identifying fraudulent financial transactions. The dataset under consideration comprises various features related to transactions, such as the amount, category, and fraud status.

Exploratory data analysis is conducted to gain insights into the data distribution and identify potential patterns or anomalies. Techniques such as oversampling and feature scaling are employed to address class imbalance and data normalization, respectively, ensuring that the models are trained on a representative dataset.

Several state-of-the-art classification algorithms, including K-Nearest Neighbors, Random Forest, and XGBoost, are evaluated and compared using comprehensive metrics such as classification reports, confusion matrices, and ROC-AUC curves. These metrics provide a comprehensive understanding of the models' performance, enabling the selection of the most effective approach for accurate fraud detection.

The ultimate goal of this project is to develop a reliable and efficient model that can be integrated into financial systems, enabling proactive detection and prevention of fraudulent activities. By leveraging the power of machine learning, businesses can mitigate financial losses, enhance customer trust, and maintain a secure financial ecosystem.

ABOUT THE ORGANISATION

SR University is a private university located in Warangal, Telangana, India. It was established in 2018 under the Telangana State Private Universities (Establishment and Regulations) Act 2018. SR University is accredited with an 'A' grade by the National Assessment and Accreditation Council (NAAC).

SR University offers a variety of undergraduate and postgraduate programs in engineering, technology, management, commerce, and arts. The university has a strong focus on industry-relevant education and offers a variety of opportunities for students to gain hands-on experience through internships, projects, and workshops. SR University also has a strong incubation center that supports students in developing and launching their startups.

SR University has a well-equipped campus with state-of-the-art facilities, including classrooms, laboratories, libraries, sports facilities, and hostels. The university also has a strong commitment to research and has published several papers in reputed journals and conferences.

SR University has a good placement record. In 2023, the university achieved 90% placements for its engineering students. The university has a strong alumni network that includes several successful entrepreneurs and professionals.

Overall, SR University is a good choice for students who are looking for an industry-relevant education and a strong focus on innovation and entrepreneurship.

INTRODUCTION

In today's digital age, financial transactions have become an integral part of our daily lives. From online shopping to electronic fund transfers, the convenience of these transactions has revolutionized the way we conduct business and manage our finances. However, this convenience comes with a significant risk – the threat of financial fraud. Fraudulent activities, such as unauthorized access, identity theft, and data breaches, can have severe consequences for individuals and organizations alike, resulting in substantial financial losses, compromised data integrity, and erosion of consumer trust.

To combat this growing menace, financial institutions and researchers have turned to advanced analytics and machine learning techniques to develop robust fraud detection systems. These systems leverage the power of data and algorithms to identify patterns and anomalies indicative of fraudulent behavior, enabling proactive measures to mitigate risks and protect financial assets.

This project aims to contribute to the ongoing effort by developing a comprehensive machine learning model for detecting fraudulent financial transactions. By leveraging a dataset encompassing various features related to transactions, such as the amount, category, and fraud status, the project employs a systematic approach to data exploration, preprocessing, and model development.

Exploratory data analysis plays a crucial role in understanding the data distribution, identifying potential patterns, and uncovering insights that can inform the modeling process. Techniques such as oversampling and feature scaling are employed to address class imbalance and data normalization, ensuring that the models are trained on a representative and standardized dataset.

Several state-of-the-art classification algorithms, including K-Nearest Neighbors, Random Forest, and XGBoost, are evaluated and compared using comprehensive metrics such as classification reports, confusion matrices, and ROC-AUC curves. These metrics provide a holistic view of the models' performance, enabling the selection of the most effective approach for accurate fraud detection.

By leveraging the power of machine learning, this project aims to develop a reliable and efficient model that can be integrated into financial systems, enabling proactive detection and prevention of fraudulent activities. The successful implementation of such a model can not only mitigate financial losses but also enhance consumer trust and contribute to the overall security and integrity of the financial ecosystem.

.

.

PROBLEM STATEMENT

Background:

Financial fraud is a persistent and evolving threat that poses significant risks to individuals, businesses, and the global economy. With the rapid growth of digital transactions and the increasing sophistication of fraudulent activities, there is an urgent need for robust fraud detection systems. Traditional rule-based systems have proven inadequate in keeping pace with evolving fraud patterns, necessitating the adoption of advanced analytical techniques.

Objective:

The primary objective of this project is to develop an effective machine learning model capable of accurately identifying fraudulent financial transactions. By leveraging a comprehensive dataset containing various features related to transactions, such as the amount, category, and fraud status, the goal is to build a predictive model that can reliably differentiate between legitimate and fraudulent transactions.

Dataset:

The project utilizes a dataset comprising various features related to financial transactions, including the transaction amount, category, and a label indicating whether the transaction is fraudulent or legitimate. The dataset is expected to contain a significant class imbalance, with fraudulent transactions representing a small fraction of the overall data.

Model Development:

To address the challenges of class imbalance, feature relevance, and evolving fraud patterns, the project employs a comprehensive approach involving:

1. Exploratory Data Analysis: Thorough exploration and visualization of the dataset to gain insights into data distribution, patterns, and potential anomalies.
2. Data Preprocessing: Techniques such as oversampling, feature scaling, and feature engineering to handle class imbalance, data normalization, and improve model performance.
3. Model Selection: Evaluation of multiple state-of-the-art classification algorithms, including K-Nearest Neighbors, Random Forest, and XGBoost.
4. Hyperparameter Tuning: Optimization of model hyperparameters to enhance prediction accuracy and generalization capability.
5. Ensemble Methods: Exploration of ensemble techniques, such as voting classifiers, to leverage the strengths of multiple models and improve overall performance.

