VINUNIVERSITY

COMPUTATIONAL MACHINE LEARNING
FOR BUSINESS ANALYTICS COURSE

GROUP PROJECT REPORT - GROUP 1

CREDIT CARD FRAUD DETECTION USING RANDOM FOREST

Team member: Le Ba Hung
Nguyen Trang Nhung
Pham Thi Ngoc Mai
Tran Tue Nhi

I. BUSINESS PROBLEM

1. The rise of credit cards

Prior to 1996, the majority of our day-to-day transactions occurred in physical form, typically through bank checks or cash. However, the advent of the internet in 1996 brought about a new era in banking, with the introduction of cashless payment methods such as credit cards (Afriyie et al., 2023). This innovation revolutionized the payment process, making it more convenient for customers. Consequently, there has been a notable surge in online transactions and purchases, particularly with the growing prominence of e-commerce in recent years.

2. Definition and statistics

A credit card is a financial service provided by financial institutions, wherein customers are typically issued with a digital or physical card to access funds and make purchases on credit. These purchases incur interest charges on the borrowed amount. NotablyIn fact, in the USA, credit cards are an extremely profitable business for banks and financial institutions, with an estimated market of more than 10.4 trillion USD in 2022 (Pokora, 2024) (see Figure 1).

The credit card market has grown substantially over the past few decades. Forbes research indicates that over 200 million Americans possess at least one credit card, with half holding more than two. This increasing number of credit cards has allowed the financial institution to access a significant amount of data from its customers, from usual spending to private information.

However, alongside the rise in credit card usage comes an uptick in security vulnerabilities, leading to unlawful exploitation. These are known as credit card fraud, which occurs when credit card information is obtained from various websites and measures. Because of their significant cost to the industry, these frauds have been a constant source of concern for clients and financial institutions.

3. The emergence of credit card fraud

During the COVID-19 pandemic, the surge in e-commerce and online services led to a substantial increase in online transactions. With 83% of American adults owning credit cards, these have become a crucial payment method (Flynn, 2023). In 2020, e-commerce accounted for over 20% of global sales, creating more opportunities for credit card fraud (see Figure 2).

Credit card fraud is a significant and growing threat to the financial sector, with global losses reaching $28.65 billion in 2019, according to the Nilson Report. The U.S. alone accounts for over a third of these losses, with $11 billion in credit card fraud reported in 2020, as noted by Julie Conroy of Aite Group (Lee, 2021). The coronavirus pandemic has further exacerbated the issue, with increased fraud activity during economic downturns. Julie Fergerson, CEO of Merchant Risk Council, expects fraud numbers to rise significantly in the coming years (Lee, 2021).

Credit card fraud has created tension between customers and financial institutions, primarily due to inadequate security measures leading to financial losses for customers. Despite the implementation of multiple fraud detection models, hackers continually find new methods to exploit loopholes. Therefore, there is a need for more robust and up-to-date models to enhance security.

4. Machine learning to the rescue

Technological solutions are critical in this fight. Mike Lemberger of Visa reported that Visa's A.I. technology prevented $25 billion in fraud 2023 (Lee, 2021). However, experts like Colin Sims of Forter remain skeptical about fully solving the issue, citing the persistent risks associated with digital money transfers. Developing robust fraud detection and prevention methods remains essential for protecting financial transactions.

Research has shown that an increase in security measures leads to a decrease in customer reluctance to use credit cards (Burt, 2019). Enhanced security would not only protect consumers from financial losses but also improve the reputation of financial institutions. This, in turn, could lead to higher customer retention and increased usage of credit card services.

Moreover, by reducing the incidence of fraud, businesses can lower their operational costs related to fraud management and chargebacks. This would enable them to allocate resources more efficiently, potentially leading to better customer service and innovation in other areas. The overall trust in digital transactions would be strengthened, fostering a more secure and reliable financial ecosystem. This advancement would benefit all stakeholders involved, from consumers to financial institutions and merchants, creating a safer and more trustworthy environment for conducting financial transactions.

Credit card fraud detection falls under binary classification, where transactions are categorized as either fraudulent or non-fraudulent. Traditional models like logistic regression or decision trees often struggle with this task due to the imbalance between the number of fraud and non-fraud cases, leading to poor accuracy

(Baesens et al., 2021). According to research, the imbalance in fraud datasets creates significant challenges in building effective models (Mohamad Aburbeian & Ashqar, 2022). However, Random Forest's ensemble approach helps mitigate this issue by leveraging multiple decision trees to enhance detection accuracy and robustness, making it well-suited for handling the complexities of credit card fraud detection.

II. MODEL BACKGROUND: RANDOM FOREST

1. Overview

Random Forest is a supervised learning technique grounded in ensemble learning principles. Ensemble learning involves combining the predictions of a multitude of models to identify the most agreed-on result. In line with this, the random forest algorithm leverages a set of decision trees, where each tree's decision is contingent upon a random subset of input features. This stochastic selection process is uniform across all trees within the forest (Breiman, 2001). The outcome of random forest is based on the aggregated decisions across various decision trees.

2. Background

a. Decision trees

Since decision trees are fundamental for developing the random forest model, brief understanding of the decision tree algorithm would be beneficial. Decision tree is, as the name suggests, a flowchart-like structure which delineates a series of decision nodes to dissect data and ultimately arrive at an outcome (Prajwala, 2015). Each decision node specifies a test on a single attribute, guiding the branching logic towards a class label denoted by a leaf node. By iteratively going through decision nodes, the model segments the dataset into increasingly homogeneous subsets. The essence of decision trees lies in their quest to identify optimal splits that effectively partition the data, facilitating accurate predictions. (What is Random Forest, 2021)

b. Ensemble methods

In machine learning, the question of whether more models improve performance is tackled by ensemble learning. This approach combines multiple models' outputs, treating them as a committee of decision-makers to achieve higher accuracy than any single model (Brown, 1970). The most popular ensemble method introduced in 1996 by Leo Breiman was coined “bootstrap aggregating”, or “bagging” (Breiman, 1996). In the context of random forest, each decision tree model is allowed to randomly sample a subset of the training data with replacement (any individual data point can be selected more than once) (What is Random Forest, 2021). Each decision tree operates independently, and the final prediction is determined by aggregating the outputs of all trees within the forest. This is helpful in reducing variance within a noisy dataset.

c. Random forest

The random forest algorithm utilizes both bagging and feature randomness to construct an uncorrelated ensemble of decision trees. “Feature randomness, or "random subspace method", involves generating a random subset of features to ensure minimal correlation among decision trees” (What is Random Forest, 2021). This leads to a key distinction: unlike decision trees which consider all potential feature splits, random forests only utilize a subset of features for each tree. Considering all potential sources of variability within the data mitigates the risk of overfitting, bias, and overall variance, thereby enhancing the precision of predictions.

