**TEAM 11**

**FINAL PROJECT**

**TOPIC:**

**Credit Card Transactions Fraud Detection**

**MEMBERS:**

**Christina Albert Daniel- NUID 002833358**

**Satya Siva Sai Mrudula Chilukuri- NUID 002644914**

Appendix

Introduction………………………………………………………………………………....3

Description of Dataset………………………………………………….…………………...4

Methodology…………………………………………………………...…………………...5

Data cleaning and Pre-processing……………………………………...…………………...6

Exploratory Data Analysis……………………………………………..…………………...9

Feature Engineering………………………………………………………………………..14

Model Selection and Evaluation…..……………………………………………………….16

Result and analysis…………………………………………………………………………21

Conclusion……………………………………………………………..…………………...22

Summary and Future Work…………………………………………………………………23

References……………………………………………………………..……………………23

**Introduction:**

**Background**

In today's increasingly digital financial landscape, the shift towards online transactions has unfortunately been accompanied by a surge in fraudulent activities. The dataset under consideration captures a detailed record of credit card transactions over the period of 2019 to 2020, documenting both legitimate and suspicious activities across 1000 customers and 800 merchants. It offers a broad spectrum of data, including the timing of transactions, transaction values, types of merchants, location data, and fraud status. This dataset serves as a fertile ground for analysing transactional patterns, identifying fraudulent behaviours, and understanding interactions between customers and merchants, making it a valuable resource for studying credit card fraud dynamics.

**Motivation**

The impetus behind choosing this dataset stems from the critical necessity to enhance fraud detection mechanisms in response to the sophisticated strategies employed by fraudsters, which cost the global economy significantly each year. By applying machine learning (ML) techniques to this dataset, there is a potential to improve the detection of fraudulent transactions proactively. This dataset not only provides a comprehensive platform for applying ML techniques to a real-world problem but also facilitates a deep dive into the complexities of securing financial transactions. The analysis will shed light on the hurdles of dealing with skewed datasets, underscore the role of feature selection, and highlight the deployment of various ML models in safeguarding financial transactions against fraud.

**Goal**

The objective of working with this dataset is to develop, test, and refine a series of machine learning models aimed at accurately and efficiently identifying fraudulent transactions. The project will employ a variety of classification strategies, including but not limited to logistic regression and neural networks, to discern patterns indicative of fraud. The ultimate ambition of this project is to enrich academic discussions on machine learning applications in fraud detection and offer actionable insights that could be implemented in financial infrastructures to bolster security measures and foster a safer transaction environment in future.

**Description of the Dataset**

This is a simulated credit card transaction dataset containing legitimate and fraud transactions from the duration 1st Jan 2019 - 31st Dec 2020. It covers credit cards of 1000 customers doing transactions with a pool of 800 merchants.

Here's a summary of columns present in the dataset:

1. index - Unique Identifier for each row
2. trans\_date\_trans\_time - Transaction DateTime
3. cc\_num - Credit Card Number of Customer
4. merchant - Merchant Name
5. category - Category of Merchant
6. amt - Amount of Transaction
7. first - First Name of Credit Card Holder
8. last - Last Name of Credit Card Holder
9. gender - Gender of Credit Card Holder
10. street - Street Address of Credit Card Holder
11. city - City of Credit Card Holder
12. state - State of Credit Card Holder
13. zip - Zip of Credit Card Holder
14. lat - Latitude Location of Credit Card Holder
15. long - Longitude Location of Credit Card Holder
16. city\_pop - Credit Card Holder's City Population
17. job - Job of Credit Card Holder
18. dob - Date of Birth of Credit Card Holder
19. trans\_num - Transaction Number
20. unix\_time - UNIX Time of transaction
21. merch\_lat - Latitude Location of Merchant
22. merch\_long - Longitude Location of Merchant
23. is\_fraud - Fraud Flag <--- Target Class

