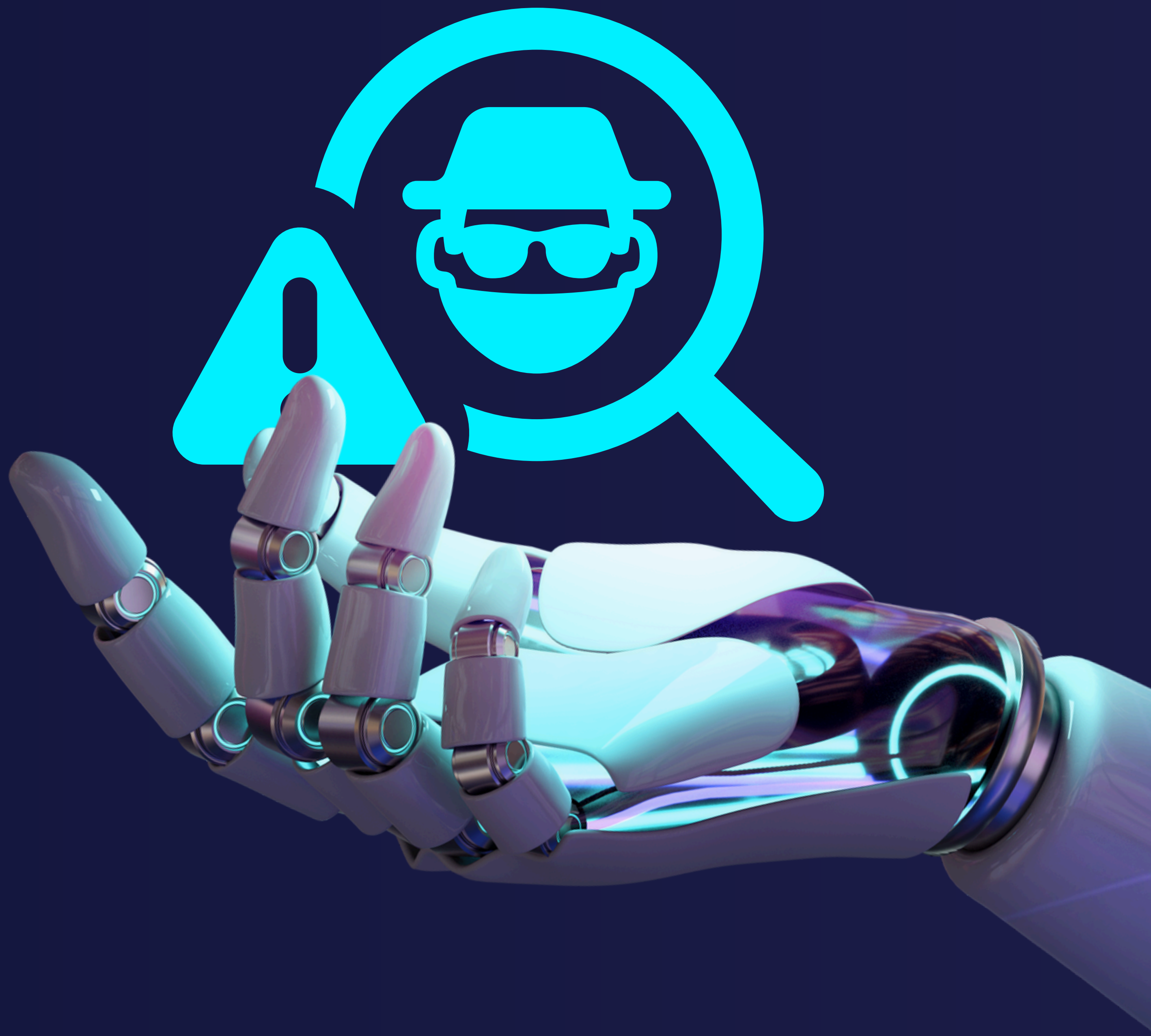
MACHINE LEARNING

FRAUD DETECTION

Random Forest in detecting credit card frauds

Group 01



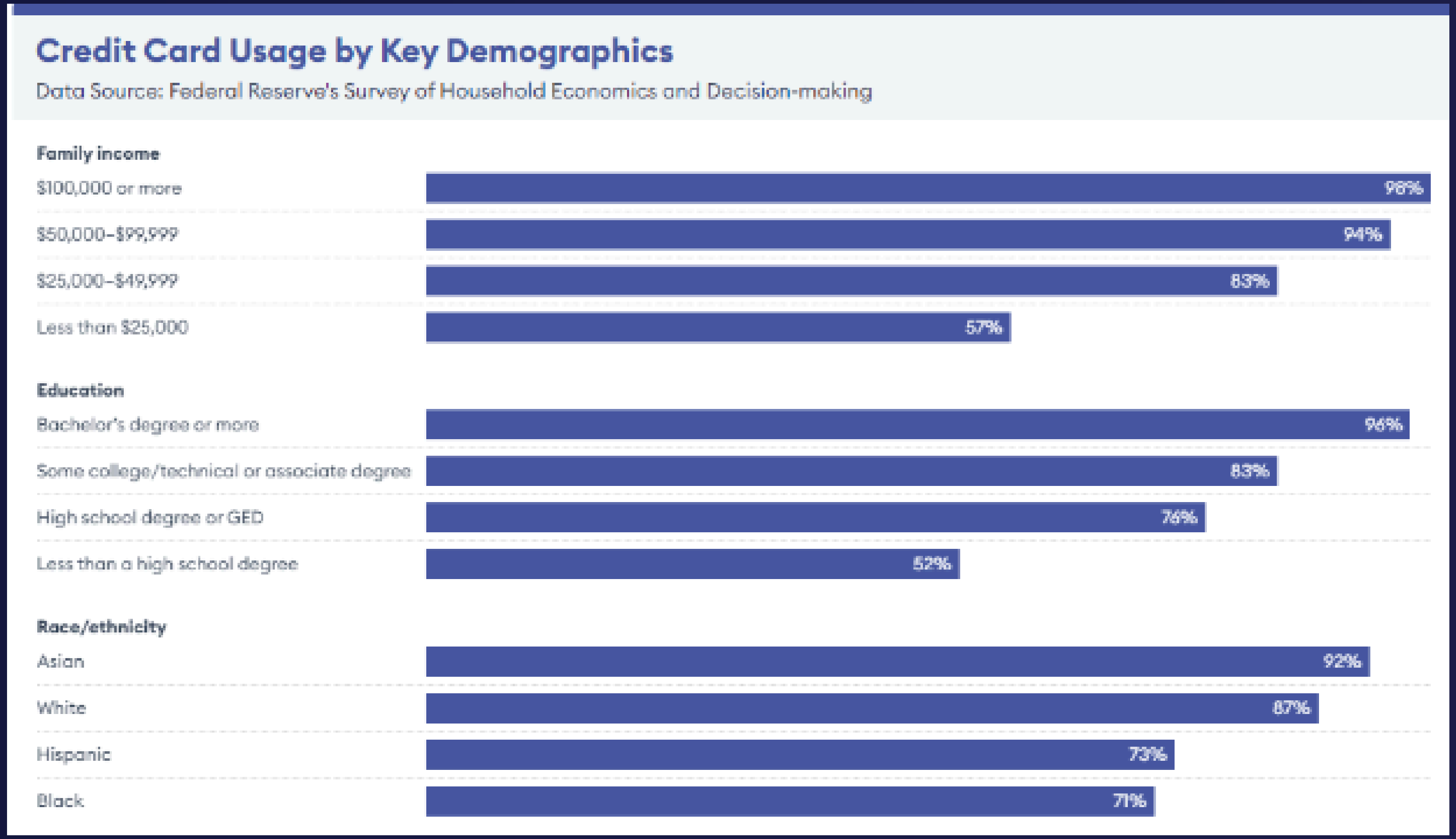
Tran Tue Nhi   Nguyen Trang Nhung   Pham Thi Ngoc Mai   Le Ba Hung

BUSINESS PROBLEM

The rise of credit cards

- Before 1996, physical transactions (bank checks or cash) were the norm.
- After 1996, with the rise of the internet, banks introduced the credit card.

=> This resulted in an **increase in online transactions and purchases** on a day-to-day basis, especially given the rise of e-commerce in recent years.