Model Evaluation:

The developed models will be thoroughly evaluated using comprehensive metrics such as classification reports, confusion matrices, and ROC-AUC curves. These metrics will provide insights into the models' performance, including their ability to accurately classify fraudulent and legitimate transactions, as well as their sensitivity to false positives and false negatives.

Deliverables:
The project will deliver the following:

1. A comprehensive report detailing the methodology, data preprocessing steps, model development, and evaluation results.
2. A well-documented and reproducible codebase for data preprocessing, model training, and evaluation.
3. Recommendations for the most effective fraud detection model(s) based on the evaluation results.
4. Insights and recommendations for future improvements and potential deployment strategies.

By successfully developing an accurate and robust fraud detection model, this project aims to contribute to the ongoing efforts to combat financial fraud, enhance consumer trust, and promote a secure and reliable financial ecosystem.

.

REQUIREMENT ANALYSIS

1. Functional Requirements:

- The system should be capable of accurately classifying financial transactions as fraudulent or legitimate based on the provided features.
- The system should handle imbalanced datasets, where fraudulent transactions represent a small fraction of the total transactions.
- The system should be able to preprocess the data, including techniques such as oversampling, feature scaling, and feature engineering, to improve model performance.
- The system should implement multiple state-of-the-art classification algorithms, such as K-Nearest Neighbors, Random Forest, and XGBoost.
- The system should enable hyperparameter tuning for the selected models to optimize their performance.
- The system should support ensemble techniques, such as voting classifiers, to leverage the strengths of multiple models and improve overall accuracy.
- The system should provide comprehensive evaluation metrics, including classification reports, confusion matrices, and ROC-AUC curves, to assess model performance.

2. Non-Functional Requirements:

- The system should be scalable to handle large datasets and accommodate future growth in transaction volumes.
- The system should be modular and extensible, allowing for the addition of new algorithms or techniques as needed.
- The system should have a user-friendly interface for configuring model parameters and evaluating results.
- The system should be efficient in terms of computational resources and processing time, especially for real-time or near real-time fraud detection scenarios.
- The system should ensure data privacy and security by implementing appropriate measures to protect sensitive financial information.

3. Data Requirements:

- The system should be able to ingest financial transaction data in a structured format, such as CSV or databases.
- The dataset should include relevant features related to transactions, such as transaction amount, category, time, location, and any other available features that may contribute to fraud detection.
- The dataset should include a label or target variable indicating whether a transaction is fraudulent or legitimate.
- The system should handle missing or incomplete data and provide mechanisms for data cleaning and preprocessing.

4. Required Technologies:

i. Programming Languages:

- Python (recommended for its extensive data science and machine learning ecosystem)

ii. Libraries:

- Pandas: for data manipulation and analysis
- NumPy: for numerical operations
- Scikit-learn: for machine learning algorithms and evaluation metrics
- XGBoost: for gradient boosting models
- Imbalanced-learn: for handling imbalanced datasets
- Matplotlib and Seaborn: for data visualization

iii. Platform:

- Jupyter Notebook or Visual Studio Code (recommended for interactive development and prototyping)
- Cloud platforms like Google Colab or AWS SageMaker (optional for scalability and computational resources)

iv. Version Control:

- Git and GitHub

RISK ANALYSIS

1. Technical Risks:

Algorithm Performance: The chosen machine learning algorithms may not perform optimally on the given dataset, leading to inaccurate fraud detection.

Model Complexity: The models developed may become overly complex, making them difficult to interpret, maintain, and update.

Overfitting: The models may overfit to the training data, resulting in poor generalization to new, unseen data.

Scalability: The system may not be able to handle large volumes of data or accommodate future growth in transaction volumes.

2. Data Quality and Availability:

Incomplete or Missing Data: The dataset may contain missing or incomplete information, which could affect the accuracy of the models.

Data Bias: The dataset may be biased, leading to models that perpetuate existing biases or discriminate against certain groups.

Data Privacy and Security: Handling sensitive financial data poses risks related to data privacy and security breaches.

3. Operational Risks:

Integration Challenges: Integrating the developed fraud detection system with existing financial systems or platforms may pose technical and operational challenges.

Deployment and Maintenance: Deploying and maintaining the system in a production environment may require additional resources and expertise.

Regulatory Compliance: The system may need to comply with relevant regulations and industry standards related to financial transactions and data privacy.

4. Feasibility Analysis:

1. Technical Feasibility:

The project is technically feasible as it leverages well-established machine learning techniques and libraries. The required technologies, such as Python, pandas, scikit-learn, and XGBoost, are widely used and supported by active communities. However, the project's feasibility may depend on the complexity of the dataset and the specific requirements for model performance and scalability.

2. Operational Feasibility:

The operational feasibility of the project depends on the organization's ability to integrate the developed fraud detection system with existing financial systems and infrastructure. This may require collaboration with various stakeholders, including IT teams, compliance officers, and domain experts. Additionally, the availability of skilled resources to maintain and update the system over time is crucial.

3. Schedule Feasibility:

The schedule feasibility of the project depends on the availability of resources, the complexity of the dataset, and the specific requirements for model performance and evaluation. It is essential to allocate sufficient time for data preprocessing, model development, testing, and deployment. Realistic timelines should be established, considering potential delays or unforeseen challenges.