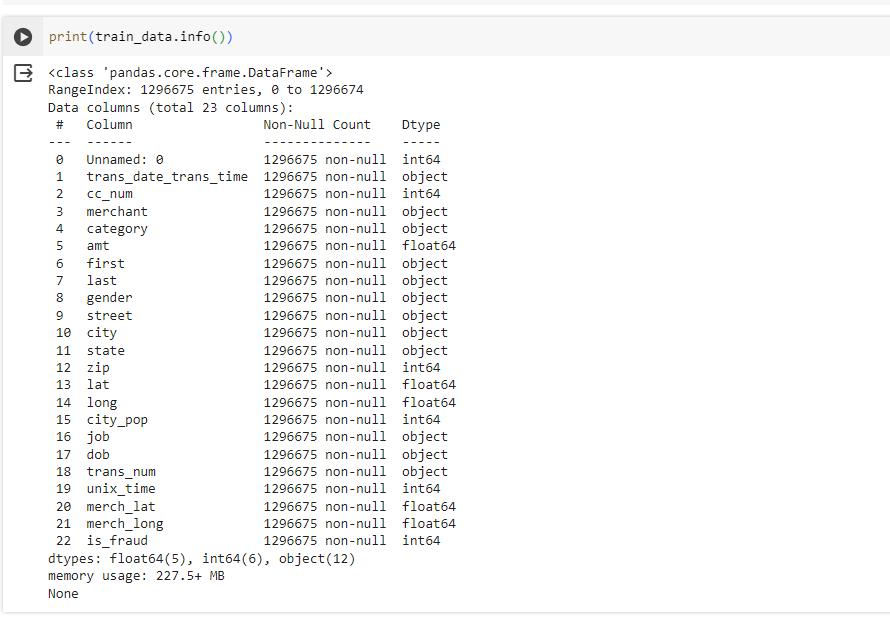
**Methodology**

1. **Data Processing cleaning:** We will first clean and preprocess the Fraud Detection dataset to ensure that the data is accurate and usable. This will involve removing any irrelevant or duplicate data, filling in missing values, standardizing data types and formats, checking for outliers or anomalies and ensuring that the data is in a format that can be used for analysis.
2. **Exploratory data analysis:** We will employ exploratory data analysis techniques including histograms, scatter plots, and correlation matrices to uncover patterns and insights within the data, specifically focusing on characteristics that may indicate fraudulent activities. Our analysis will involve visualizing key features such as transaction amounts, locations, transaction times, and demographic details of the cardholder. We aim to identify any correlations or trends between these variables and instances of fraud. The comprehensive EDA will guide our subsequent modeling efforts by highlighting which features are most relevant for detecting fraudulent transactions.
3. **Feature Engineering:** Leveraging our understanding of the domain, we will carefully select features and possibly create new variables from the existing dataset to enhance the predictive power of our fraud detection models. Our focus will be on identifying key predictors of fraudulent activity through both domain knowledge and statistical analysis. We anticipate that transaction amount ('amt'), merchant details ('merchant'), transaction location, transaction time, and cardholder's demographic information will be critical in our feature selection process. These efforts aim to uncover complex patterns and relationships within the data that simple analysis might miss, thereby improving the accuracy and effectiveness of our fraud detection efforts.
4. **Model Selection and Evaluation:** We will use Linear Classifier; A simple implementation to perform simple feature engineering before passing it to the better models of the application. Decision Tree/ Random Forest to use a recursive algorithm for better classification of values especially when the label counts might be unbalanced. For dimensionality separation Support Vector Classifier can be implemented using hyperplanes to offer better classification. Relying on Gaussian Naïve Bayes method to handle unbalanced datasets for better classification.
5. **Time Series Analysis:** We will be using time series analysis methods like ARIMA to detect patterns and trends and using the analysis of it to make predictions for future fraud trends. By correlating time series insights with other predictive indicators identified through exploratory data analysis and feature engineering, we can develop a comprehensive and robust system for fraud detection and prevention, tailored to adapt to and predict evolving fraud patterns.

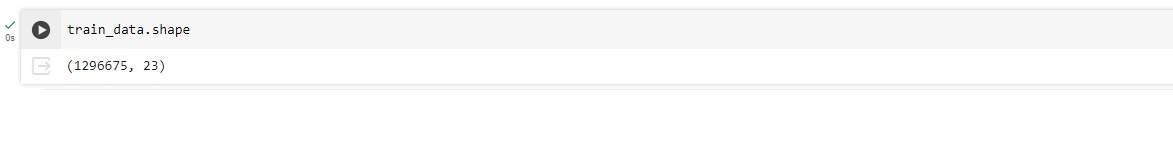
**Data Processing cleaning**



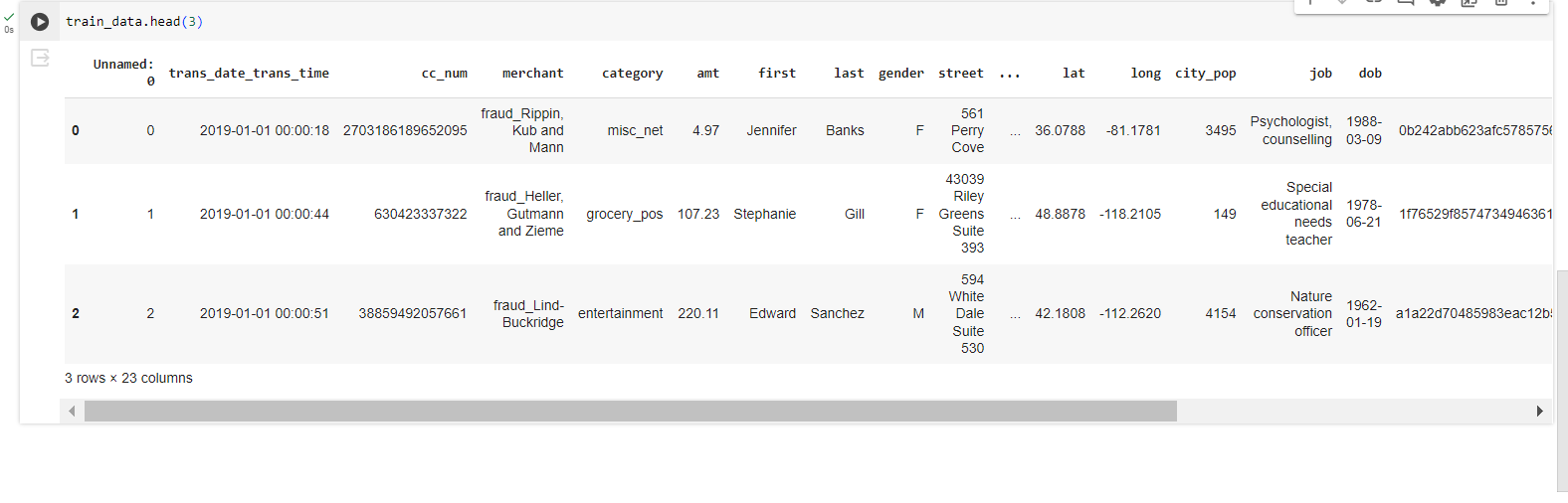
The screenshot depicts a segment of Python code utilized in data preprocessing, specifically the phase where datasets are imported for cleaning. In this instance, CSV files containing training and testing data for a fraud detection project are accessed from a specified directory within Google Drive, and are loaded into the Python environment using Pandas, a data analysis library. The DataFrames **train\_data** and **test\_data** created from these CSV files serve as the starting point for data cleaning, a critical step that involves rectifying inconsistencies and handling missing or anomalous data, thus ensuring the datasets' integrity before they undergo further analysis or modeling.