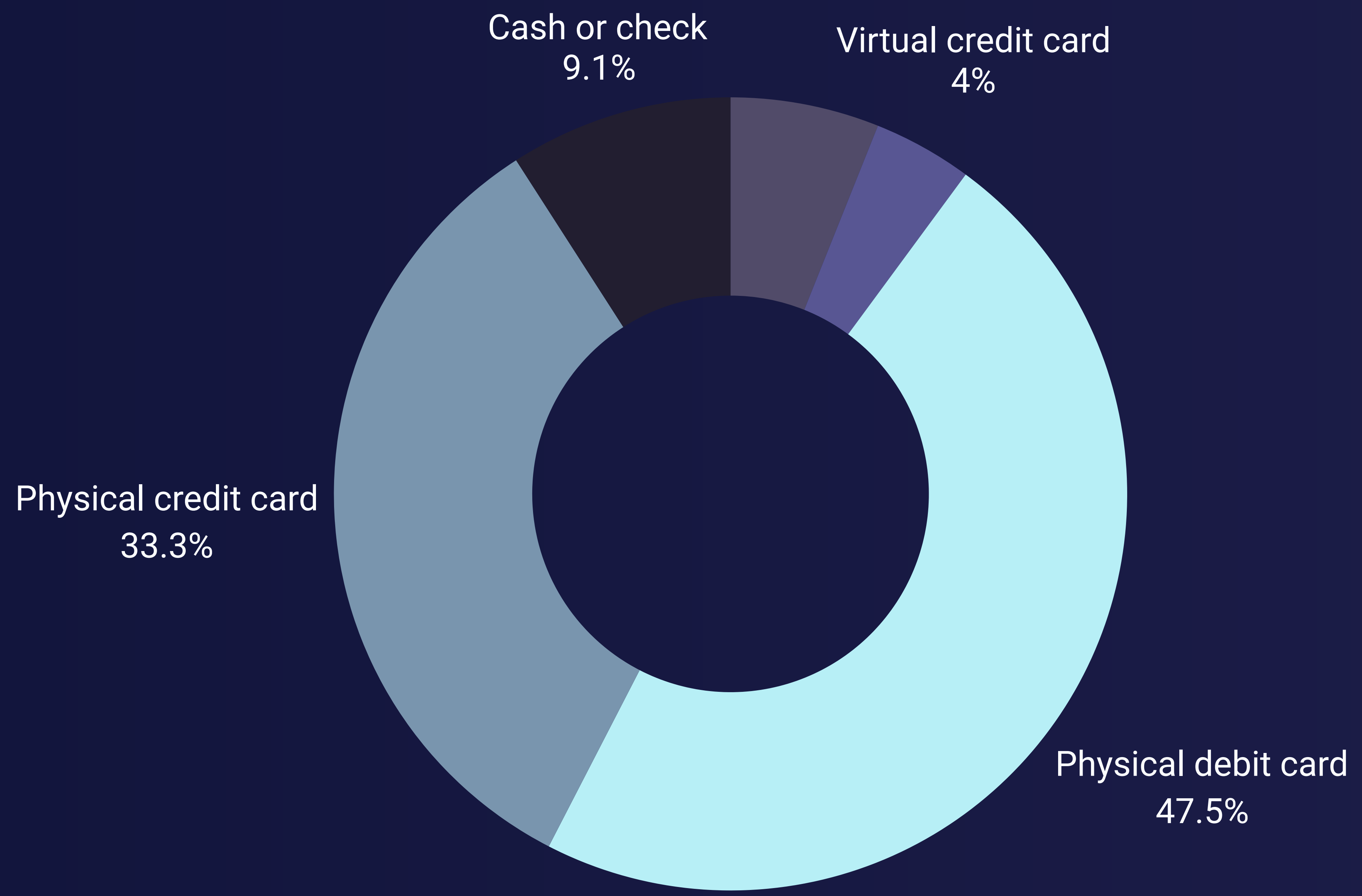


BUSINESS PROBLEM

Statistics on credit cards

- Market for credit card in the USA expected to worth more than 10.4 trillion USD in 2022
- Forbes research indicated in 2023, around 200 million Americans have at least 1 credit card.
- In 2023, credit cards and debit cards account for 90.9% of retail sales transactions.

=> Credit card market expected to expand at a CAGR of 2.84% during the forecast period of 2023-2028, reaching USD 12.6 trillion by 2028.



BUSINESS PROBLEM

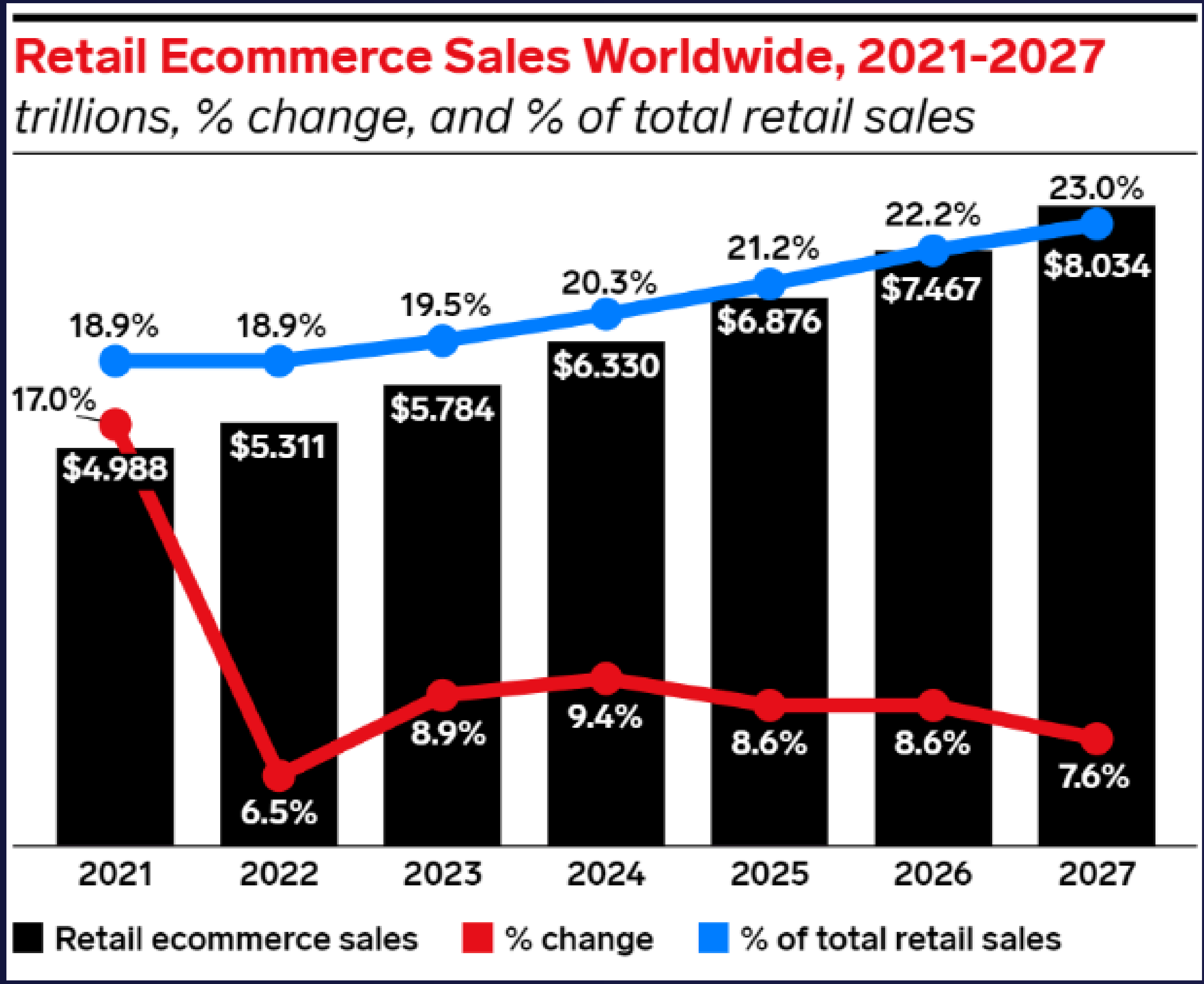
THE EMERGENCE OF CREDIT CARD FRAUD

01 According to Nielsen report, credit card loss total for 28.65 billion in 2019

02 U.S. alone accounts for over a third of these losses, with $11 billion in credit card fraud reported in 2020

03 The e-commerce sectors expected to have a CAGR of roughly 8.8% from 2023-2027

04 E-commerce expected to reach 30% of market share by 2027



SOLVING CREDIT CARD WITH RANDOM FORREST MODEL

Credit card fraud detection falls under binary classification, where transactions are categorized as either fraudulent or non-fraudulent.