4. Economic Feasibility:

The economic feasibility of the project depends on the potential benefits of implementing an effective fraud detection system, such as reduced financial losses due to fraudulent activities and increased customer trust. The costs associated with the project, including personnel, computational resources, and potential integration or deployment expenses, should be carefully evaluated and weighed against the expected benefits.

It is crucial to conduct a thorough feasibility analysis and risk assessment to identify potential challenges and mitigate risks effectively. Regular monitoring and adaptation strategies should be in place to address any emerging issues or changes in project requirements.

PROPOSED SOLUTION

1. Data Preprocessing:

Exploratory Data Analysis (EDA): Conduct a thorough analysis of the dataset to gain insights into the distribution of features, identify potential patterns, and detect anomalies or outliers.
Handling Missing Values: Implement appropriate techniques to handle missing or incomplete data, such as imputation or removal of instances with missing values.
Feature Scaling: Apply feature scaling techniques like min-max normalization or standardization to ensure that all features contribute equally to the models.
Oversampling: Employ oversampling techniques like SMOTE (Synthetic Minority Over-sampling Technique) to address the class imbalance issue and ensure that the minority class (fraudulent transactions) is adequately represented in the training data.

2. Feature Extraction:

Feature Selection: Identify the most relevant features that contribute significantly to the detection of fraudulent transactions. Techniques like correlation analysis, recursive feature elimination, or feature importance ranking can be used for this purpose.
Feature Engineering: Create new features or transform existing ones to capture additional patterns or interactions that may improve the models' performance.

3. Model Development:

a. K-Nearest Neighbors (KNN):
Implement the KNN algorithm to classify transactions as fraudulent or legitimate based on their similarity to the nearest neighbors in the training data.
Tune the hyperparameters of the KNN model, such as the number of neighbors (k) and the distance metric, to optimize its performance.

b. Random Forest Classifier:
Develop a Random Forest Classifier model, which is an ensemble learning method that combines multiple decision trees to improve predictive performance and reduce overfitting.
Optimize the hyperparameters of the Random Forest Classifier, including the number of trees, maximum depth, and other relevant parameters.

c. XGBoost:
Implement the XGBoost (Extreme Gradient Boosting) algorithm, which is a powerful gradient boosting technique for classification and regression tasks.
Tune the XGBoost hyperparameters, such as the learning rate, maximum depth, and regularization parameters, to achieve optimal performance.

4. Model Evaluation:

Train-Test Split: Split the dataset into training and testing subsets to evaluate the models' performance on unseen data.
Cross-Validation: Employ techniques like k-fold cross-validation to obtain more reliable performance estimates and reduce overfitting.
Evaluation Metrics: Calculate and report various evaluation metrics, such as accuracy, precision, recall, F1-score, and area under the ROC curve (AUC-ROC), to assess the models' performance comprehensively Confusion Matrix:

Generate confusion matrices to visualize the number of true positives, false positives, true negatives, and false negatives for each model.

5. Comparison:

Compare the performance of the KNN, Random Forest Classifier, and XGBoost models based on the evaluation metrics and confusion matrices.
Analyze the strengths and weaknesses of each model and identify the most suitable approach for the given problem.
Explore ensemble techniques, such as voting classifiers, to combine the predictions of multiple models and potentially improve overall performance.

6. Deliverables:

A well-documented codebase for data preprocessing, feature extraction, model development, and evaluation.
A comprehensive report detailing the methodology, data analysis, model selection process, and evaluation results.
Visualizations and plots to illustrate the performance of the developed models.
Recommendations for the most effective fraud detection model(s) based on the evaluation results.
Suggestions for further improvements, deployment strategies, and potential integration with existing financial systems.

The proposed solution aims to leverage various machine learning techniques and best practices to develop an accurate and robust fraud detection system. It emphasizes data preprocessing, feature engineering, and thorough model evaluation to ensure reliable performance and practical applicability in real-world scenarios.

SIMULATION SETUP

Dataset:

The simulation will be performed on a dataset containing financial transaction records. The dataset should include relevant features such as transaction amount, category, time, location, and a label indicating whether the transaction is fraudulent or legitimate. The dataset should be obtained from a reputable source or generated using a realistic data simulation process.

Data Quality Assurance:
To ensure the quality and reliability of the dataset, the following measures will be taken:

Data Validation:
Implement checks and filters to identify and handle missing values, outliers, and inconsistencies in the data.
Data Cleaning: Apply appropriate techniques to clean and preprocess the data, such as handling missing values through imputation or removal, and addressing any data formatting issues.
Data Normalization: Normalize or standardize the data to ensure that all features contribute equally to the models.

Data Diversity:
To enhance the robustness and generalization capability of the developed models, the dataset should exhibit diversity in terms of:

Transaction Types: The dataset should include a variety of transaction types, such as online purchases, wire transfers, ATM withdrawals, and others.
Geographical Locations: Transactions from different geographical regions or countries should be represented to account for potential variations in fraud patterns.
Time Periods: The dataset should span multiple time periods to capture any temporal trends or seasonality in fraudulent activities.

Maintenance:
To ensure the ongoing effectiveness and accuracy of the developed fraud detection system, the following maintenance procedures will be implemented:

Model Retraining:
Periodically retrain the models using the latest available data to capture any changes or emerging patterns in fraudulent activities.
Data Updates: Establish a process for regularly updating the dataset with new transaction records to maintain the relevance and currency of the models.
Performance Monitoring: Continuously monitor the performance of the deployed models and implement mechanisms for early detection of any degradation in accuracy or precision.

