**Project Report**

**On**

**Credit Card Fraud Detection using Machine Learning and Data Science**



*Submitted*

*In partial fulfilment*

*For the award of the Degree of*

**PG-Diploma in Big Data Analytics**

**(C-DAC, ACTS (Pune))**

**Guided By: Submitted By:**

Mr. Prakash Sinha Ankita Patil (240340125010) Kartik Sapra (240340125026)

Shruti Gawande (240340125047)

Trupti Dhanawate (240340125052)

Yash Verma(240340125058)

**Centre for Development of Advanced Computing (C-DAC), ACTS (Pune- 411008)**

## Acknowledgement

This is to acknowledge our indebtedness to our Project Guide, **Mr. Prakash Sinha,** C-DAC ACTS, Pune for his constant guidance and helpful suggestion for preparing this project **Credit Card Fraud Detection using Machine Learning and Data Science.** We express our deep gratitude towards him for inspiration, personal involvement. His guidance and support helped us to overcome various obstacles and intricacies during the course of project work.

We take this opportunity to thank Head of the department **Mr. Gaur Sunder** for providing us such a great infrastructure and environment for our overall development.

We express sincere thanks to **Mrs. Namrata Ailawar** **(Process Owner)** for their kind cooperation and extendible support towards the completion of our project.

It is our great pleasure in expressing sincere and deep gratitude towards **Mrs. Risha P R (Program Head)** and **Ms. Pratiksha Gacche (Course Coordinator, PG-DBDA)** for their valuable guidance and constant support throughout this work and help to pursue additional studies.

Also, our warm thanks to **C-DAC ACTS Pune**, which provided us this opportunity to carry out, this prestigious Project and enhance our learning in various technical fields.

Ankita Patil (240340125010) Kartik Sapra (240340125026)

Shruti Gawande (240340125047)

Trupti Dhanawate (240340125052)

Yash Verma(240340125058)

## ABSTRACT

Abstract— It is vital that credit card companies are able to identify fraudulent credit card transactions so that customers are not charged for items that they did not purchase. Such problems can be tackled with Data Science and its importance, along with Machine Learning, cannot be overstated. This project intends to illustrate the modelling of a data set using machine learning with Credit Card Fraud Detection. The Credit Card Fraud Detection Problem includes modelling past credit card transactions with the data of the ones that turned out to be fraud. This model is then used to recognize whether a new transaction is fraudulent or not. Our objective here is to detect 100% of the fraudulent transactions while minimizing the incorrect fraud classifications. Credit Card Fraud Detection is a typical sample of classification. In this process, we have focused on analyzing and pre-processing data sets as well as the deployment of multiple anomaly detection algorithms such as Local Outlier Factor and Isolation Forest algorithm on the PCA transformed Credit Card Transaction data.

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **S. No** | **Title** | **Page No.** |
|  | **Front Page Acknowledgement Abstract**  **Table of Contents** | **I II III**  **IV** |
| **1**  **1.11.2** | **Introduction**  Introduction  Objective and Specifications | **01-02**  **01**  **02** |
| **2** | **Literature Review** | **03-04** |
| **3** | **Methodology/ Techniques** | **05-20** |
| **3.1** | Approach and Methodology/ Techniques | **05** |
| **3.2** | Data Analysis | **05** |
| **3.3**  **3.4**  **3.5** | Exploratory Data Analysis(EDA)  Data Visualization using Tableau  Model Description | **08**  **14**  **16** |
| **4** | **Implementation** | **20** |
| **4.1** | Implementation | **20** |
|  |  |  |
| **5** | **Results** | **21-23** |
| **5.1** | Results | **21** |
| **6** | **Conclusion** | **24** |
| **6.1** | Conclusion | **24** |
| **7** | **References** | **25** |
| **7.1** | References | **25** |

**Chapter 1**

**Introduction**

Detection of text regions either from handwritten or printed document images containing various non-textual information is a difficult task, and it can be more challenging to locate the position of the text regions when we deal with a doctor’s prescription.