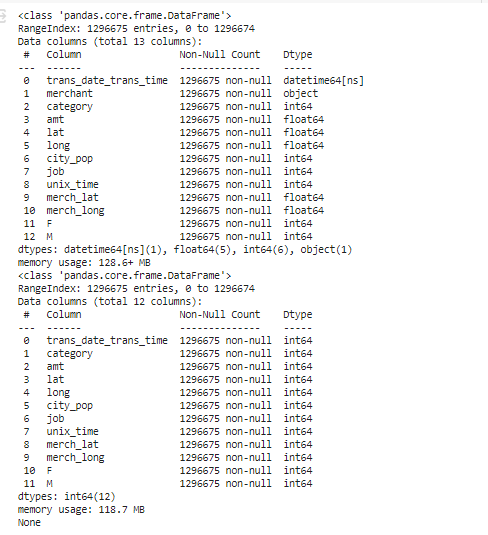
Following the initial data loading step, We conducted a thorough examination of the dataset's structure using the **info()** method on the **train\_data** DataFrame. This exploratory procedure provided me with a comprehensive overview of the dataset's composition, confirming the absence of missing values across all 1,296,675 records and 23 features. The dataset is a mix of numerical and categorical data, indicated by data types such as **int64**, **float64**, and **object**. The feature 'is\_fraud' caught my attention as it is critical for the predictive modeling of fraudulent transactions. Additionally, We noticed an 'Unnamed: 0' column, which suggests a redundant index that may require removal to streamline the dataset. The memory footprint of approximately 227.5 MB indicates a substantial dataset size, prompting me to consider efficient data handling strategies in subsequent processing steps. This preliminary analysis was crucial in planning the next stages of data cleaning and feature engineering to prepare the data for effective fraud detection analysis.



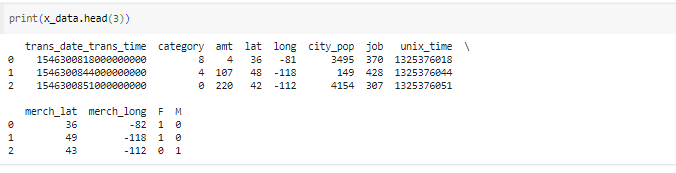
We verified the dimensions of our training dataset using the **shape** attribute, which confirmed that we have 1,296,675 transactions spread across 23 attributes to analyze for our fraud detection project.



We performed an initial data inspection on our training set by displaying the first three entries with the **head()** method, which provided a snapshot of various attributes, including transaction details, customer demographics, and job information. This preview was essential for familiarizing ourselves with the data format and content, setting the stage for deeper data exploration and subsequent preprocessing tasks.

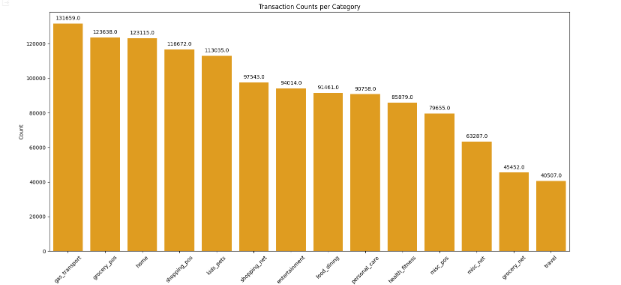


We progressed in our data preprocessing by transforming categorical variables using one-hot encoding to prepare them for machine learning algorithms. This transformation was carried out using Pandas' **get\_dummies()** function, making the data more suitable for analysis by converting categorical data into a numerical format. We further ensured the dataset's integrity by checking for null values before and after encoding, confirming the dataset was ready for the next steps in our model-building process.