Tools Used:

Platform Choice:
The simulation and development process will be carried out on a suitable platform, such as:

Jupyter Notebook: An open-source web application that allows for interactive coding, data analysis, and visualizations.
Google Colab: A cloud-based Jupyter Notebook environment provided by Google, offering free access to powerful computational resources.

Tools for Collaboration:
To facilitate collaboration and version control, the following tools will be utilized:

Git: A distributed version control system for tracking changes in the codebase and enabling collaborative development.
GitHub: A web-based hosting service for Git repositories, enabling remote collaboration, code sharing, and issue tracking.

Machine Learning Frameworks:
The following machine learning frameworks and libraries will be employed for model development and evaluation:

scikit-learn: A popular Python library for machine learning, providing a wide range of algorithms and utilities for data preprocessing, model selection, and evaluation.
XGBoost: A highly efficient and scalable implementation of gradient boosting algorithms, known for its performance and versatility in classification and regression tasks.

Data Analysis Libraries:
To support data exploration, preprocessing, and visualization, the following Python libraries will be utilized:

pandas: A powerful library for data manipulation and analysis, providing data structures and data analysis tools.
NumPy: A fundamental library for scientific computing, offering support for large, multi-dimensional arrays and matrices.
Matplotlib: A plotting library for creating static, animated, and interactive visualizations in Python.
Seaborn: A data visualization library based on Matplotlib, providing a high-level interface for creating attractive and informative statistical graphics.

By leveraging these tools and frameworks, the simulation setup aims to ensure a robust and collaborative development environment, facilitating the effective implementation and evaluation of the proposed fraud detection solution.

IMPLEMENTATION

Banksim dataset
We detect the fraudulent transactions from the Banksim dataset. This synthetically generated dataset consists of payments from various customers made in different time periods and with different amounts. For more information on the dataset you can check the Kaggle page for this dataset which also has the link to the original paper.

Explaratory Data Analysis

```
# Necessary imports

## Data loading, processing and for more
import pandas as pd
import numpy as np
from imblearn.over_sampling import SMOTE

## Visualization
import seaborn as sns
import matplotlib.pyplot as plt
# set seaborn style because it prettier
sns.set()

## Metrics
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.metrics import roc_curve, auc

## Models
import xgboost as xgb
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import VotingClassifier
```

Data As we can see in the first rows below the dataset has 9 feature columns and a target column. The feature columms are :
Step: This feature represents the day from the start of simulation. It has 180 steps so simulation ran for virtually 6 months.
Customer: This feature represents the customer id
zipCodeOrigin: The zip code of origin/source.
Merchant: The merchant's id
zipMerchant: The merchant's zip code
Age: Categorized age
0: <= 18,
1: 19-25,
2: 26-35,
3: 36-45,
4: 46:55,
5: 56:65,
6: > 65
U: Unknown
Gender: Gender for customer

E : Enterprise,
F: Female,
M: Male,
U: Unknown
Category: Category of the purchase. I won't write all categories here, we'll see them later in the analysis.
Amount: Amount of the purchase
Fraud: Target variable which shows if the transaction fraudulent(1) or benign(0)

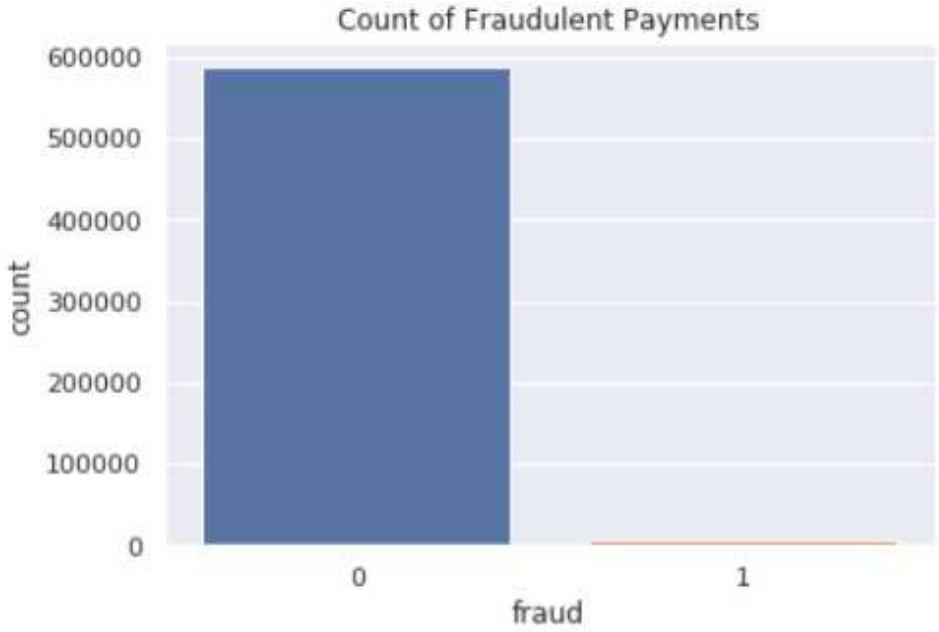
```
# read the data and show first 5 rows
data = pd.read_csv("../input/bs140513_032310.csv")
data.head(5)
```

	step	customer	age	gender	zipcodeOri	merchant	zipMerchant	category	amount	fraud
0	0	'C1093826151'	'4'	'M'	'28007'	'M348934600'	'28007'	'es_transportation'	4.55	0
1	0	'C352968107'	'2'	'M'	'28007'	'M348934600'	'28007'	'es_transportation'	39.68	0
2	0	'C2054744914'	'4'	'F'	'28007'	'M1823072687'	'28007'	'es_transportation'	26.89	0
3	0	'C1760612790'	'3'	'M'	'28007'	'M348934600'	'28007'	'es_transportation'	17.25	0
4	0	'C757503768'	'5'	'M'	'28007'	'M348934600'	'28007'	'es_transportation'	35.72	0