3. Properties

a. Input

The random forest model can handle binary, continuous, and categorical data. It takes in different transaction attributes, encompassing variables such as transaction amount, timestamp, geographical location, merchant category, among others. These attributes serve as the foundation for the model's predictive capabilities.

b. Output

For classification tasks, the output of the random forest is the class selected by most trees. In fraud detection tasks, the output is a binary prediction denoting the likelihood of a transaction being fraudulent or legitimate, indicated respectively by '1' or '0'.

c. Parameters (Probst et al, 2019) (see Figure 3)

- mtry: The lower values of mtry result in more diverse and less correlated trees, which can enhance the stability of the aggregated predictions. However, the trade-off is lower accuracy on each tree because the splits are made from a smaller set of candidate variables, potentially ignoring more informative splits. By default, mtry of $\sqrt{p}$ for classification is a reasonable value but can be improved depending on the amount of relevant predictor variables. Lower mtry values generally lead to better performance in classification tasks with many relevant variables, as they allow less influential variables to contribute. Conversely, higher mtry values are beneficial in high-dimensional settings or when relevant variables are few, ensuring that important variables are likely included in the candidate set. Computation time decreases approximately linearly with lower mtry values due to fewer variables being evaluated for splits.

- **Sample size:** Similar to mtry, decreasing the amount of observations lowers the correlation of decision trees but impedes the performance of individual trees. Optimal sample size is problem-dependent and can be estimated using out-of-bag predictions.

Better performance is often observed with smaller than default sample sizes (default usually equals the dataset size with replacement).

- **Node size:** Decreasing the node size increases the depth and splits that the model gets to train. Smaller node sizes generally improve performance by allowing more detailed splits but cost more computational time. The default value of node size for binary classification is usually 1 (Kassambara, 2018).

- **Number of Trees:** Some evidence suggests that the choice of number of trees might depend on the dataset (the number of rows) (How many trees in the random forest?, 2020). In general, a larger number of trees could reduce overfitting and improve performance, but the marginal benefit decreases as more trees are added. The best performance is usually obtained with 100 trees in a large data set.

- **Splitting rule:** The default split strategy usually follows a rule to choose a split from mtry variables that minimizes the Gini impurity. The Gini impurity allows the optimal split to be chosen to split the nodes.

4. Why random forest?

- **Complexity and flexibility:** Random Forests can model intricate patterns and interactions between variables without overfitting, making them suitable for complex scenarios like fraud detection. Moreover, compared to more complex models like support vector machines, Random Forests are faster to train and evaluate, providing a practical solution for real-time fraud detection (Kumar, 2021).

- **Reduced risk of overfitting and noise:** Decision trees are prone to overfitting, especially when they are deep and complex. However, Random Forests mitigate this risk by aggregating the predictions of multiple uncorrelated trees as discussed above. This averages out the errors of individual trees, reducing the overall variance and prediction error. As a result, Random Forests generalize better to new, unseen data, making them reliable for fraud detection where new fraud patterns continually emerge.

- **Handling missing values:** Missing values is a common issue in transactional data. The feature bagging process ensures that the model remains robust even when some data points are missing.

- **Feature importance and interpretability:** Random forest provides valuable insights into the importance of various features in predicting fraud using various methods such as Gini impurity. This interpretability is essential for fraud analysts to understand which transaction attributes are most indicative of fraudulent behavior. It aids in refining fraud detection strategies and improving the overall effectiveness of the fraud detection system.

- **Proven effectiveness in fraud detection:** Random Forests have been widely adopted and validated in the field of fraud detection. Numerous studies and practical applications have demonstrated their effectiveness in identifying fraudulent activities with high accuracy (Aburbeian & Ashqar, 2023).

III. DATA DESCRIPTION

1. Data Overview and Sources

The dataset we chose is "Fraud Detection" sourced from Kaggle. It includes simulated credit card transactions including both legitimate and fraudulent activities between January 1, 2019 to December 31, 2020. This extensive dataset includes transactions from 1,000 customers interacting with 800 different merchants, providing a rich and diverse set of data points for analysis. The dataset is significantly imbalanced, with fraudulent transactions representing a very small percentage of total transactions. This poses a common challenge in fraud detection, requiring the use of advanced techniques to ensure accurate detection of rare fraudulent events.

2. Types of Data

The dataset contains numerical, categorical and time series data:

- Numerical: amt, lat, long, city_pop, unix_time, merch_lat, merch_long, is_fraud

- Categorical: zip, cc_num, merchant, category, first, last, gender, street, city, state, job, trans_num, unix_time

- Time Series: trans_date_trans_time, dob

3. Data Relevance

The dataset's comprehensive features, including temporal, geographical, transaction, customer, and merchant data, are beneficial for detecting credit card fraud with a Random Forest model. Temporal and location data help identify unusual patterns, while transaction details and customer profiles reveal anomalies. Merchant information and the fraud label enable the model to learn from historical data and accurately predict fraud. This rich data ensures effective detection and prevention of fraudulent activities, minimizing financial losses and enhancing transaction security.

4. Compatibility with Random Forest

The dataset is compatible with the Random Forest model after the following pre-processing steps:

- Data Cleaning
- Normalize
- Encode
- Feature Engineering

IV. PIPELINE

STEP 1. Data Preprocessing

a. Collect Data:

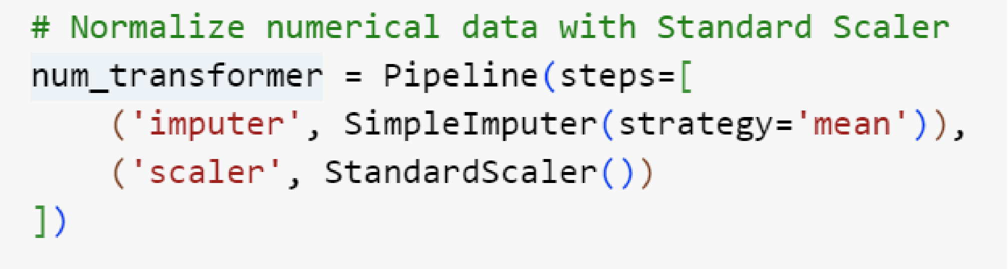
The data is downloaded from the Kaggle dataset which is a publicly available dataset and has undergone pre-processing to ensure anonymity and data integrity. The dataset is structured and formatted in CSV, making it suitable for direct ingestion into machine learning pipelines. Then we load the data as two separate datasets, “train_data” and “test_data”, to facilitate the model training and evaluation processes.

b. Prepare the Data:

- Clean data:

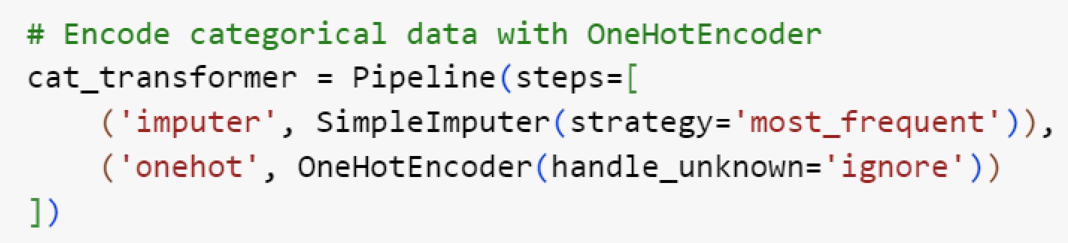
+ Check for missing value: Here we can see 21 data points missing in both datasets (see Figure 4).
+ Address missing values: Since 21 is significantly smaller compared to the total instances in the datasets, it can be simply dealt with by dropping null values (see Figure 5).