01 Visa’s A.I. technology prevented $25 billion in fraud 2023

02 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.

03 Random Forest's ensemble approach helps mitigate this issue by leveraging multiple decision trees to enhance detection accuracy and robustness

MODEL BACKGROUND

RANDOM FORREST

- A **supervised** learning technique grounded in **ensemble learning** principles.
- 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 (ref 1). The outcome of random forest is based on the **aggregated decisions** across various decision trees.

Decision trees

- Decision trees are **fundamental** to random forest models.
- They are **flowchart-like** structures that use decision nodes to analyze data and make predictions.
- Each **decision node** tests a single attribute, guiding the model towards a class label.
- By iteratively evaluating decision nodes, the dataset is **segmented into more homogeneous subsets.**
- Decision trees aim to find **optimal splits** in the data to improve prediction accuracy.

+

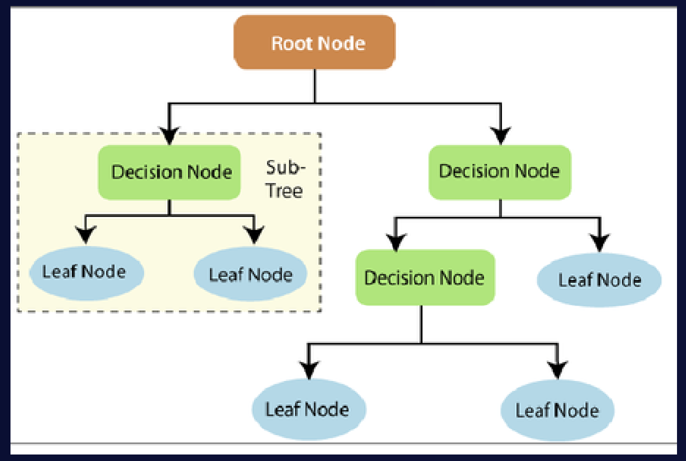
Ensemble methods

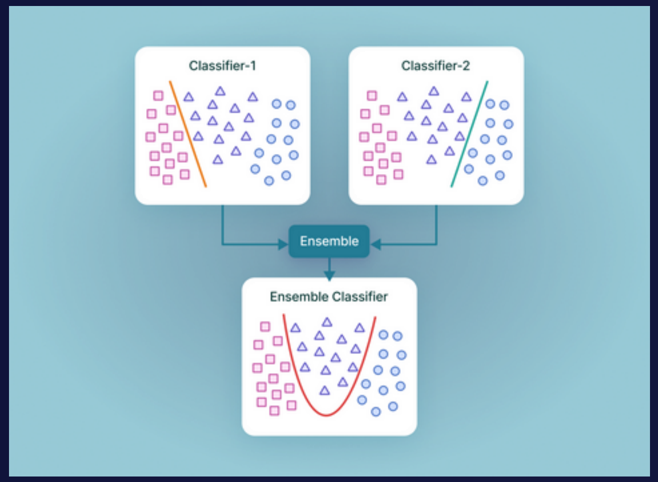
- Addresses the question of whether using **more models** can enhance performance in machine learning.
- It **combines outputs** from multiple models to achieve **higher accuracy** than any single model.
- The most popular ensemble method is called "**b**ootstrap **agg**regat**ing**" or "**bagging**".
- In random forests, each decision tree model **randomly samples a subset** of the training data **with replacement**.
- Each decision tree operates **independently**, and the final prediction is based on **aggregating the outputs** of all trees in the forest.

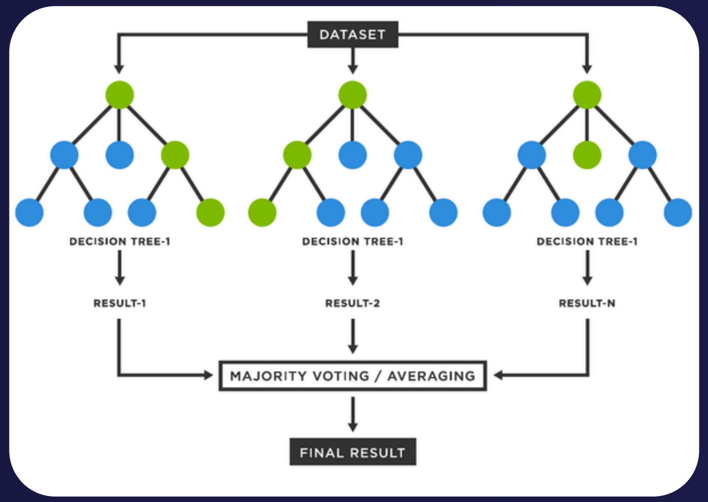
=

Random forest

- Employs **bagging** and **feature randomness** to create an uncorrelated ensemble of decision trees.
- **Feature randomness**, or "random subspace method," involves selecting a random subset of features to **minimize correlation** among trees. => Unlike decision trees, random forests **only use a subset of features** for each tree.
- Considering various sources of variability within the data helps **prevent overfitting, bias, and overall variance, improving prediction accuracy.**







MODEL PROPERTIES

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.