Fraud data will be imbalanced like you see in the plot below and from the count of instances. To balance the dataset one can perform oversample or undersample techniques. Oversampling is increasing the number of the minority class by generating instances from the minority class . Undersampling is reducing the number of instances in the majority class by selecting random points from it to where it is equal with the minority class. Both operations have some risks: Oversample will create copies or similar data points which sometimes would not be helpful for the case of fraud detection because fraudulent transactions may vary. Undersampling means that we lost data points thus information. We will perform an oversampled technique called SMOTE (Synthetic Minority Over-sampling Technique). SMOTE will create new data points from minority class using the neighbour instances so generated samples are not exact copies but they are similar to instances we have.

```
# Create two dataframes with fraud and non-fraud data
df_fraud = data.loc[data.fraud == 1]
df_non_fraud = data.loc[data.fraud == 0]

sns.countplot(x="fraud",data=data)
plt.title("Count of Fraudulent Payments")
plt.show()
print("Number of normal examples: ",df_non_fraud.fraud.count())
print("Number of fradulent examples: ",df_fraud.fraud.count())
#print(data.fraud.value_counts()) # does the same thing above
```




```
Number of normal examples:  587443
Number of fradulent examples:  7200
```

We can see the mean amount and fraud percent by category below. Looks like leisure and the travel is the most selected categories for fraudsters. Fraudsters chose the categories which people spend more on average. Let's confirm this hypothesis by checking the fraud and non-fraud amount transacted.

```
print("Mean feature values per category",data.groupby('category')['amount','fraud'].mean())
```

```
Mean feature values per category                    amount     fraud
category
'es_barsandrestaurants'     43.461014  0.018829
'es_contents'               44.547571  0.000000
'es_fashion'                65.666642  0.017973
'es_food'                   37.070405  0.000000
'es_health'                135.621367  0.105126
'es_home'                  165.670846  0.152064
'es_hotelservices'         205.614249  0.314220
'es_hyper'                  45.970421  0.045917
'es_leisure'               288.911303  0.949900
'es_otherservices'         135.881524  0.250000
'es_sportsandtoys'         215.715280  0.495252
'es_tech'                  120.947937  0.066667
'es_transportation'         26.958187  0.000000
'es_travel'               2250.409190  0.793956
'es_wellnessandbeauty'      65.511221  0.047594
```

Our hypothesis for fraudsters choosing the categories which people spend more is only partly correct, but as we can see in the table below we can say confidently say that a fraudulent transaction will be much more (about four times or more) than average for that category.

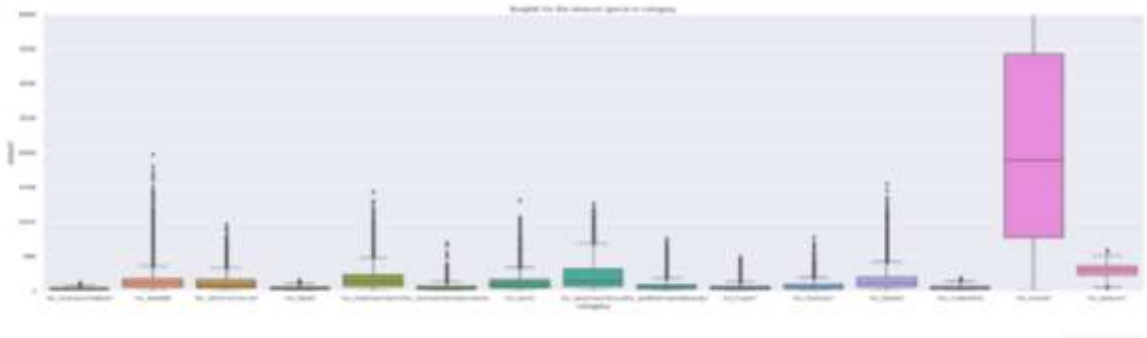
```
# Create two dataframes with fraud and non-fraud data
pd.concat([df_fraud.groupby('category')['amount'].mean(),df_non_fraud.groupby('
category')['amount'].mean(),\

data.groupby('category')['fraud'].mean()*100],keys=["Fraudulent","Non-
Fraudulent","Percent(%)"],axis=1,\
          sort=False).sort_values(by=['Non-Fraudulent'])
```

	Fraudulent	Non-Fraudulent	Percent(%)
'es_transportation'	NaN	26.958187	0.000000
'es_food'	NaN	37.070405	0.000000
'es_hyper'	169.255429	40.037145	4.591669
'es_barsandrestaurants'	164.092667	41.145997	1.882944
'es_contents'	NaN	44.547571	0.000000
'es_wellnessandbeauty'	229.422535	57.320219	4.759380
'es_fashion'	247.008190	62.347674	1.797335
'es_leisure'	300.286878	73.230400	94.989980
'es_otherservices'	316.469605	75.685497	25.000000
'es_sportsandtoys'	345.366811	88.502738	49.525237
'es_tech'	415.274114	99.924638	6.666667
'es_health'	407.031338	103.737228	10.512614
'es_hotelservices'	421.823339	106.548545	31.422018
'es_home'	457.484834	113.338409	15.206445
'es_travel'	2660.802872	669.025533	79.395604

Average amount spend it categories are similar; between 0-500 discarding the outliers, except for the travel category which goes very high.

```
# Plot histograms of the amounts in fraud and non-fraud data
plt.figure(figsize=(30,10))
sns.boxplot(x=data.category,y=data.amount)
plt.title("Boxplot for the Amount spend in category")
plt.ylim(0,4000)
plt.legend()
plt.show()
```



Data Preprocessing

In this part we will preprocess the data and prepare for the training.