**1.1 Introduction**

'Fraud' in credit card transactions is unauthorized and unwanted usage of an account by someone other than the owner of that account. Necessary prevention measures can be taken to stop this abuse and the behavior of such fraudulent practices can be studied to minimize it and protect against similar occurrences in the future. In other words, Credit Card Fraud can be defined as a case where a person uses someone else’s credit card for personal reasons while the owner and the card issuing authorities are unaware of the fact that the card is being used. Fraud detection involves monitoring the activities of populations of users in order to estimate, perceive or avoid objectionable behaviors, which consist of fraud, intrusion, and defaulting. This is a very relevant problem that demands the attention of communities such as machine learning and data science where the solution to this problem can be automated. This problem is particularly challenging from the perspective of learning, as it is characterized by various factors such as class imbalance. The number of valid transactions far outnumber fraudulent ones. Also, the transaction patterns often change their statistical properties over the course of time.

These are not the only challenges in the implementation of a real-world fraud detection system, however. In real world examples, the massive stream of payment requests is quickly scanned by automatic tools that determine which transactions to authorize.

Machine learning algorithms are employed to analyze all the authorized transactions and report the suspicious ones. These reports are investigated by professionals who contact the cardholders to confirm if the transaction was genuine or fraudulent. The investigators provide a feedback to the automated system which is used to train and update the algorithm to eventually improve the fraud-detection performance over time.

Fraud detection methods are continuously developed to defend criminals in adapting to their fraudulent strategies.

These frauds are classified as:

• Credit Card Frauds: Online and Offline

• Card Theft

• Account Bankruptcy

• Device Intrusion

• Application Fraud

• Counterfeit Card

• Telecommunication Fraud

Some of the currently used approaches to detection of such fraud are:

• Artificial Neural Network

• Fuzzy Logic

• Genetic Algorithm

• Logistic Regression

• Decision tree

• Support Vector Machines

• Bayesian Networks

• Hidden Markov Model

• K-Nearest Neighbor

**1.2 Objective**

The main aim of this project is the detection of credit card fraudulent transactions, as it’s important to figure out the fraudulent transactions so that customers don’t get charged for the purchase of products that they didn’t buy. The detection of the credit card fraudulent transactions will be performed with multiple ML techniques then a comparison will be made between the outcomes and results of each technique to find the best and most suited model in the detection of the credit card transaction that are fraudulent, graphs and numbers will be provided as well. In addition, exploring previous literatures and different techniques used to distinguish the fraud within a dataset.

**Chapter 2**

**LITERATURE REVIEW**

It is essential for credit card companies to establish credit card transactions that fraudulent from transactions that are non-fraudulent, so that their customers’ accounts won’t get affected and charged for products that the customers didn’t buy (Maniraj et al., 2019). There are many financial Companies and institutions that lose massive amounts of money because of fraud and fraudsters that are seeking different approaches continuously to violate the rules and commit illegal actions; therefore, systems of fraud detection are essential for all banks that issue credit cards to decrease their losses (Zareapoor et al., 2012). There are multiple methods used to detect fraudulent behaviors such as Neural Network (NN), Decision Trees, K-Nearest Neighbor algorithms, and Support Vector Machines (SVM). Those ML methods can either be applied independently or can be used collectively with the addition of ensemble or meta-learning techniques to develop classifiers (Zareapoor et al., 2012).

Zareapoor and his research team used multiple techniques to determine the best performing model in detecting fraudulent transactions, which was established using the accuracy of the model, the speed in detecting and the cost. The models used were Neural Network, Bayesian Network, SVM, KNN and more. The comparison table provided in the research paper showed that Bayesian Network was very fast in finding the transactions that are fraudulent, with high accuracy. The NN performed well as well as the detection was fast, with a medium accuracy. KNN’s speed was good with a medium accuracy, and finally SVM scored one of the lower scores, as the speed was low, and the accuracy was medium. As for the cost All models built were expansive (Zareapoor et al., 2012).