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'.

HYPERPARAMETER

For parameter tuning, the goal is to achieve **low correlation (ρ)** between trees while maintaining **reasonable strength** in tree construction.

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

MODEL ADVANTAGES

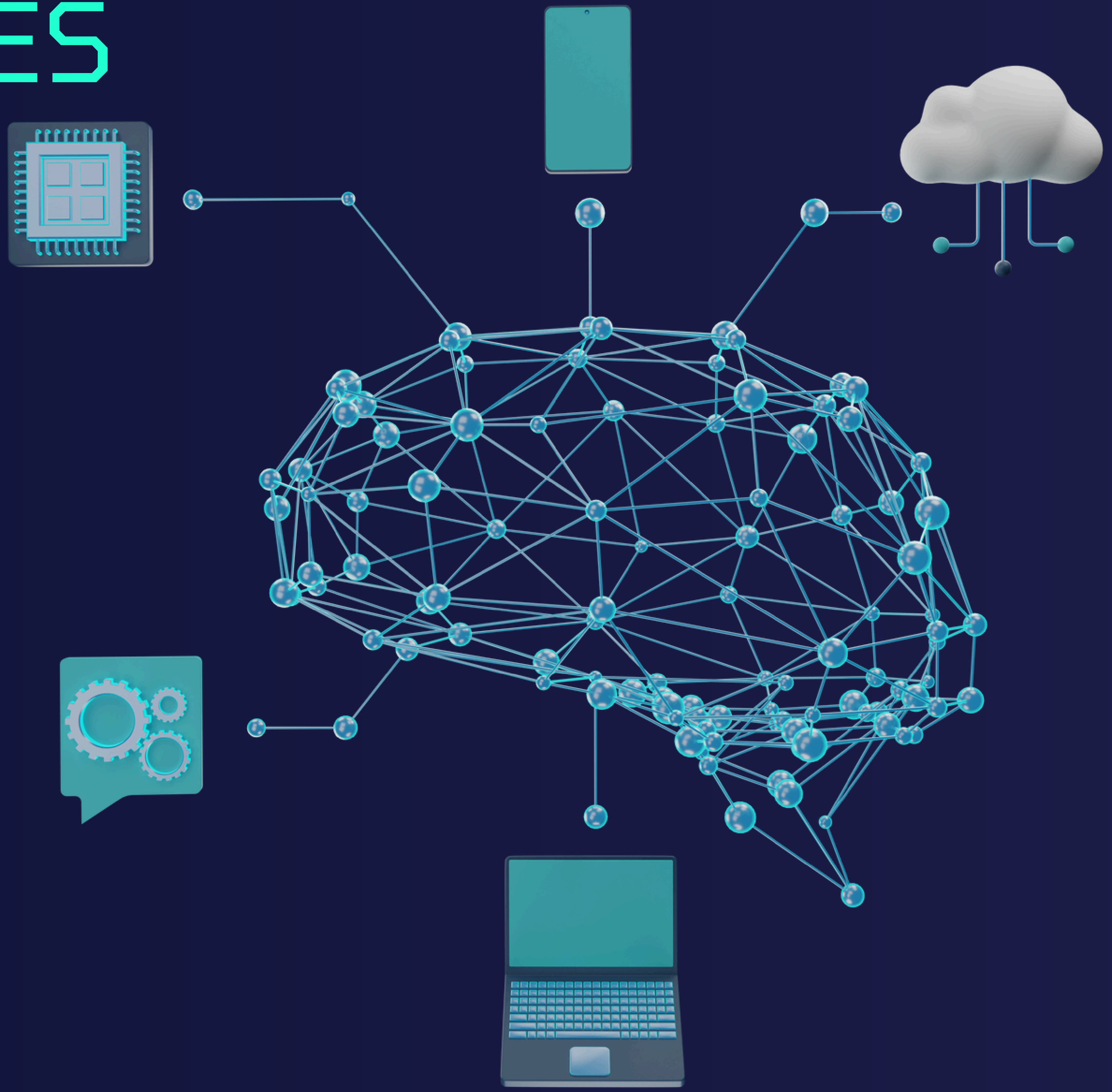
01 High complexity and flexibility

02 Reduced risk of overfitting and noise

03 Handling missing values

04 Feature importance and interpretability

05 Proven effectiveness in fraud detection



DATA DESCRIPTION

01 Data Overview and Sources

- Sourced from **Kaggle**
- from January 1, 2019, to December 31, 2020
- 1,000 customers and 800 merchants
- Highly **imbalanced dataset**

02 Types of Data

- **numerical** data
- **categorical** data
- **time series** data

03 Data Relevance

- Diverse features
- Identify **patterns** and **anomalies**
- **Historical data** for model learning
- Effective fraud detection with Random Forest