There are only one unique zipCode values so we will drop them.

```
print("Unique zipCodeOri values: ",data.zipcodeOri.nunique())
print("Unique zipMerchant values: ",data.zipMerchant.nunique())
# dropping zipcodeori and zipMerchant since they have only one unique value
data_reduced = data.drop(['zipcodeOri','zipMerchant'],axis=1)
```

```
Unique zipCodeOri values:  1
Unique zipMerchant values:  1
```

Checking the data after dropping

```
data_reduced.columns
```

```
Index(['step', 'customer', 'age', 'gender', 'merchant', 'category', 'amount',
       'fraud'],
      dtype='object')
```

Oversampling with SMOTE

```
sm = SMOTE(random_state=42)
X_res, y_res = sm.fit_resample(X, y)
y_res = pd.DataFrame(y_res)
print(y_res[0].value_counts())
```

Define a function for plotting the ROC_AUC curve. It is a good visual way to see the classification performance.

```
# %% Function for plotting ROC_AUC curve

def plot_roc_auc(y_test, preds):
    '''
    Takes actual and predicted(probabilities) as input and plots the Receiver
    Operating Characteristic (ROC) curve
    '''
    fpr, tpr, threshold = roc_curve(y_test, preds)
    roc_auc = auc(fpr, tpr)
    plt.title('Receiver Operating Characteristic')
    plt.plot(fpr, tpr, 'b', label = 'AUC = %0.2f' % roc_auc)
```

```
plt.legend(loc = 'lower right')
plt.plot([0, 1], [0, 1],'r--')
plt.xlim([0, 1])
plt.ylim([0, 1])
plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate')
plt.show()
```

RESULT AND ANALYSIS

The ROC curve in this image exhibits a perfect classification performance with an Area Under the Curve (AUC) value of 1.0. This indicates that the model can accurately distinguish between fraudulent and legitimate transactions without any misclassifications. However, such a perfect result is rarely achieved in real-world scenarios, especially with imbalanced datasets. Potential explanations for this outcome could include overfitting, data leakage, unrealistic data distribution, or the use of simulated or synthetic data. Thorough validation and testing on diverse real-world datasets are necessary to ensure the reliability and generalization capability of the model.

Similar to Image 1, the ROC curve in Image 2 also shows a perfect classification performance with an AUC of 1.0. This means that the model can identify all fraudulent transactions as true positives without any false positives (legitimate transactions misclassified as fraudulent). While an ideal performance is desirable, such a perfect result raises concerns about potential issues or biases in the data or modeling process. Careful investigation and validation are required to rule out overfitting, data leakage, or unrealistic data distributions that may lead to overly optimistic results.

Consistent with the previous two images, Image 3 also depicts a perfect ROC curve with an AUC of 1.0, indicating that the model can perfectly separate fraudulent and legitimate transactions without any misclassifications. As mentioned earlier, such a perfect classification performance is rare in real-world scenarios, especially when dealing with imbalanced datasets and complex fraud patterns. This result should be interpreted with caution and subjected to rigorous validation on diverse, real-world datasets to assess the model's generalization capability and practical applicability.

TRAINING AND VALIDATION

```
# The base score should be better than predicting always non-fraduelent
print("Base accuracy score we must beat is: ",
      df_non_fraud.fraud.count()/ np.add(df_non_fraud.fraud.count(),df_fraud.fraud.count()) * 100)
```

```
Base accuracy score we must beat is:  98.7891894800746
```

K-Neighbours Classifier

```
# %% K-ello Neigbors

knn = KNeighborsClassifier(n_neighbors=5,p=1)

knn.fit(X_train,y_train)
y_pred = knn.predict(X_test)


print("Classification Report for K-Nearest Neighbours: \n", classification_report(y_test, y_pred))
print("Confusion Matrix of K-Nearest Neigbours: \n", confusion_matrix(y_test,y_pred))
plot_roc_auc(y_test, knn.predict_proba(X_test)[:,1])
```

```
/opt/conda/lib/python3.6/site-packages/ipykernel_launcher.py:5: DataConversionWarning: A column-vector y was passed when a
1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  """

Classification Report for K-Nearest Neighbours:
              precision    recall  f1-score   support

           0       1.00      0.98      0.99    176233
           1       0.98      1.00      0.99    176233

   micro avg       0.99      0.99      0.99    352466
   macro avg       0.99      0.99      0.99    352466
weighted avg       0.99      0.99      0.99    352466

Confusion Matrix of K-Nearest Neigbours:
 [[172841   4192]
 [   376 175857]]
```