The model used by Alenzi and Aljehane to detect fraud in credit cards was Logistic Regression, their model scored 97.2% in accuracy, 97% sensitivity and 2.8% Error Rate. A comparison was performed between their model and two other classifier which are 5 Voting Classifier and KNN. VC scored 90% in accuracy, 88% sensitivity and 10% error rate, as for KNN where k = 1:10, the accuracy of the model was 93%, the sensitivity 94% and 7% for the error rate (Alenzi & Aljehane, 2020).

The classification approach used by Dheepa and Dhanapal was the behavior-based classification approach, by using Support Vector Machine, where the behavioral patterns of the customers were analyzed to distinguish credit card fraud, such as the amount, date, time, place, and frequency of card usage. The accuracy achieved by their approach was more than 80% (Dheepa & Dhanapal, 2012).

Gupta’s team worked on implementing an automated model that uses various ML techniques to detect fraudulent instances that are related economically to users but is specializing more in credit card transactions, according to Gupta and his team Out of all the techniques that they used Naïve Bayes had an outstanding performance in distinguishing fraudulent transactions as the accuracy of it was 80.4% and the area under the curve is 96.3% (Gupta et al., 2021).

**Chapter 3**

**Methodology and Techniques**

**3.1 Dataset**

The dataset was retrieved from an open-source website, Kaggle.com. The Credit Card Transactions Dataset offers comprehensive records of credit card transactions, featuring essential details such as transaction times and amounts. This extensive dataset includes over 1.25 million rows, capturing a wide range of financial activities to support in-depth analysis.

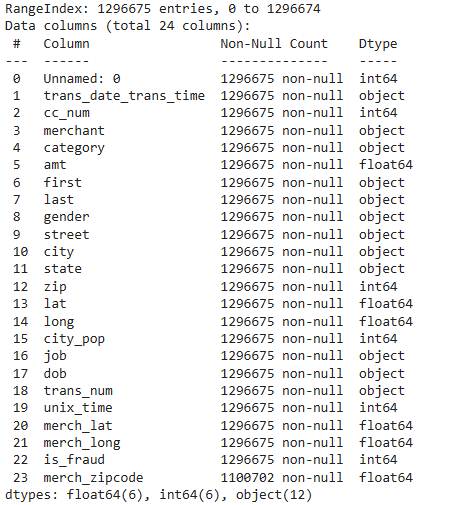
In addition to transaction specifics, the dataset provides valuable information about both the personal details of cardholders and the merchants where transactions occurred. This data is crucial for understanding spending patterns, conducting fraud detection, and developing targeted marketing strategies.

The dataset's broad scope and detailed records make it an invaluable resource for various applications, from predictive modelling and behavioural analysis to geospatial studies and anomaly detection. It is designed to facilitate advanced financial analyses and help users derive actionable insights from transactional data.