- **Normalize**: Normalize numerical data with Standard Scaler



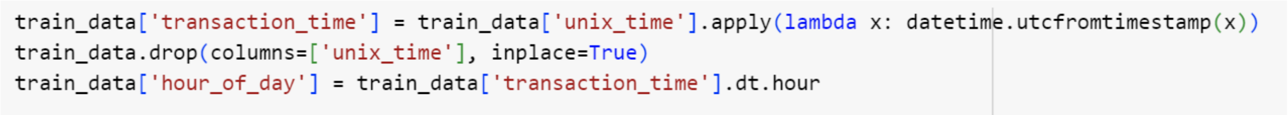
```
# Normalize numerical data with Standard Scaler
num_transformer = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='mean')),
    ('scaler', StandardScaler())
])
```

- Encode: Transform categorical data into numeric formats through one-hot encoding for categorical features



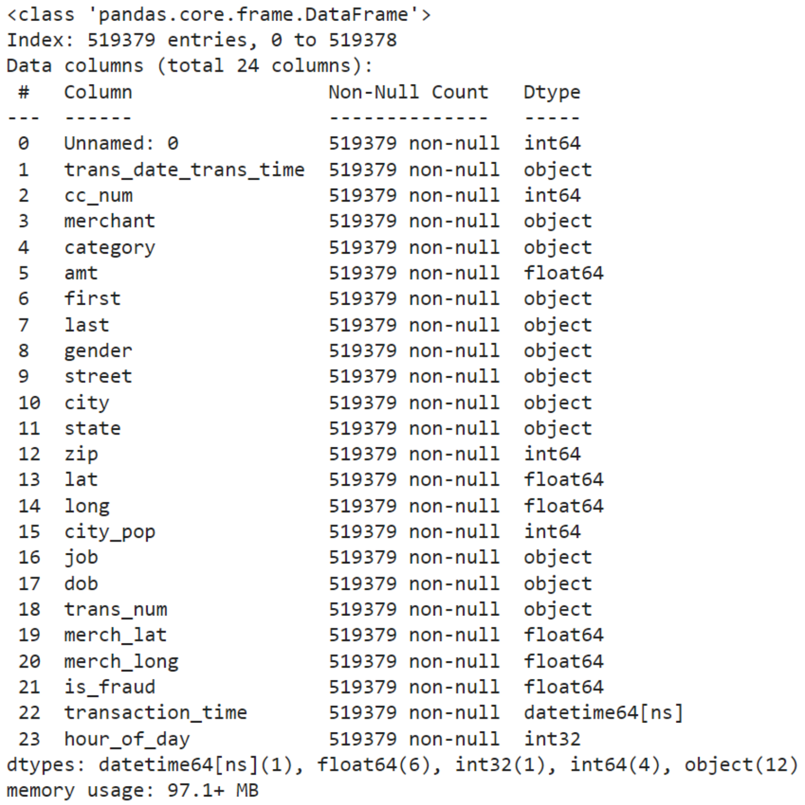
```
# Encode categorical data with OneHotEncoder
cat_transformer = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='most_frequent')),
    ('onehot', OneHotEncoder(handle_unknown='ignore'))
])
```

- Feature Engineering: Create new feature "hour of the day" to enhance patterns analysis



```
train_data['transaction_time'] = train_data['unix_time'].apply(lambda x: datetime.utcfromtimestamp(x))
train_data.drop(columns=['unix_time'], inplace=True)
train_data['hour_of_day'] = train_data['transaction_time'].dt.hour
```

=> New columns after adding feature



```
<class 'pandas.core.frame.DataFrame'>
Index: 519379 entries, 0 to 519378
Data columns (total 24 columns):
 #   Column                 Non-Null Count   Dtype
---  ------                 --------------   -----
 0   Unnamed: 0             519379 non-null  int64
 1   trans_date_trans_time  519379 non-null  object
 2   cc_num                 519379 non-null  int64
 3   merchant               519379 non-null  object
 4   category               519379 non-null  object
 5   amt                    519379 non-null  float64
 6   first                  519379 non-null  object
 7   last                   519379 non-null  object
 8   gender                 519379 non-null  object
 9   street                 519379 non-null  object
 10  city                   519379 non-null  object
 11  state                  519379 non-null  object
 12  zip                    519379 non-null  int64
 13  lat                    519379 non-null  float64
 14  long                   519379 non-null  float64
 15  city_pop               519379 non-null  int64
 16  job                    519379 non-null  object
 17  dob                    519379 non-null  object
 18  trans_num              519379 non-null  object
 19  merch_lat              519379 non-null  float64
 20  merch_long             519379 non-null  float64
 21  is_fraud               519379 non-null  float64
 22  transaction_time       519379 non-null  datetime64[ns]
 23  hour_of_day            519379 non-null  int32
dtypes: datetime64[ns](1), float64(6), int32(1), int64(4), object(12)
memory usage: 97.1+ MB
```

STEP 2. Exploratory Data Analysis

- Spending Category VS Fraud (see Figure 6): This chart helps identify which spending categories are more susceptible to fraud compared to their normal distribution.

The most significant finding is that grocery_pos transactions are notably more susceptible to fraud, showing the highest figure. This suggests that point-of-sale transactions in grocery stores are a prime target for fraudsters. This vulnerability is likely due to the high volume of transactions in grocery stores and potentially less stringent verification processes at checkout points. The rapid pace and sheer number of transactions in grocery stores might overwhelm basic security measures, allowing fraudulent activities to go undetected more easily. Following closely are shopping_net, misc_net, gas_transportation, shopping_pos.

On the other hand, categories like misc_pos, grocery_net, health_fitness, travel, kids_pets show lower levels of fraud despite still being present. This lower frequency of fraud suggests that these categories are either less attractive or more difficult targets for fraudsters. For example, travel transactions might involve higher scrutiny due to their typically higher value, and purchases related to kids and pets might not attract as much fraudulent activity due to their specialized nature. Despite the lower incidence of fraud, these categories should not be neglected as they still pose potential risks.

- Gender VS Fraud (see Figure 7): This chart shows if there's any gender bias in fraudulent transactions, helping understand demographic vulnerabilities.

It is worth noticing that female cardholders fall victim to credit card frauds slightly less than their male counterparts. However, the data shows that the distribution appears quite similar for both genders.

The key takeaway is that there is no significant gender bias in fraudulent transactions. This suggests that demographic vulnerabilities in terms of gender are not a major factor in this dataset. The fraud detection model should thus focus more on transactional and behavioral patterns rather than demographic factors such as gender.