04 Compatibility with Random Forest

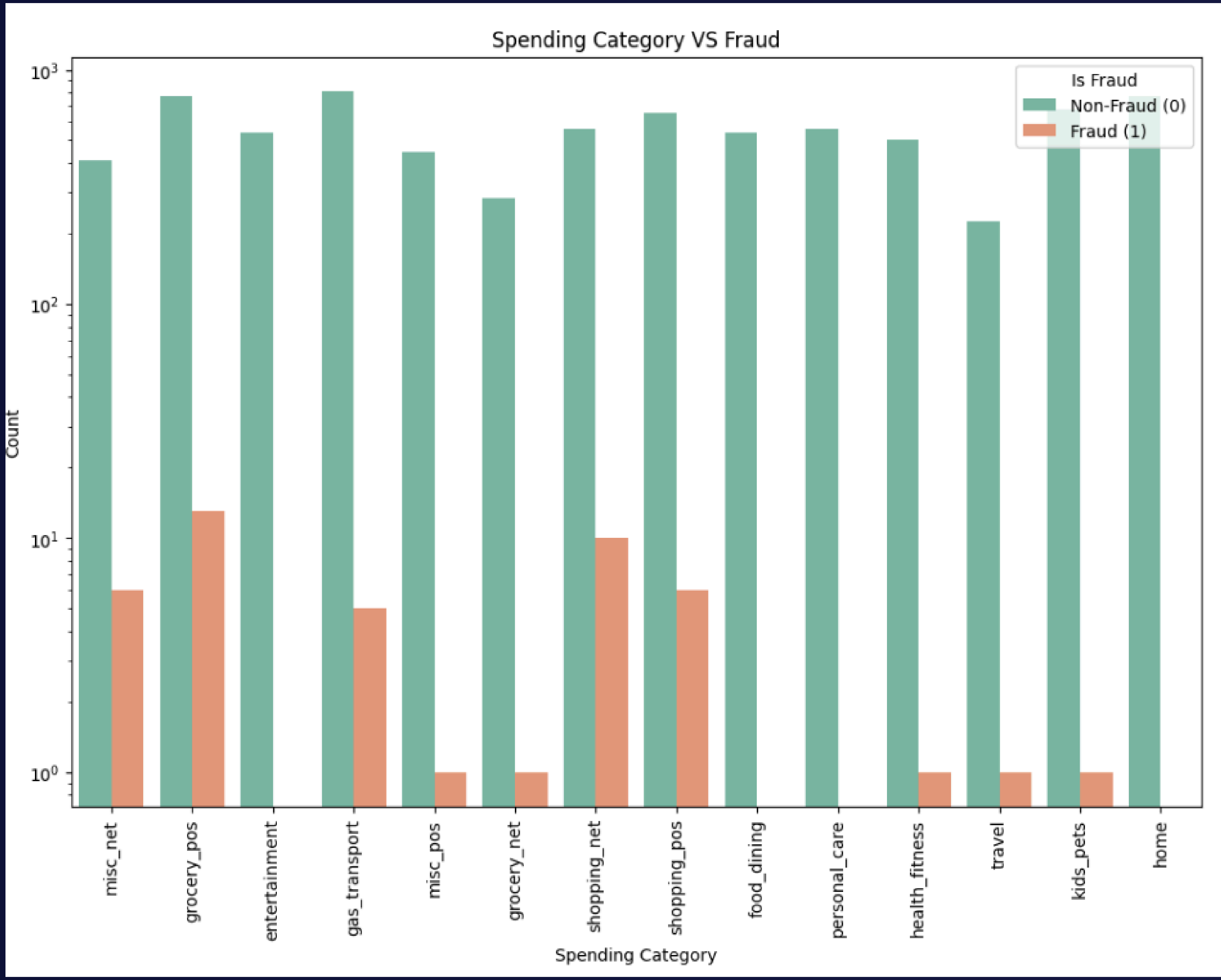
- Data **Cleaning**
- Normalize
- Encode
- Feature **Engineering**

STEP 1. DATA PREPROCESSING

a. Collect Data:
The data is downloaded from the Kaggle dataset. Then we load the data as two separate datasets, “train_data” and “test_data”.

b. Prepare the Data:
- **Clean data**:
  - Check for missing value
  - Address missing data by dropping null values
- **Normalize**: Scale the data to prevent bias towards higher magnitude features using StandardScaler.
- **Encode**: Transform categorical data into numeric formats through one-hot encoding (for categorical features) and standard scaler (for numerical features).
- **Feature Engineering**: Create new feature => New column “hour of day” after adding feature