**3.2 Data Analysis:**

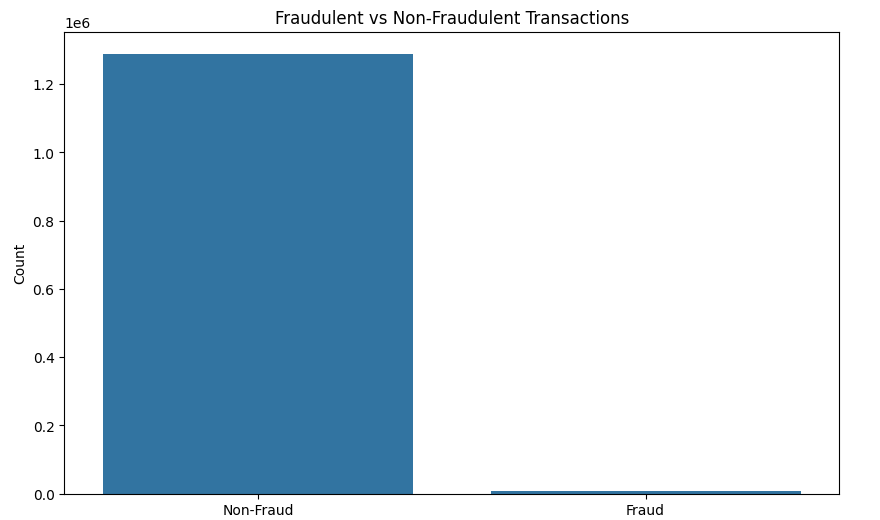
**3.2.1 Data Preparation**

The figure Fig.1 below shows the structure of the dataset where all attributes are shown, with their type. ‘is\_fraud’, this is the target variable indicating fraud status, identify the 0 as Not Fraud and the 1 as Fraud to ease the process of creating the model and obtain visualizations.



**Fig.1 Dataset Structure**

The second figure shows the distribution of the variable “is\_fraud” represents the non-fraudulent transactions (12,89,169), and the fraudulent transactions (7,506).



**Fig.2 Fraudulent transactions**

**3.2.2 Data Preprocessing**

**Date and Time Extraction:**  
The trans\_date\_trans\_time column, originally containing both the transaction date and time, was split into two separate columns: trans\_date and trans\_time. This separation facilitates more granular analysis of transactions based on date and time.

**Date of Birth Conversion and Age Calculation:**  
The dob column, which initially contained the date of birth of the customers in string format, was converted to a datetime format. From this, the year of birth was extracted, and the current year was used to calculate the exact age of each customer.

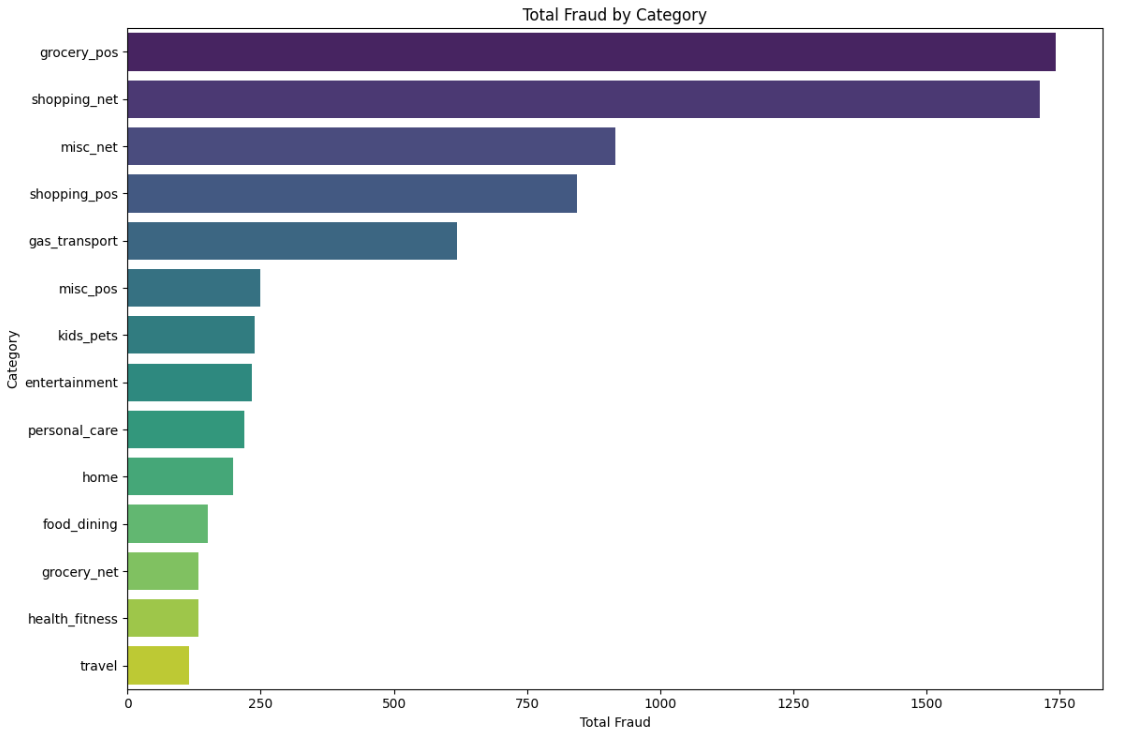
**Age Bucketing:**  
An age\_bucket column was created to categorize customers into age groups. The age ranges were defined as:

* Less than 18 years
* 18 to 29 years
* 30 to 59 years
* 60 to 80 years

**3.3 Exploratory Data Analysis (EDA)**

To gain insights from the dataset, several visualizations were created, focusing on key aspects of the data:

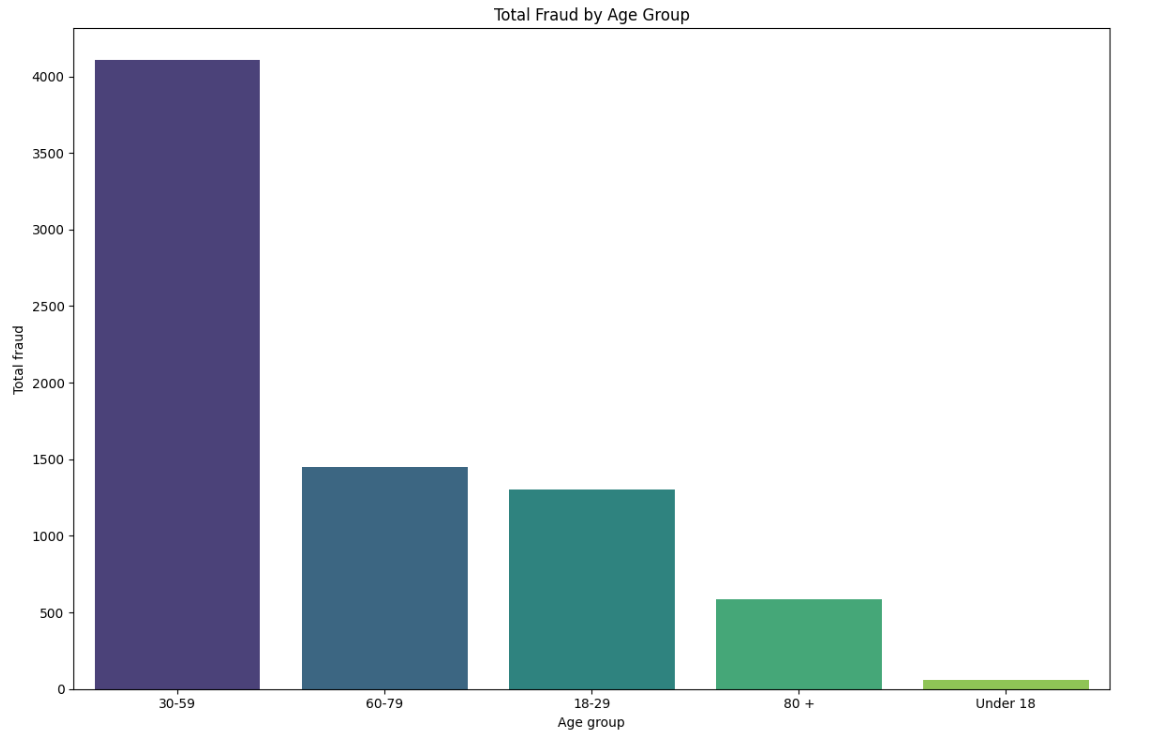
1. **Total Fraud by Category**:  
    A bar chart was employed to highlight the total amount of fraud detected in each category. This visualization was crucial for identifying which categories were more prone to fraudulent activities.

**Fig.3 Number of Farud transactions across categories**

1. **Fraud by Gender**:  
    A comparison chart was created to examine the relationship between gender and the occurrence of fraud. This allowed for an analysis of any gender-specific trends in fraudulent transactions.