- State VS Fraud (see Figure 8): Identifies states with higher fraud rates, useful for geographically targeted interventions.

The consistent presence of fraud across all states underscores the importance of robust fraud detection mechanisms nationwide. High-transaction states like North Carolina, California, Texas, and New York require particular attention due to their larger volume of transactions, both legitimate and fraudulent. However, the consistent presence of fraud in every state indicates that anti-fraud measures should be uniformly implemented nationwide.

- Cyclicality of Credit Card Fraud (see Figure 9): Understanding temporal patterns in fraud can help in scheduling fraud detection activities more effectively.

Fraudulent activities exhibit a notable surge between the late hours of 10 PM and 3 AM. Conversely, the early morning period, spanning from 4 AM to 11 AM, witnesses a stark decline in fraudulent incidents, with occurrences rarely surpassing 5 cases per hour. This trend suggests a clear preference among fraudsters for conducting their illicit activities during the late night, potentially capitalizing on decreased surveillance during these hours.

Key takeaways include focusing fraud detection efforts and resources during peak hours, especially late at night, to enhance effectiveness. Understanding these temporal patterns allows for better scheduling of monitoring activities and allocation of resources to mitigate fraud risks during the most vulnerable periods.

STEP 3. Train the Model: Configure and train a random forest model using the training data.

- Handle imbalance dataset: Use SMOTE to improve model performance

- Split the Data: Divide the data into training and testing sets to evaluate model performance later.

- Model Training: Use logistic regression to fit the model on the training data.

STEP 4. Evaluate the Model: Assess the model with the testing data using metrics suited for imbalance, such as precision, recall, and F1-score (see Figure 10).

- **Make Predictions**: Use the trained model to predict the labels of the testing data.

- **Calculate Metrics**: Compute precision, recall, and F1-score to understand the model's performance, especially in handling the imbalanced nature of the dataset.

STEP 5. Model Optimization and Tuning

Hyperparameter Tuning for Random Forest Classifier: Generalizes well to new, unseen data and provides reliable predictions. Use Random Grid to find the best combination of hyperparameters for the Random Forest classifier.

STEP 6. Re-evaluate the Optimized Model

Re-evaluate on Testing Data (see Figure 11): Use the optimized model to predict and calculate evaluation metrics on the testing data again.

V. CONSTRAINTS

A consideration of the feasibility of scaling the model onto an actual product, with real-life business constraints.

- Data Quality: The model's efficacy hinges significantly on data quality. Even minor deviations in data quality can lead to substantial discrepancies in performance. Real-world data is often incomplete or biased, which can cause poor model performance post-deployment. For example, a minor error in property details like room count can result in substantial financial discrepancies, as seen with Willow (Top 10 ML Model Failures, 2024).

- Real-time Processing Speed: Fraud detection demands swift processing, often in real-time or near-real-time. However, Random Forest models, due to their computational complexity, can introduce delays. In the banking industry, where time sensitivity is paramount, systems must exhibit low latency to swiftly evaluate transactions and make immediate decisions.

- Extra costs: A Random Forest model involves significant computational resources. Businesses need robust infrastructure (cloud computing, distributed systems) to handle the load. Implementing and maintaining such infrastructure can be costly.

- Interpretability and Compliance

+ **Explainability:** Random Forest models are generally more interpretable than other complex models, but explaining decisions to non-technical stakeholders and regulatory bodies can still be challenging.

+ **Compliance:** Adherence to regulations (e.g., GDPR, PCI DSS) regarding data privacy and security is mandatory. The model and data processing must comply with these regulations.

- Integration with Existing Systems

+ **Compatibility:** The model needs to integrate seamlessly with existing transaction processing systems, customer databases, and other IT infrastructure.

+ **APIs and Interfaces:** Development of APIs and user interfaces to facilitate smooth interaction between the model and business applications.

VI. CONCLUSION

Credit card fraud is a significant concern with serious repercussions, necessitating robust measures to predict and prevent fraudulent activities. Given the typically imbalanced nature of fraud datasets, choosing an appropriate model is crucial. Random Forest emerges as a strong candidate due to its complexity and flexibility, reduced risk of overfitting and noise, ability to handle missing values, feature importance and interpretability, and proven effectiveness in fraud detection. However, several considerations must be addressed post-deployment. Ensuring high data quality, maintaining real-time processing speed, managing extra costs, and ensuring interpretability and compliance are critical. Additionally, seamless integration with existing systems is essential for the model's effective implementation and sustained success in combating credit card fraud.

REFERENCES

Aburbeian, A. M. & Ashqar, H. I. (2023). Credit card fraud detection using enhanced random forest classifier for imbalanced data. https://www.researchgate.net/publication/369199151_Credit_Card_Fraud_Detection_Using_Enhanced_Random_Forest_Classifier_for_Imbalanced_Data

Afriyie, J. K., Tawiah, K., Pels, W. A., Addai-Henne, S., Dwamena, H. A., Owiredu, E. O., Ayeh, S. A., & Eshun, J. (2023). A supervised machine learning algorithm for detecting and predicting fraud in credit card transactions. Decision Analytics Journal, 6, 100163. https://doi.org/10.1016/j.dajour.2023.100163

Baesens, B., Höppner, S., & Verdonck, T. (2021). Data engineering for fraud detection. Decision Support Systems, 150, 113492. https://doi.org/10.1016/j.dss.2021.113492

Breiman, L. (1996). Bagging predictors. https://link.springer.com/content/pdf/10.1007/BF00058655.pdf

Breiman, L. (2001, October). Random forests - machine learning. SpringerLink. https://link.springer.com/article/10.1023/a:1010933404324

Brown, G. (1970, January 1). Ensemble learning. SpringerLink. https://link.springer.com/referenceworkentry/10.1007/978-0-387-30164-8_252

Burt, A. (2019, August 23). Cybersecurity Is Putting Customer Trust at the Center of Competition. Harvard Business Review. https://hbr.org/2019/03/cybersecurity-is-putting-customer-trust-at-the-center-of-competition

Credit Card Transactions Fraud Detection Dataset. (2020, August 5). Kaggle. https://www.kaggle.com/datasets/kartik2112/fraud-detection?datasetId=817870&sortBy=voteCount

Dcallahan, & Dcallahan. (2023, May 5). 2023 Findings from the Diary of Consumer Payment Choice - San Francisco Fed. SF Fed. https://www.frbsf.org/research-and-insights/publications/fed-notes/2023/05/2023-findings-from-the-diary-of-consumer-payment-choice/

Flynn, J. (2023, June 28). 30+ Credit Card Statistics [2023]: Credit Card Debt, Fraud, Usage, And Ownership Facts. Zippia. https://www.zippia.com/advice/credit-card-statistics/

Kassambara. (2018, March 10). Bagging and Random Forest Essentials. STHDA. http://www.sthda.com/english/articles/35-statistical-machine-learning-essentials/140-bagging-and-ra ndom-forest-essentials/#:~:text=Note%20that%2C%20the%20random%20forest,default%20for%20regression%20is%205