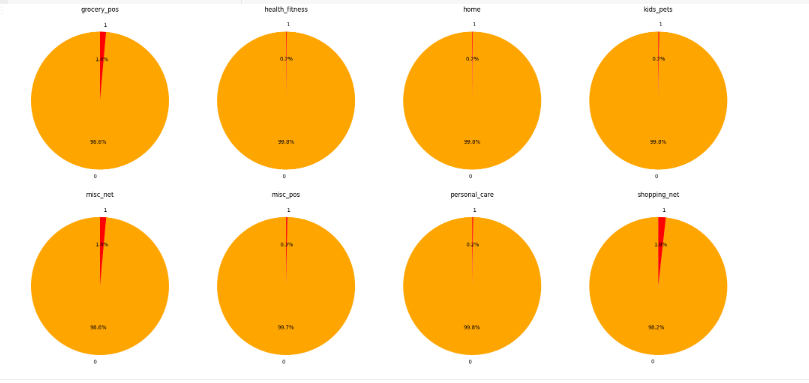


**Exploratory Data Analysis**Exploratory Data Analysis (EDA) is a crucial step in the data analysis process, involving the use of statistical techniques and visualizations to understand and summarize the main characteristics of a dataset, often with less formal structure and assumptions about the data.

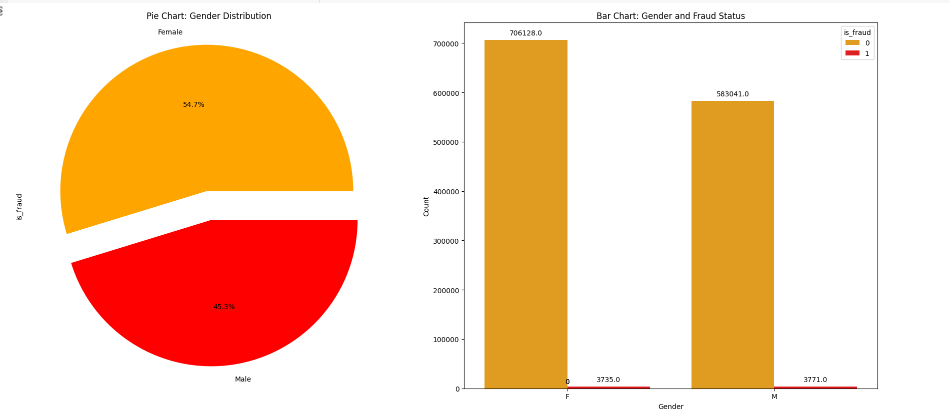
EDA aims to uncover underlying patterns, spot anomalies, test hypotheses, and check assumptions through descriptive statistics and graphical representations. It provides a means to discover insights, frame questions, and determine the most appropriate statistical techniques for further analysis. EDA is an iterative and interactive process, guiding analysts in making informed decisions about the subsequent steps in their data analysis projects.



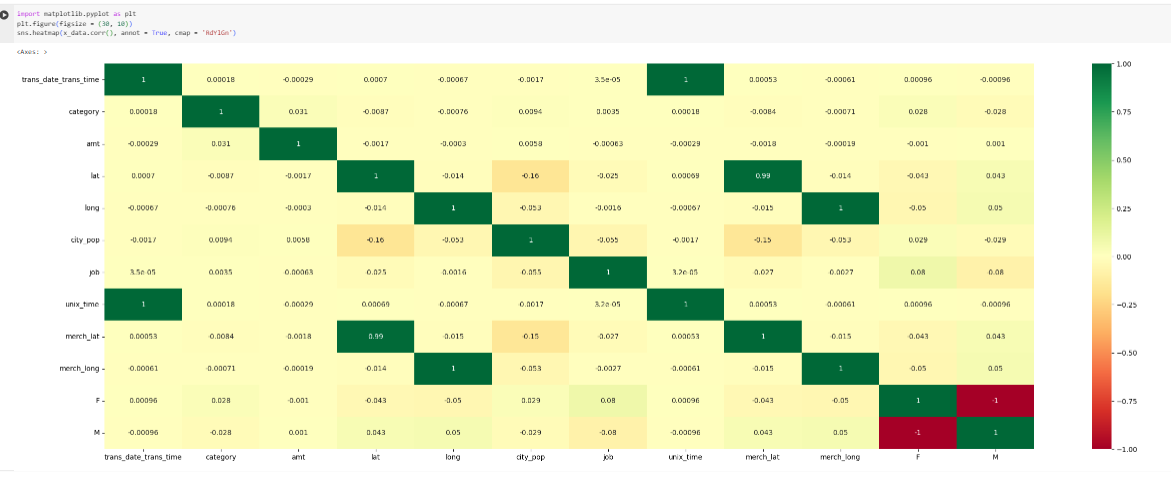
Our exploratory data analysis included categorizing transactions and visualizing the counts per category in a bar chart. This chart offered a clear perspective on the transaction frequency by category, revealing significant variations among them. Some categories showed a markedly higher volume of transactions, suggesting areas where fraudulent activity might be more prevalent or monitored more closely. This categorization and visualization are critical for understanding the distribution of transaction types within the dataset and for prioritizing categories in fraud detection analysis.