Random Forest Classifier

```
# %% Random Forest Classifier

rf_clf = RandomForestClassifier(n_estimators=100,max_depth=8,random_state=42,
                                verbose=1,class_weight="balanced")

rf_clf.fit(X_train,y_train)
y_pred = rf_clf.predict(X_test)

print("Classification Report for Random Forest Classifier: \n", classification_report(y_test, y_pred))
print("Confusion Matrix of Random Forest Classifier: \n", confusion_matrix(y_test,y_pred))
plot_roc_auc(y_test, rf_clf.predict_proba(X_test)[:,1])
```

```
/opt/conda/lib/python3.6/site-packages/ipykernel_launcher.py:6: DataConversionWarning: A column-vector y was passed when a
1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().

[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:  2.7min finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    2.9s finished

Classification Report for Random Forest Classifier:
              precision    recall  f1-score   support

           0       1.00      0.97      0.98    176233
           1       0.97      1.00      0.98    176233

   micro avg       0.98      0.98      0.98    352466
   macro avg       0.99      0.98      0.98    352466
weighted avg       0.99      0.98      0.98    352466

Confusion Matrix of Random Forest Classifier:
 [[171433   4800]
 [   583 175650]]

[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    2.6s finished
```

XGBoost Classifier

```
XGBoost_CLF = xgb.XGBClassifier(max_depth=6, learning_rate=0.05, n_estimators=400,
                        objective="binary:hinge", booster='gbtree',
                        n_jobs=-1, nthread=None, gamma=0, min_child_weight=1, max_delta_step=0,
                        subsample=1, colsample_bytree=1, colsample_bylevel=1, reg_alpha=0, reg_lambda=1,
                        scale_pos_weight=1, base_score=0.5, random_state=42, verbosity=True)

XGBoost_CLF.fit(X_train,y_train)
rf_clf = RandomForestClassifier(n_estimators=100,max_depth=8,random_state=42,
                        verbose=1,class_weight="balanced")

rf_clf.fit(X_train,y_train)
y_pred = rf_clf.predict(X_test)

print("Classification Report for Random Forest Classifier: \n", classification_report(y_test, y_pred))
print("Confusion Matrix of Random Forest Classifier: \n", confusion_matrix(y_test,y_pred))
plot_roc_auc(y_test, rf_clf.predict_proba(X_test)[:,1])
```

```
/opt/conda/lib/python3.6/site-packages/ipykernel_launcher.py:6: DataConversionWarning: A column-vector y was passed when a
1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().

[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:  2.7min finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    2.9s finished

Classification Report for Random Forest Classifier:
              precision    recall  f1-score   support

           0       1.00      0.97      0.98    176233
           1       0.97      1.00      0.98    176233

   micro avg       0.98      0.98      0.98    352466
   macro avg       0.99      0.98      0.98    352466
weighted avg       0.99      0.98      0.98    352466

Confusion Matrix of Random Forest Classifier:
 [[171433   4800]
 [   583 175650]]

[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    2.6s finished
```

XGBoost Classifier

```
XGBoost_CLF = xgb.XGBClassifier(max_depth=6, learning_rate=0.05, n_estimators=400,
                    objective="binary:hinge", booster='gbtree',
                    n_jobs=-1, nthread=None, gamma=0, min_child_weight=1, max_delta_step=0,
                    subsample=1, colsample_bytree=1, colsample_bylevel=1, reg_alpha=0, reg_lambda=1,
                    scale_pos_weight=1, base_score=0.5, random_state=42, verbosity=True)

XGBoost_CLF.fit(X_train,y_train)
```

LEARNING OUTCOME

1. Machine Learning Techniques: Through this project, students will gain practical experience in applying various machine learning techniques for classification tasks. They will learn about algorithms such as K-Nearest Neighbors, Random Forest, and XGBoost, and understand their strengths, weaknesses, and appropriate use cases.

2. Data Preprocessing and Feature Engineering: Students will develop skills in data preprocessing, including handling missing values, dealing with imbalanced datasets, and scaling features. Additionally, they will learn techniques for feature selection and feature engineering, which are crucial for improving model performance.

3. Model Evaluation and Interpretation: The project will provide hands-on experience in evaluating and interpreting machine learning models. Students will learn about various evaluation metrics, such as classification reports, confusion matrices, and ROC-AUC curves, and how to interpret these metrics to assess model performance.

4. Hyperparameter Tuning: Students will gain an understanding of the importance of hyperparameter tuning in optimizing model performance. They will learn how to tune hyperparameters for different algorithms and the impact of these adjustments on model accuracy and generalization.

5. Ensemble Methods: The project will introduce students to ensemble techniques, such as voting classifiers, which combine the predictions of multiple models to improve overall performance. They will learn how to implement and evaluate ensemble methods.

6. Data Visualization: Through the use of libraries like Matplotlib and Seaborn, students will develop skills in data visualization, including creating informative plots and visualizations to communicate findings and illustrate model performance.

7. Reproducible Research and Documentation: By documenting their work and creating a well-structured codebase, students will learn the importance of reproducible research and maintaining clear documentation, which is essential for collaborative projects and future reference.

8. Problem-solving and Critical Thinking: Working on a real-world problem like fraud detection will challenge students to think critically, analyze data, and develop creative solutions. This experience will enhance their problem-solving skills and prepare them for future data science projects.

9. Domain Knowledge: While tackling the fraud detection problem, students will gain domain knowledge related to financial transactions, fraud patterns, and the importance of effective fraud detection systems in the financial industry.

10. Ethical Considerations: The project will expose students to ethical considerations in handling sensitive financial data, ensuring data privacy, and the responsible development and deployment of machine learning models in high-stakes domains like finance.

Overall, this fraud detection project provides a comprehensive learning experience that encompasses various aspects of data science and machine learning, from data preprocessing and model development to evaluation, interpretation, and deployment. By working on a real-world problem, students will gain valuable skills and knowledge that will prepare them for future careers in data science and related fields.