Kumar, P. (2021, June 26). Time complexity of ML Models. Medium. https://medium.com/analytics-vidhya/time-complexity-of-ml-models-4ec39fad2770

Lee, J. (2021, February 1). Credit card fraud will increase due to the Covid pandemic, experts warn. CNBC. https://www.cnbc.com/2021/01/27/credit-card-fraud-is-on-the-rise-due-to-covid-pandemic.html

MLJAR. (2020, June 30). How many trees in the random forest? https://mljar.com/blog/how-many-trees-in-random-forest/

Obinopaul. (n.d.). *GitHub - obinopaul/Credit-Card-Fraud-Detection-Using-Random-Forest: This repository contains a credit card transactions fraud detection model built using the Random Forest algorithm. The model uses a large dataset of credit card transactions to train the algorithm and identify patterns and anomalies that may indicate fraud.* GitHub. https://github.com/obinopaul/Credit-Card-Fraud-Detection-Using-Random-Forest?tab=readme-ov-file

Pokora, B. (2024, March 28). Credit Card Statistics And Trends 2024. Forbes Advisor. https://www.forbes.com/advisor/credit-cards/credit-card-statistics/

Prajwala, T. R. (2015, January). A comparative study on decision tree and Random Forest using r tool. https://www.researchgate.net/publication/272425234_A_Comparative_Study_on_Decision_Tree_and_Random_Forest_Using_R_Tool

Probst, P., Wright, M., Boulesteix, A. L. (2019, February 27). Hyperparameters and tuning strategies for Random Forest. https://arxiv.org/pdf/1804.03515

Shenoy, K. (2020, August 5). *Credit Card Transactions Fraud Detection Dataset*. Kaggle. https://www.kaggle.com/datasets/kartik2112/fraud-detection?datasetId=817870&sortBy=voteCount

Top 10 ML model failures you should know about. (2024, March 30). Deepchecks. https://deepchecks.com/top-10-ml-model-failures-you-should-know-about/

What is Random Forest? (2021, October 20). IBM. https://www.ibm.com/topics/random-forest

APPENDIX

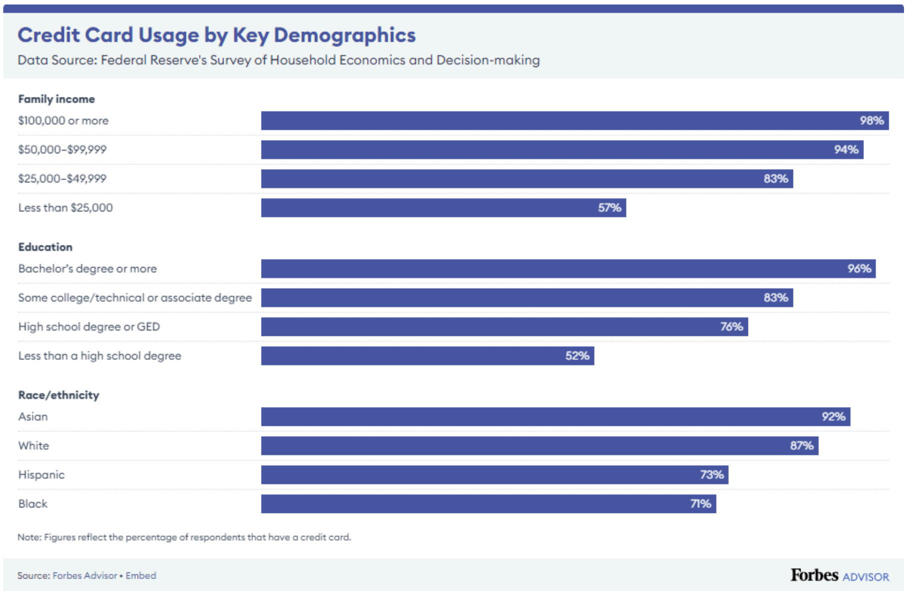


Figure 1 (Pokora, 2024)

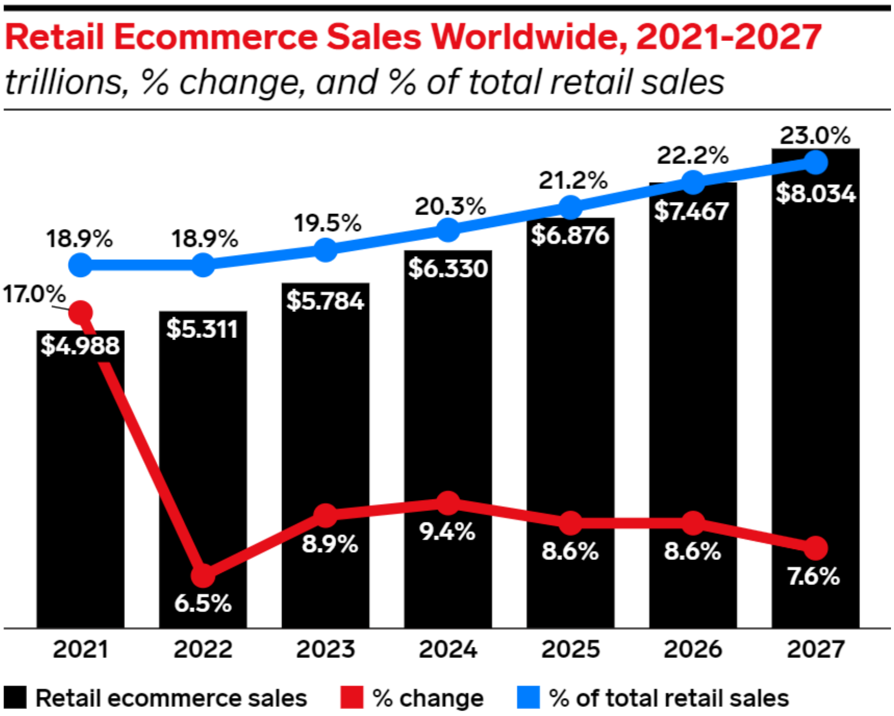
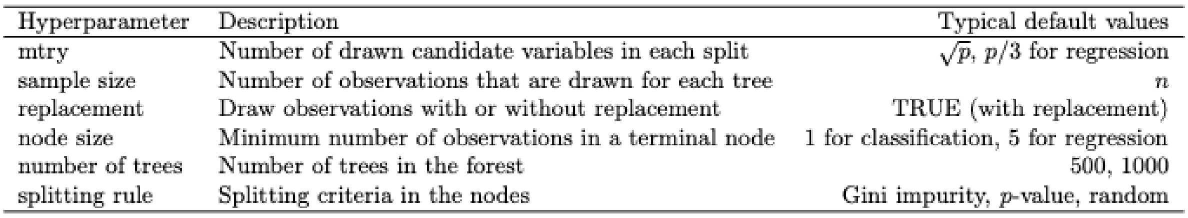
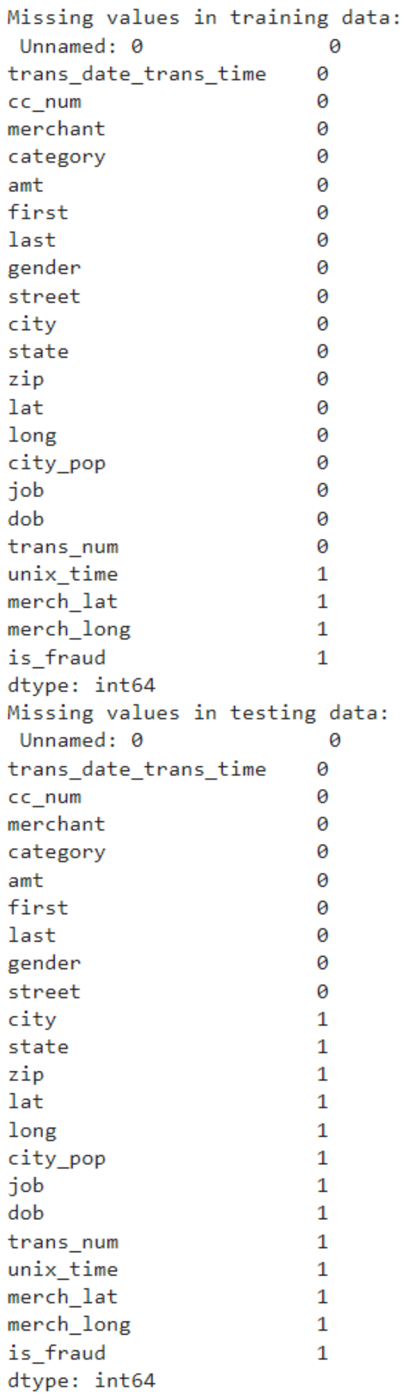