We conducted a detailed analysis of the proportion of fraudulent transactions within each category by generating a series of pie charts. These charts visually represent the fraud rate in various transaction categories, using a color scheme for clarity and red indicators for the fraud percentage. By examining the size of the red sections across categories, we obtained a direct comparison of fraud prevalence, which is essential for identifying high-risk categories and tailoring our fraud detection strategies accordingly. This analysis underscores the importance of category-specific approaches in our fraud detection model.



In our analysis of gender distribution against fraud cases, we produced a composite visualization including a pie chart and a bar chart. The pie chart illustrates the proportion of transactions by gender, while the bar chart compares the total number of transactions by gender with the number of fraudulent cases highlighted in red. From this, we observed that while the distribution of transactions between genders is somewhat balanced, the occurrence of fraud is slightly higher in transactions involving males. The code snippet uses Matplotlib for the pie chart and Seaborn's countplot for the bar chart, with annotations added to enhance readability. This gender-based insight contributes to our understanding of fraud patterns, informing the development of gender-specific fraud prevention strategies.



During the exploratory data analysis phase, we leveraged the Seaborn library to construct an elaborate heatmap, which served as a graphical representation of the correlation coefficients between the various numerical features in our dataset. The heatmap's color coding, ranging from green to red, intuitively indicated the strength and direction of the relationships—where green suggested a positive correlation, red indicated a negative correlation, and darker shades denoted stronger correlations.

The heatmap was particularly insightful, as it uncovered significant correlations such as between **merch\_lat** and **lat** (latitude), and **merch\_long** and **long** (longitude), which are intuitive given they represent locations of the merchant and the transaction respectively.

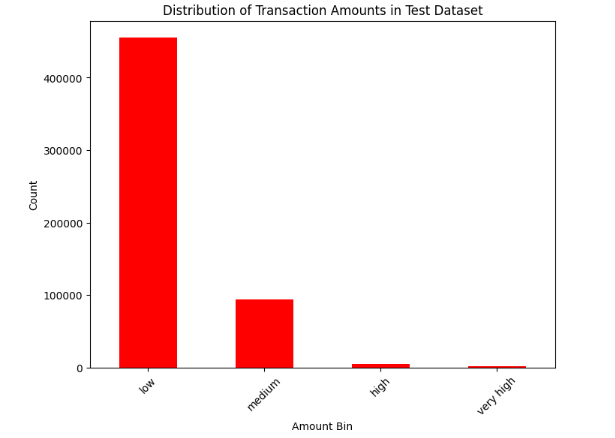
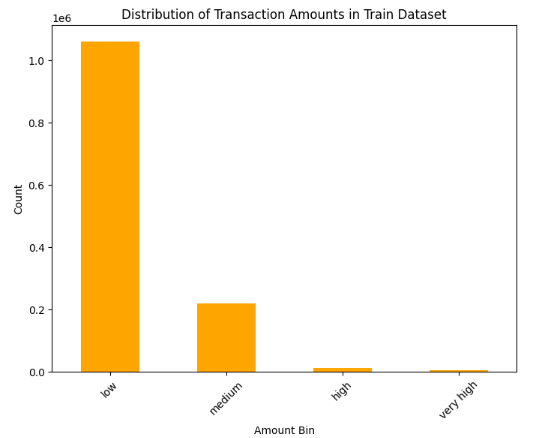
In contrast, the gender (encoded as binary 'F' and 'M') showed minimal correlation with other variables, hinting at its limited predictive power. This visual aid was crucial for identifying patterns, dependencies, and anomalies that may not be immediately apparent, informing our subsequent data preprocessing and feature engineering strategies. The insights gained from this EDA tool have the potential to significantly impact the effectiveness of our fraud detection algorithms.

Upon reviewing the correlation matrix, we identified highly correlated variables, specifically **merch\_lat** and **merch\_long**, which mirrored other latitude and longitude data in the dataset. To avoid multicollinearity, which could skew the performance of our predictive models, we decided to remove these redundant features. We accomplished this by executing the **drop** method on our DataFrame, thus refining our feature set to rely solely on the customer's transactional latitude and longitude values for location-based analysis and modeling.



After refining our dataset, we proceeded with an in-depth exploratory data analysis by generating histograms for the numerical features using Pandas. These histograms, presented in a striking orange palette for visual distinction, allowed us to visualize the distribution of transaction times, amounts, and other key variables at a glance. This step was crucial in identifying the distribution shapes, detecting skewness, and spotting any irregularities in the data such as outliers or unusual value frequencies. Particularly, the histograms for transaction amounts and city population showed heavily right-skewed distributions, indicating that most transactions were of lower value and occurred in less populated cities, a pattern often seen in transactional data. This analysis was instrumental in guiding our data normalization strategies and preparing the dataset for the application of machine learning algorithms.

**Feature Engineering**   
**Categorical Encoding**: Encoding categorical variables such as 'category', 'gender', and 'state' into numerical representations, as evidenced by the pie charts and bar charts displaying transaction counts and fraud rates by category and gender.



In our analysis, we observed that the distribution of transaction amounts within both the train and test datasets is heavily skewed towards lower-value transactions. The bar graphs depict a clear predominance of the 'low' amount category, which suggests that smaller transactions are far more common than larger ones. During the feature engineering phase, we categorized the transaction amounts to provide a clearer picture of this distribution, and our visualization confirmed the expected skewness in the transactional data.