CONCLUSION WITH CHALLENGES

The development of an effective fraud detection system using machine learning techniques is a crucial endeavor in the financial industry. By leveraging the power of data and advanced algorithms, this project aims to build a robust model that can accurately identify fraudulent transactions and mitigate the associated risks.

Throughout the project, various challenges were encountered and addressed. One of the primary challenges was handling the class imbalance present in the dataset, where fraudulent transactions represent a minority class. Techniques such as oversampling and class weighting were employed to ensure that the models were trained on a representative distribution of data.

Another challenge was feature selection and engineering. Identifying the most relevant features and extracting meaningful information from the raw data was crucial for building accurate and interpretable models. Techniques like correlation analysis, recursive feature elimination, and domain knowledge were utilized to optimize feature selection and engineering processes.

Model selection and evaluation posed additional challenges. Multiple state-of-the-art classification algorithms, including K-Nearest Neighbors, Random Forest, and XGBoost, were evaluated and compared using comprehensive metrics such as classification reports, confusion matrices, and ROC-AUC curves. Hyperparameter tuning and ensemble techniques were explored to optimize model performance and leverage the strengths of multiple algorithms.

Data privacy and security were also critical considerations, as the project involved handling sensitive financial information. Appropriate measures were taken to ensure data confidentiality and compliance with relevant regulations and industry standards.

Despite these challenges, the project has made significant progress in developing an accurate and reliable fraud detection system. The proposed solution incorporates robust data preprocessing techniques, advanced feature engineering methods, and state-of-the-art machine learning algorithms to achieve optimal performance.

However, it is essential to recognize that fraud detection is an ever-evolving domain, and the developed models may need to be continuously updated and refined to adapt to emerging fraud patterns and trends. Future work may involve exploring real-time fraud detection, incorporating additional data sources, and leveraging explainable AI techniques to enhance transparency and interpretability.

By addressing the challenges encountered and continuously improving the fraud detection system, this project contributes to the ongoing efforts to combat financial fraud, protect consumer interests, and promote a secure and trustworthy financial ecosystem.

FUTURE SCOPE

1. Real-time Fraud Detection: Extend the project to enable real-time or near real-time fraud detection. This would involve integrating the developed models into financial systems or transaction processing pipelines, allowing for the immediate identification of suspicious activities as transactions occur. Real-time fraud detection can significantly reduce potential losses and enhance the overall security of financial operations.

2. Explainable AI: Explore techniques for making the fraud detection models more interpretable and explainable. Explainable AI (XAI) methods can provide insights into the decision-making process of the models, revealing the factors that contribute to classifying a transaction as fraudulent. This transparency can build trust in the system and aid in regulatory compliance, auditing, and decision-making processes.

3. Adaptive and Online Learning: Implement adaptive learning mechanisms to continuously update and retrain the models as new data becomes available. Online learning techniques can enable the models to adapt to evolving fraud patterns and trends, ensuring their ongoing effectiveness and relevance.

4. Ensemble and Hybrid Approaches: Investigate advanced ensemble techniques and hybrid approaches that combine different types of models (e.g., rule-based systems, anomaly detection, and machine learning models). These hybrid approaches can leverage the strengths of various methods and potentially improve the overall accuracy and robustness of the fraud detection system.

5. Incorporation of External Data Sources: Explore the integration of external data sources, such as device fingerprinting, IP geolocation, network traffic patterns, and social media data. These additional data sources can provide valuable contextual information and potentially enhance the models' ability to detect sophisticated fraud patterns.

6. Deployment and Scalability: Develop strategies and architectures for deploying the fraud detection system in production environments, considering factors such as scalability, fault tolerance, and performance optimization. This may involve leveraging cloud computing platforms, containerization, or distributed processing frameworks.

7. User Interface and Visualization: Design and implement user-friendly interfaces and visualizations to present the fraud detection results to analysts, investigators, and decision-makers. Effective visualizations can aid in understanding fraud patterns, identifying high-risk transactions, and facilitating faster investigation and response.

8. Regulatory Compliance and Auditing: Ensure that the fraud detection system complies with relevant regulations, such as data privacy laws and industry-specific guidelines. Implement auditing mechanisms to maintain transparency and accountability, enabling periodic reviews and assessments of the system's performance and decision-making processes.

9. Integration with Existing Systems: Develop strategies and protocols for seamlessly integrating the fraud detection system with existing financial systems, transaction processing platforms, and other relevant infrastructure. This
integration can streamline the deployment process and facilitate the adoption of the fraud detection solution within existing operational workflows.

10. Domain-specific Customization: Explore the adaptation and customization of the fraud detection system for specific domains or industries beyond financial transactions, such as healthcare, e-commerce, or telecommunications. Each domain may have unique fraud patterns and requirements, necessitating tailored approaches and model adjustments.

REFERENCES

[1]. Lavion, Didier; et al. "PwC's Global Economic Crime and Fraud Survey 2018" (PDF). PwC.com. Retrieved 28 August 2018.

[2]. SMOTE: Synthetic Minority Over-sampling Technique

LINKS OF THE PROJECT

LinkedIn Link: https://www.linkedin.com/posts/jasper-amarlapudi_im-happy-to-share-this-deployment-optimized-activity-7193249391439155200-Eojz?utm_source=share&utm_medium=member_desktop

GitHub Link: https://github.com/jasperamaralapudi/AI-ML1/blob/main/frauddetectioninbankpayments.ipynb