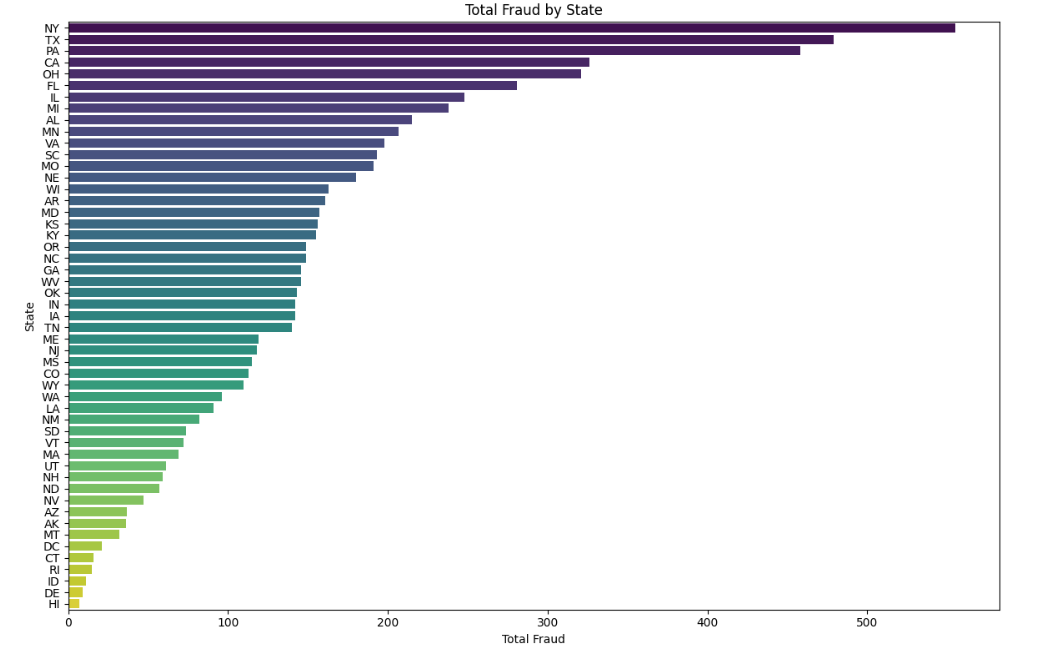
**Fig.4 Number of Farud transactions by Gender**

1. **Number of Fraud Transactions per Age Group**:  
   A bar chart was used to display the number of fraud transactions across different age groups. This visualization helped in understanding which age groups were more inactive in transactions.