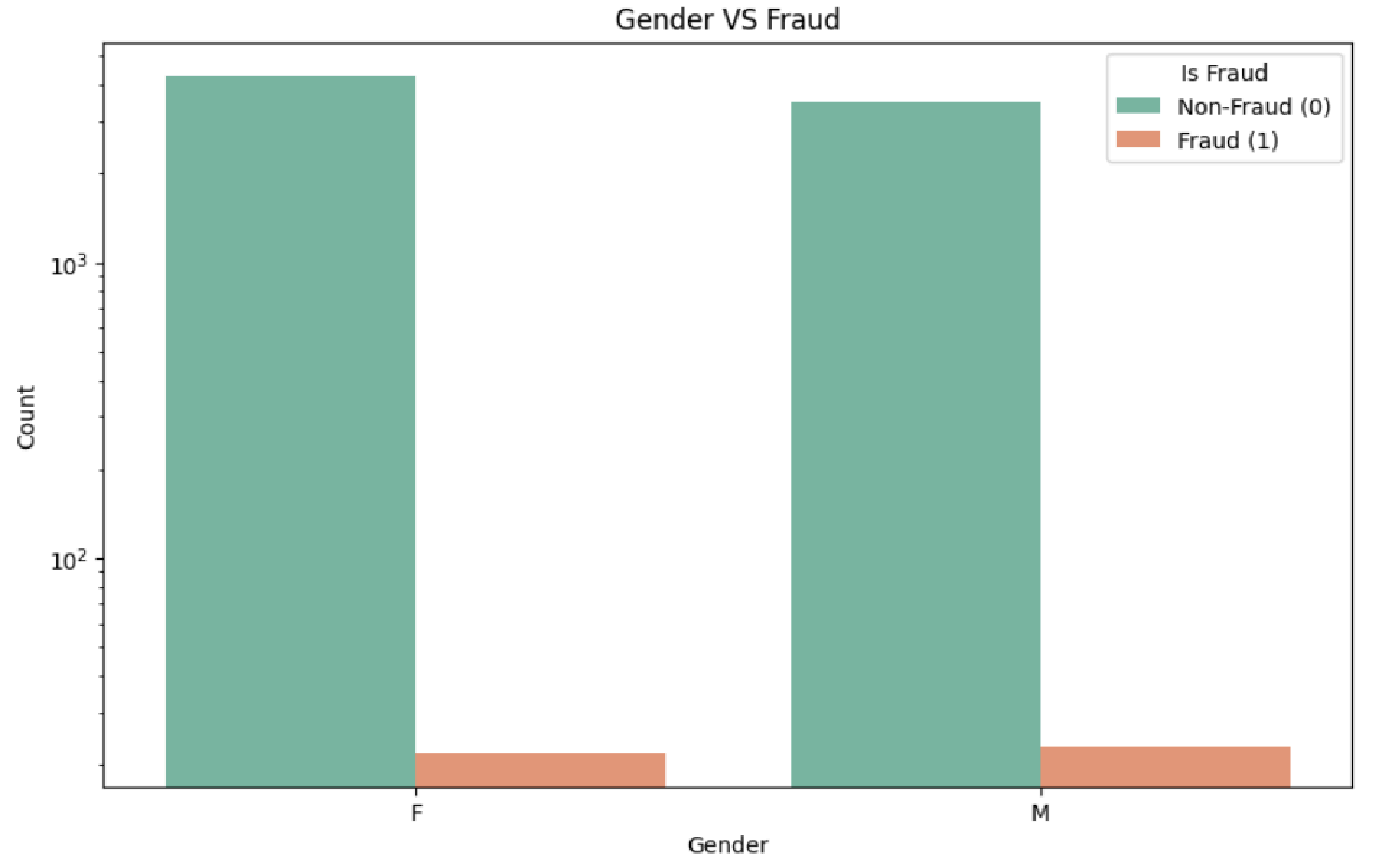
STEP 2. EXPLORATORY DATA ANALYSIS



Spending vs Fraud

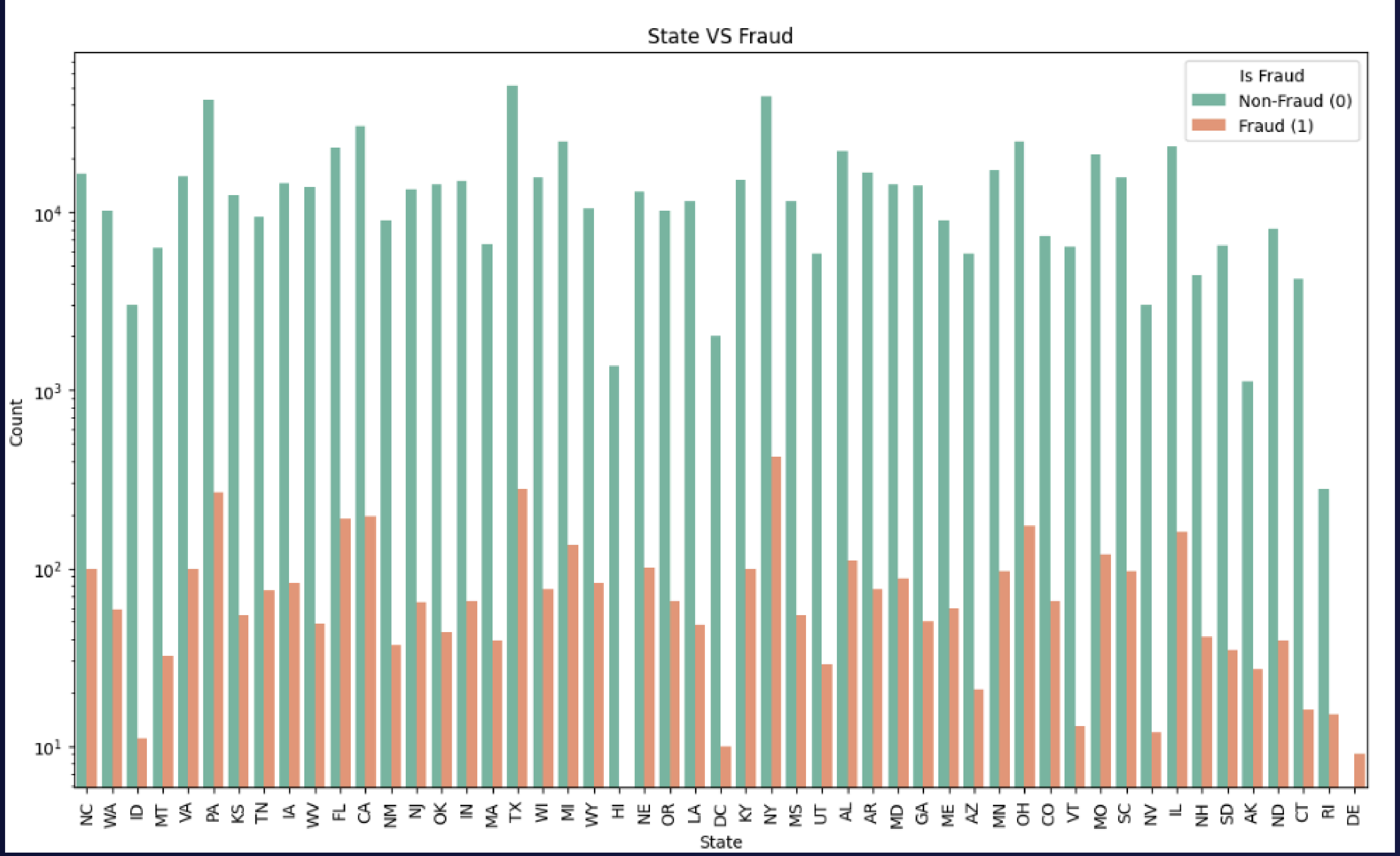
PIPELINE

STEP 2. EXPLORATORY DATA ANALYSIS



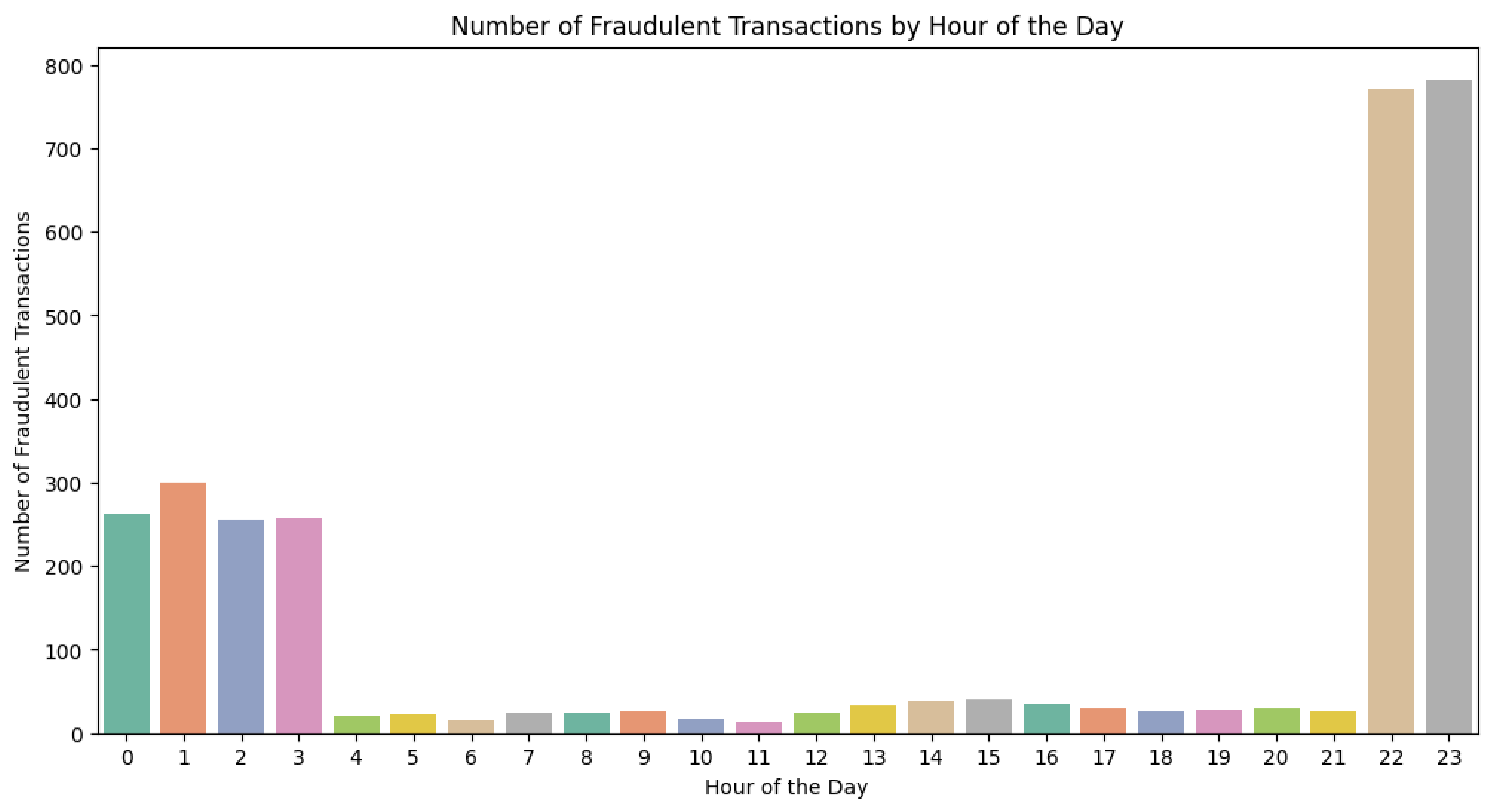
Gender vs Fraud

STEP 2. EXPLORATORY DATA ANALYSIS



State vs Fraud

STEP 2. EXPLORATORY DATA ANALYSIS



Hours of Day vs Fraud

STEP 3. TRAIN THE MODEL

- **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.

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

STEP 4. EVALUATE THE MODEL

- **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 & 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

Use the optimized model to predict and calculate evaluation metrics on the testing data again.

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

CONSTRAINTS

CONSTRAINTS

Data Quality

Interpretability and Compliance

Real-time Processing Speed

Integration with Existing Systems

Extra costs

THANK YOU

MACHINE LEARNING FINAL PRESENTATION