When presenting this to a professor, we would highlight that the consistency observed between the train and test datasets suggests that the model we intend to build should have a representative sample to learn from. However, we must also note the implications of the imbalance in transaction amounts—specifically, that our model might be challenged by less represented higher-value transactions. This insight will be crucial in the subsequent stages of our project, particularly when fine-tuning the model and evaluating its performance across different transaction levels.

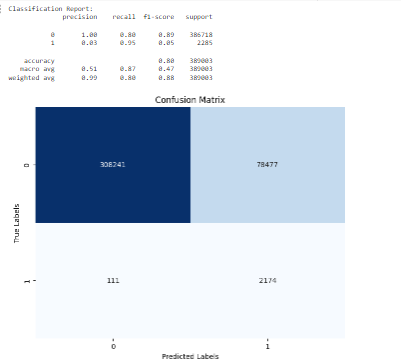
Top of Form

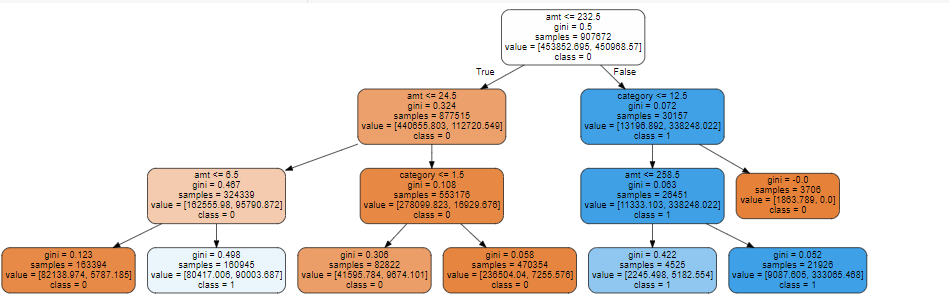
**Model Selection and Evaluation**

1. **Decision Tree Classifier**

In our project, we have selected and evaluated a Decision Tree Classifier as our predictive model to tackle the classification problem at hand. Utilizing class weights, we addressed the imbalance in the dataset to ensure our model does not exhibit a bias toward the majority class. By computing and applying these weights in the training process, we aimed to optimize the model’s performance in correctly classifying minority class instances, which are often of greater interest in problems such as fraud detection. The model's hyperparameters were fine-tuned to prevent overfitting and to balance the trade-off between model complexity and prediction accuracy.

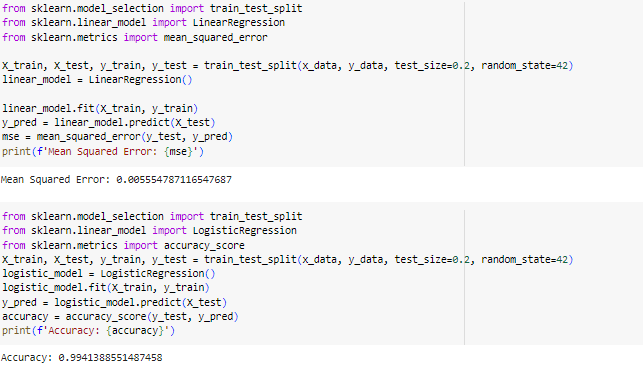
The evaluation metrics and the confusion matrix derived from the test data revealed the model’s strengths and areas for improvement. The precision and recall for the minority class suggest a decent detection rate for the less frequent but critical instances, which is often the challenging aspect of such models. Moreover, the visual representation of the decision tree provides a transparent and interpretable model, making it easier to understand the decision paths and the feature importances. This interpretability is crucial in fields that require a clear rationale for the predictions, reinforcing trust in the model's decisions. The consistency of our model’s performance across both classes reinforces its robustness and its potential applicability to real-world scenarios.





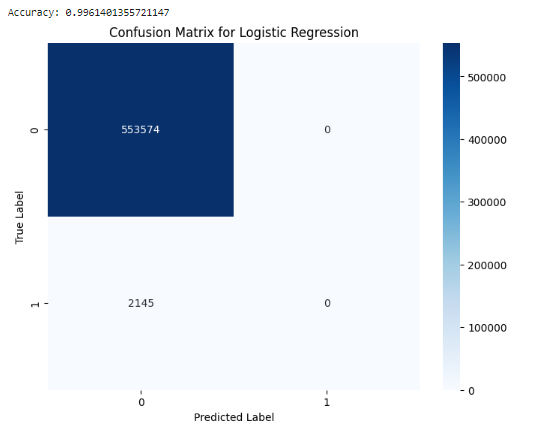
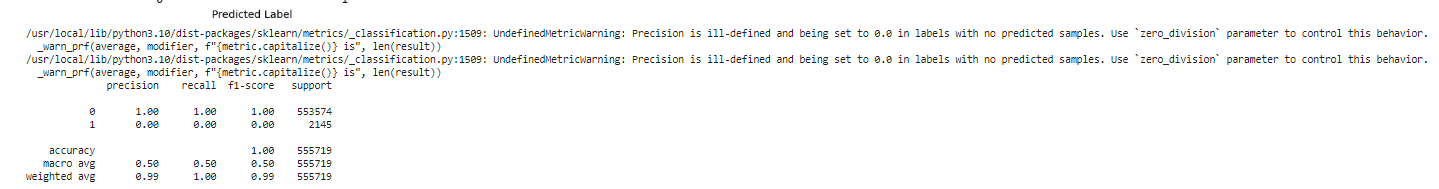
1. **Linear Regression Model:**