Figure 2



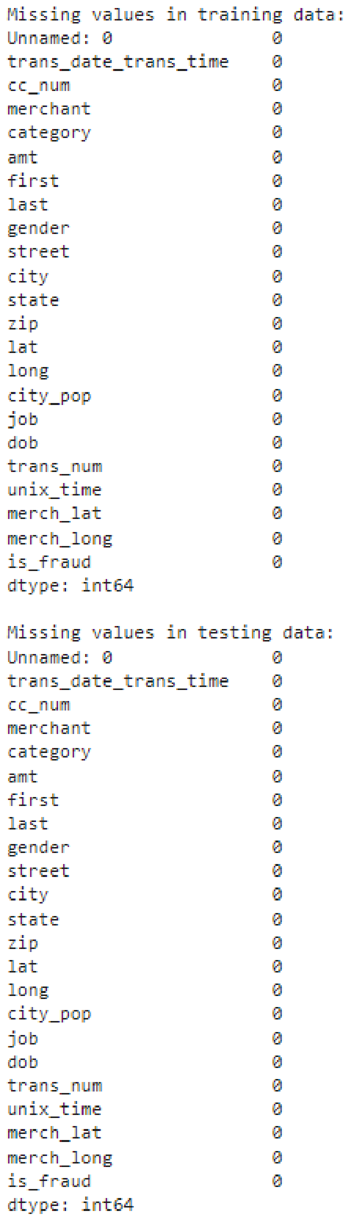
Hyperparameter	Description	Typical default values
mtry	Number of drawn candidate variables in each split	$\sqrt{p}$, $p/3$ for regression
sample size	Number of observations that are drawn for each tree	n
replacement	Draw observations with or without replacement	TRUE (with replacement)
node size	Minimum number of observations in a terminal node	1 for classification, 5 for regression
number of trees	Number of trees in the forest	500, 1000
splitting rule	Splitting criteria in the nodes	Gini impurity, p-value, random

Figure 3 (Probst et al, 2019)



```
Missing values in training data:
 Unnamed: 0               0
trans_date_trans_time    0
cc_num                   0
merchant                 0
category                 0
amt                      0
first                    0
last                     0
gender                   0
street                   0
city                     0
state                    0
zip                      0
lat                      0
long                     0
city_pop                 0
job                      0
dob                      0
trans_num                0
unix_time                1
merch_lat                1
merch_long               1
is_fraud                 1
dtype: int64
Missing values in testing data:
 Unnamed: 0               0
trans_date_trans_time    0
cc_num                   0
merchant                 0
category                 0
amt                      0
first                    0
last                     0
gender                   0
street                   0
city                     1
state                    1
zip                      1
lat                      1
long                     1
city_pop                 1
job                      1
dob                      1
trans_num                1
unix_time                1
merch_lat                1
merch_long               1
is_fraud                 1
dtype: int64
```

Figure 4



```
Missing values in training data:
Unnamed: 0               0
trans_date_trans_time    0
cc_num                   0
merchant                 0
category                 0
amt                      0
first                    0
last                     0
gender                   0
street                   0
city                     0
state                    0
zip                      0
lat                      0
long                     0
city_pop                 0
job                      0
dob                      0
trans_num                0
unix_time                0
merch_lat                0
merch_long               0
is_fraud                 0
dtype: int64

Missing values in testing data:
Unnamed: 0               0
trans_date_trans_time    0
cc_num                   0
merchant                 0
category                 0
amt                      0
first                    0
last                     0
gender                   0
street                   0
city                     0
state                    0
zip                      0
lat                      0
long                     0
city_pop                 0
job                      0
dob                      0
trans_num                0
unix_time                0
merch_lat                0
merch_long               0
is_fraud                 0
dtype: int64
```

Figure 5

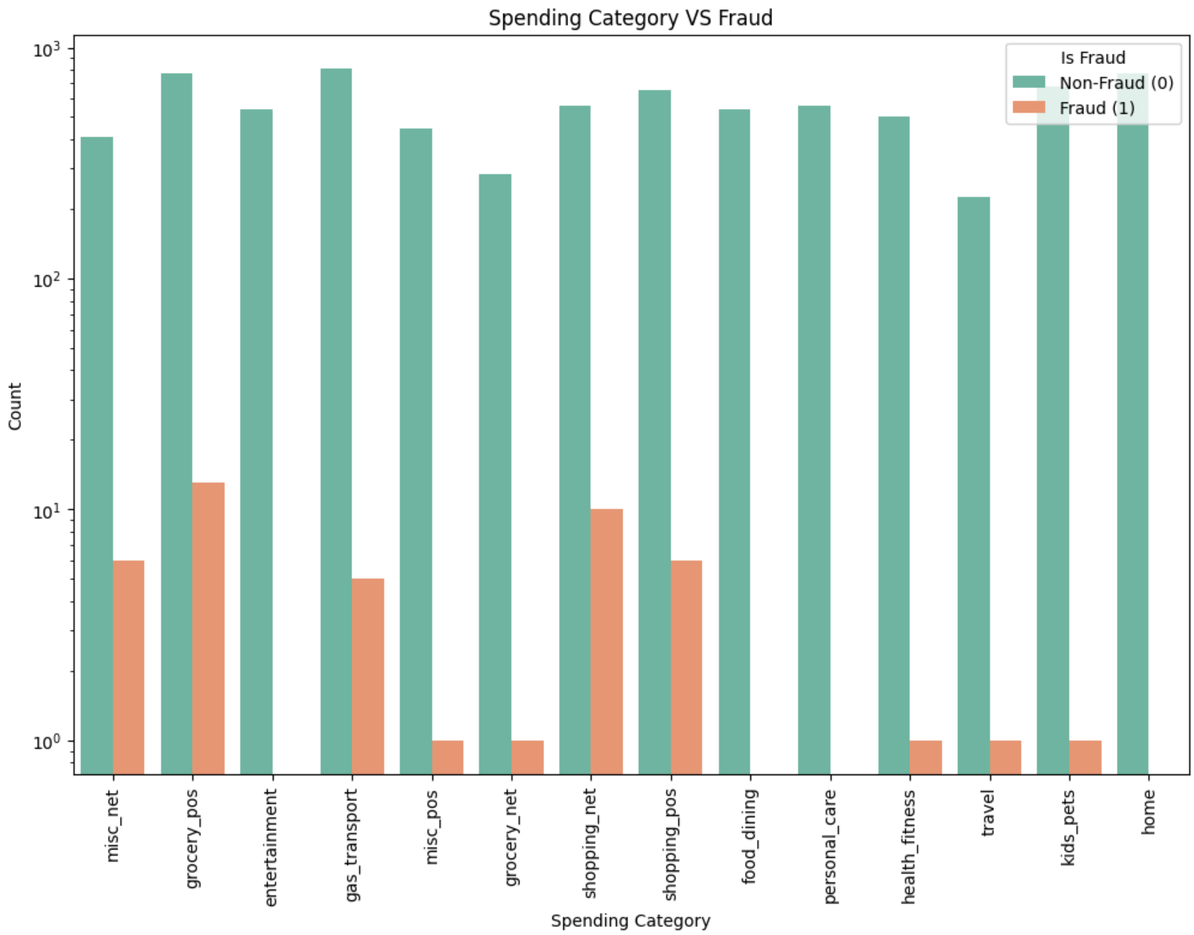


Figure 6

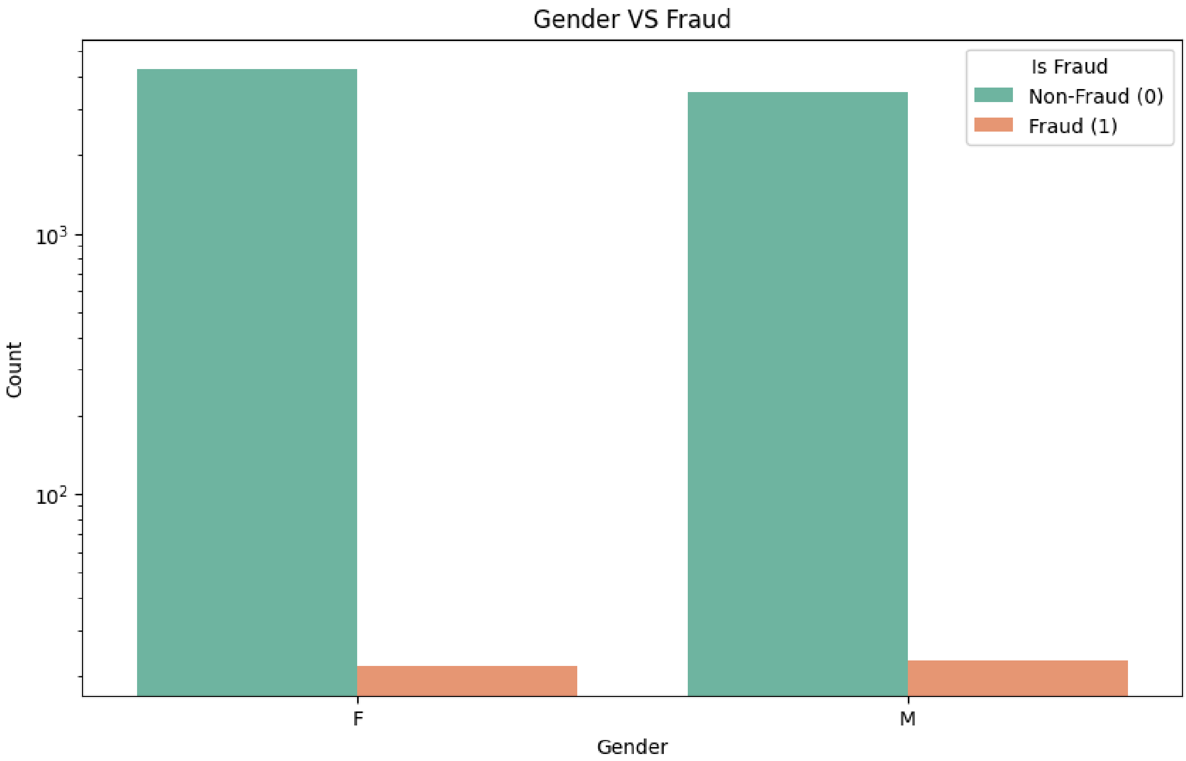


Figure 7

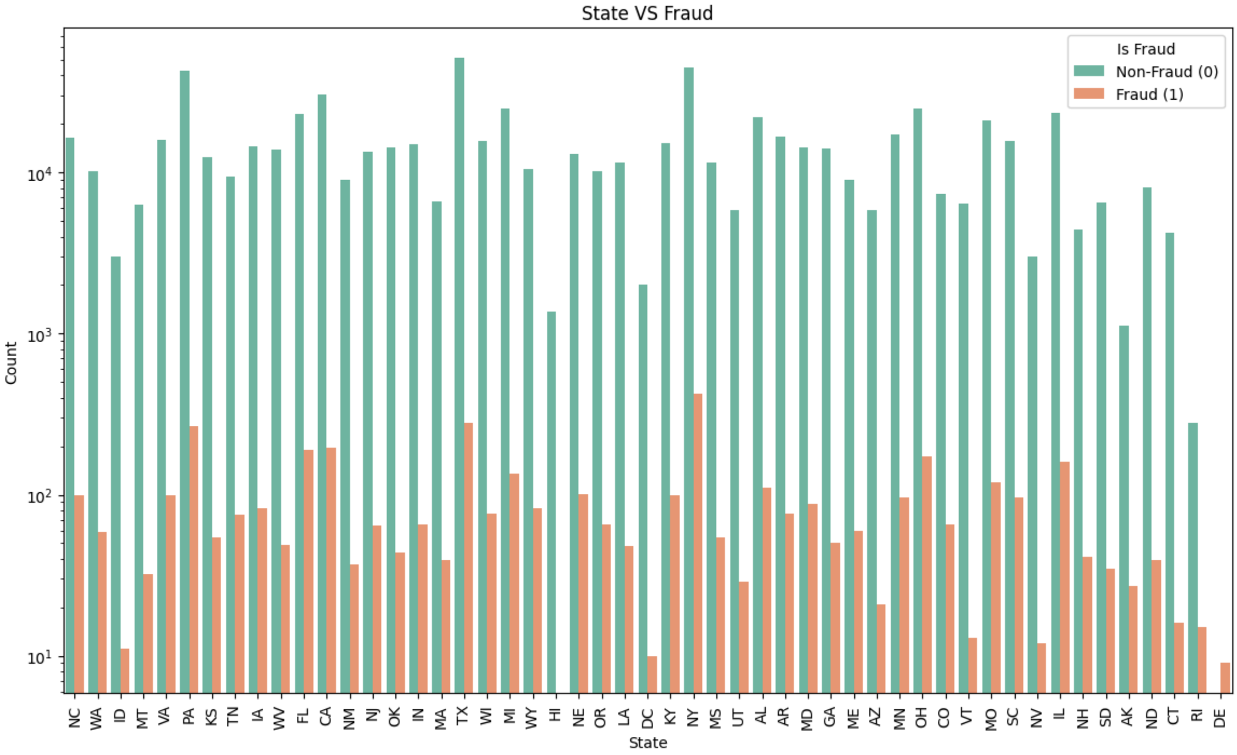


Figure 8

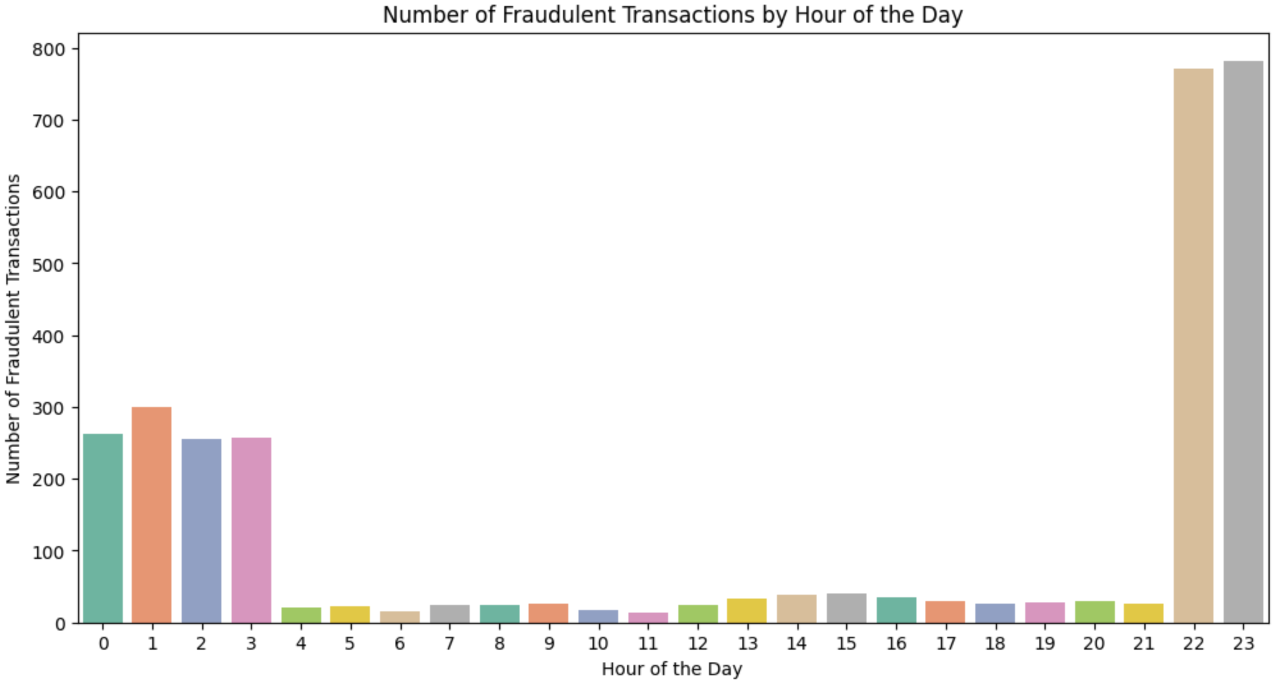
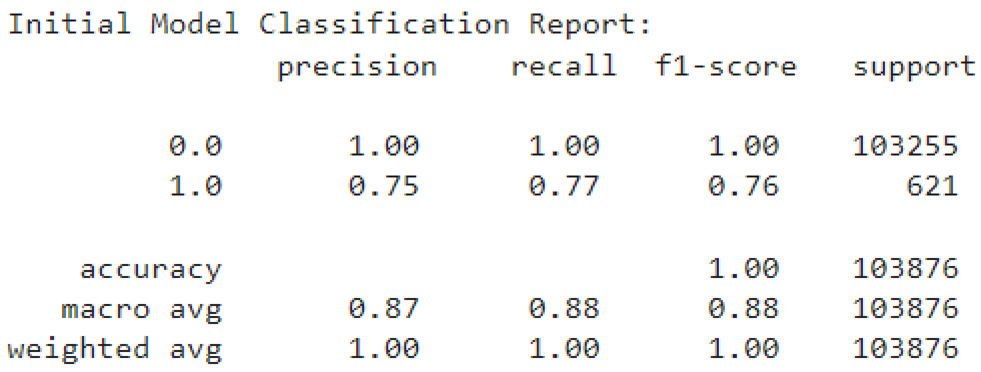


Figure 9

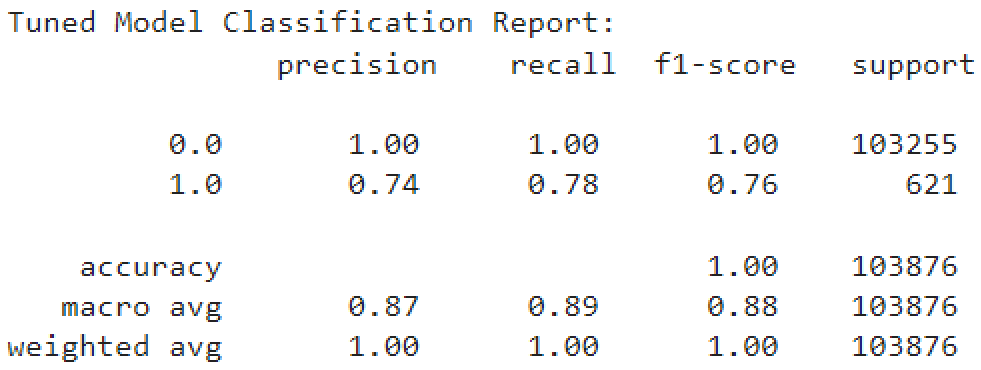


```
Initial Model Classification Report:
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00    103255
         1.0       0.75      0.77      0.76       621

    accuracy                           1.00    103876
   macro avg       0.87      0.88      0.88    103876
weighted avg       1.00      1.00      1.00    103876
```

Figure 10



```
Tuned Model Classification Report:
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00    103255
         1.0       0.74      0.78      0.76       621

    accuracy                           1.00    103876
   macro avg       0.87      0.89      0.88    103876
weighted avg       1.00      1.00      1.00    103876
```

Figure 11