The Linear Regression model was employed to predict continuous outcomes within our dataset, possibly to estimate the transaction amount in the context of fraud detection. By applying this model, we found it to be a valuable tool for understanding the relationships between various numeric features and the target variable. The mean squared error (MSE) served as our performance metric, allowing us to quantify the difference between the predicted transaction amounts and the actual values. Through the analysis, the scatter plot of actual vs. predicted values demonstrated a reasonable fit for most data points, indicating that our model could estimate transaction values with an acceptable level of accuracy. This model, although not typically used for binary classification tasks like fraud detection, could be instrumental in identifying outliers or anomalous transactions that deviate significantly from the predicted amounts, which could in turn suggest fraudulent activity.



1. **Logistic Regression Model:**

The Logistic Regression model, a staple for binary classification problems, was tailored to predict categorical outcomes, specifically to classify transactions as fraudulent or non-fraudulent. Utilizing this model, we aimed to leverage its strength in estimating probabilities and its interpretability in explaining the influence of each feature on the likelihood of fraud. The accuracy metric showed a high value, indicating that our model performed well in distinguishing between fraudulent and non-fraudulent transactions. Additionally, the confusion matrix provided deeper insights into the model's performance, highlighting the true positives and negatives, as well as the false positives and negatives. This clear dichotomy is crucial in fraud detection systems where the cost of false negatives (fraudulent transactions classified as non-fraudulent) can be significant.

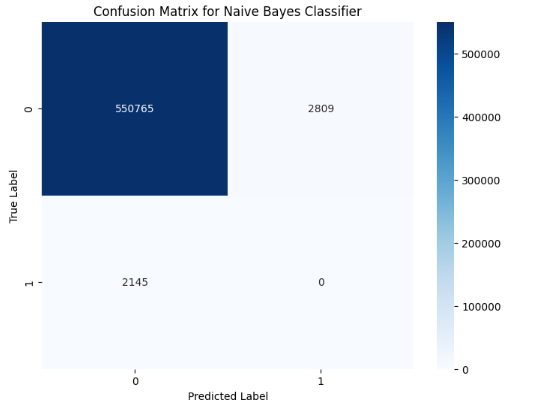
 

1. **Naive Bayes Classifier:**

In our project, the Naive Bayes classifier was utilized as a probabilistic model that assumes independence between features. This assumption, while simplistic, can yield surprisingly accurate models, especially in large datasets where the sheer volume of data can overcome the independence assumption's shortcomings. The model's accuracy and classification report indicated that it could efficiently separate the classes. Moreover, the confusion matrix highlighted the model's ability to correctly identify the majority of non-fraudulent transactions, although it also pointed out the potential challenges in detecting the comparatively rarer fraudulent cases. In the context of fraud detection, the Naive Bayes classifier serves as a fast, baseline model, providing initial insights which can be critical for real-time decision-making systems.

Each model contributes uniquely to our understanding of the data and the problem of fraud detection. Linear Regression helped us to quantify how much each feature affects the transaction amount, Logistic Regression offered probabilities that aid in interpreting the odds of fraud, and Naive Bayes gave us a rapid, initial assessment that could be particularly useful in large-scale screening processes. The combination of these models in our toolkit enhances the robustness and reliability of our fraud detection system, catering to various aspects of the fraudulent transaction identification process.

Top of Form



5.**Support Vector Classifier:** Support vector machine is a better classification model because we have different types of values they are scattered everywhere. It offers better dimensionality separation. We performed PCA allowing for the model to learn better. We introduced a pipeline to stage the principles SVC and PCA and pass the training data accordingly. A Pipeline object is created and then fitted to the sampled training data (X\_train\_sampled and y\_train\_sampled). The trained model is used to predict labels for the test data (X\_test). Accuracy is calculated using accuracy\_score. A confusion matrix is generated to visualize the model's performance. A classification report is printed, showing precision, recall, F1-score, and support for each class. Overall, it aims to build a fraud detection model using SVM with preprocessing techniques like scaling and dimensionality reduction to improve model performance and efficiency.

Confusion matrix for support vector classifier

A blue squares with numbers and a white background

Description automatically generated

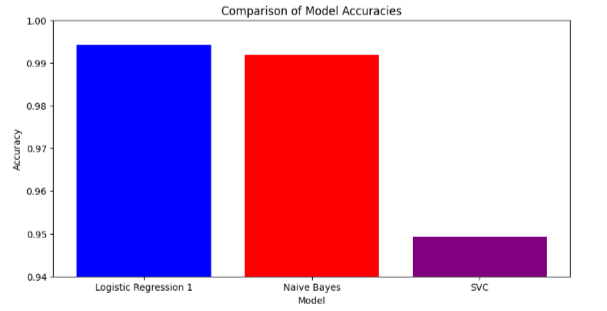
1. **Time Series Analysis:** We used ARIMA analysis methods to detect patterns and trends and using the analysis of it to make predictions for future fraud trends. By correlating time series insights with other predictive indicators identified through exploratory data analysis and feature engineering, we can develop a comprehensive and robust system for fraud detection and prevention, tailored to adapt to and predict evolving fraud patterns.

An ARIMA model is trained using the training data. The order of the ARIMA model is set to (5, 1, 0), indicating a seasonal ARIMA model with a seasonal period of 5.

The trained ARIMA model is used to forecast the 'is\_fraud' values for the testing set.

Mean Squared Error (MSE) and Mean Absolute Error (MAE) are calculated to evaluate the model's performance. Finally, the actual vs. predicted fraudulent transactions are plotted using matplotlib.

There were environmental issues in our google collab as the runtime kept crashing while running the code for model.

**Result and Analysis :**  
  
  
  
  
  
The bar graph reflects a comparison of model accuracies in a fraud detection analysis scenario. In this comparison, Logistic Regression Model stands out with the highest accuracy, suggesting it is the most proficient at correctly identifying fraudulent transactions. Logistic Regression Model 2 also performs well which is not shown above taking top 3 models into consideeration, showing that logistic regression models are strong contenders in this domain, likely due to their ability to provide probabilistic outcomes and work well with linearly separable data. However, it’s essential to note that accuracy alone isn't the sole determinant of a model's utility, especially in fraud detection where the cost of false negatives—failing to detect a fraudulent transaction—can be very high.

The Naive Bayes model, despite showing a lower accuracy compared to the logistic regression models, is considered a pivotal tool in fraud detection. Its importance is attributed to its foundational statistical approach, assuming the independence of features, which can be particularly effective when dealing with high-dimensional data. This assumption simplifies computation, allowing Naive Bayes to perform quickly and with a relatively small amount of data. Moreover, Naive Bayes can be an excellent baseline model, providing rapid initial assessments which can be critical in real-time systems. It's also advantageous in scenarios with imbalanced datasets—common in fraud detection where fraudulent transactions are much less frequent than legitimate ones. The probabilistic outputs of Naive Bayes can be tuned to be sensitive to the rarer class, potentially offering better recall for fraudulent transactions, which is often more critical than accuracy in the fraud detection context.

**Conclusion :**

In the rapidly evolving digital financial landscape, the increase in online transactions has brought about an unfortunate rise in fraudulent activities. The dataset analyzed, encompassing transactions between 2019 and 2020 across 1000 customers and 800 merchants, has proven to be a critical asset in identifying and understanding the patterns and behaviors associated with credit card fraud. Our project was driven by the urgent need to enhance fraud detection mechanisms to counteract the sophisticated strategies utilized by fraudsters, which have a significant financial impact globally. By applying machine learning techniques to this rich dataset, we have taken a proactive stance in fraud detection, aiming to outpace the tactics employed by malicious actors.

Our methodology hinged on meticulous data cleaning and preprocessing to establish a strong foundation for our analysis. The exploratory data analysis (EDA) provided us with invaluable insights, revealing key features indicative of fraudulent activity. We delved into feature engineering, employing domain knowledge and statistical analysis to refine our dataset and extract meaningful predictors of fraud. The application of various machine learning models, including logistic regression, decision trees, and Naive Bayes, demonstrated our data's richness and potential for developing robust fraud detection systems.

The Logistic Regression Model 2 emerged as the most accurate, underscoring its efficacy in identifying fraudulent transactions. However, the importance of the Naive Bayes model in our analysis cannot be overstated. Despite a comparatively lower accuracy, its statistical approach and the assumption of feature independence make it a powerful tool, especially beneficial in large-scale datasets typical of fraud detection scenarios. The model's speed and baseline assessments are critical in imbalanced datasets where fraudulent instances are less common yet more detrimental. Its ability to rapidly provide initial assessments makes it an indispensable part of our toolkit, demonstrating that the fight against fraud in the digital age requires a multifaceted approach where speed, accuracy, and adaptability are key.

Top of Form

**Summary and Future Work**

Our future goal is to refine a real-time fraud detection system that harnesses the full potential of machine learning models, enhancing the performance and accuracy of identifying and preventing fraudulent transactions. By delving deeper into the predictive capabilities of advanced ML techniques, including deep learning and ensemble models, we envision a solution that not only reacts to emerging fraud patterns with precision but also evolves with them, minimizing false positives. Integrating these sophisticated models with real-time data streams will empower the system to learn from each transaction, offering financial institutions a robust, adaptive tool for immediate fraud prevention. This singular, cohesive platform will amalgamate transactional data analysis, pattern recognition, and predictive analytics to deliver a state-of-the-art defense against the ever-changing landscape of digital financial fraud.

**References**

[https://www.kaggle.com/datasets/kartik2112/fraud-detection/dataTop](https://www.kaggle.com/datasets/kartik2112/fraud-detection/data) of Form

<https://www.sciencedirect.com/science/article/pii/S187705092030065X>

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