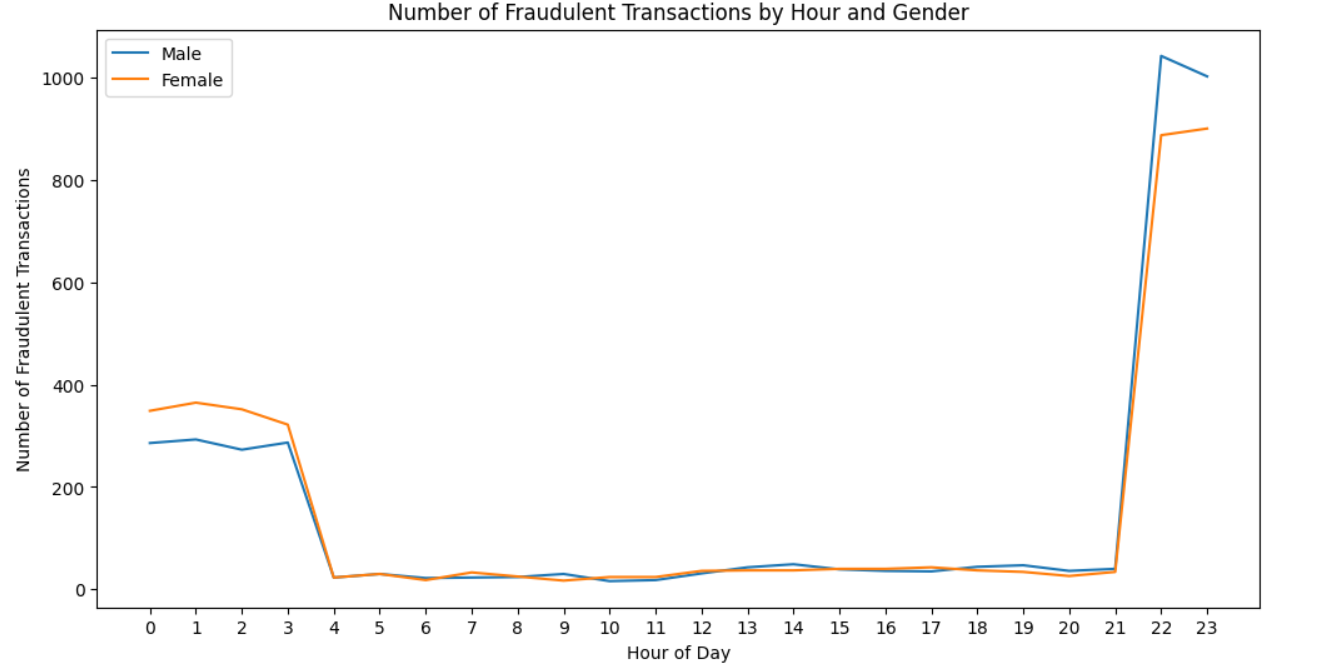
**Fig.5 Number of Farud transactions per Age Group**

1. **Total Fraud by State**:

A bar chart was used to show the total fraud amount across different states. This was useful for identifying regions with higher fraud incidents.

 **Fig.6 Number of Farud transactions by state**

1. **Fraud by Time**:  
   A line graph was created to explore how fraud occurrences varied over different times of the day grouped by gender. This time-series analysis provided insights into peak hours for fraudulent activities.

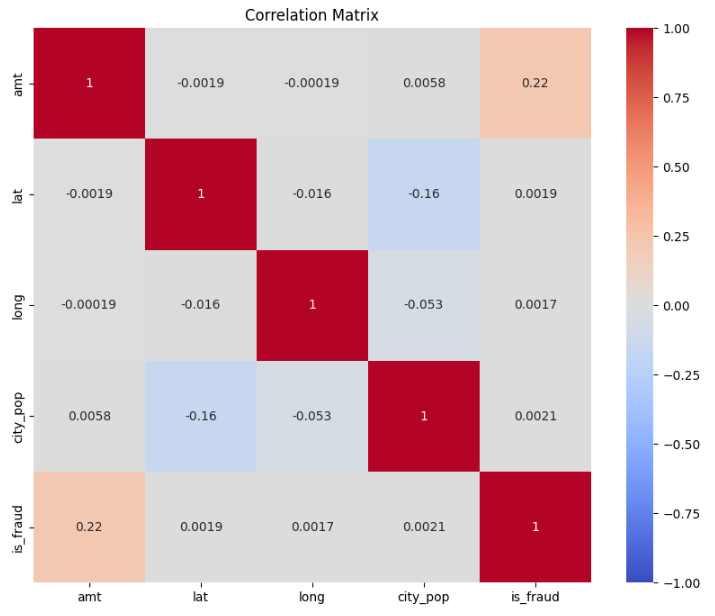
 **Fig.7 Number of Farud transactions by Hour and Gender**

1. **Correlation Matrix**:

The correlation matrix is a key tool for identifying relationships between features. It helps you see which variables are strongly correlated, which can inform feature selection, engineering, and modeling decisions.

To explore the relationships between key features in the dataset, a correlation matrix was computed for the variables amount, longitude, latitude, and city population. The purpose of this analysis was to identify any strong linear relationships between these variables that could impact further analysis and modeling.

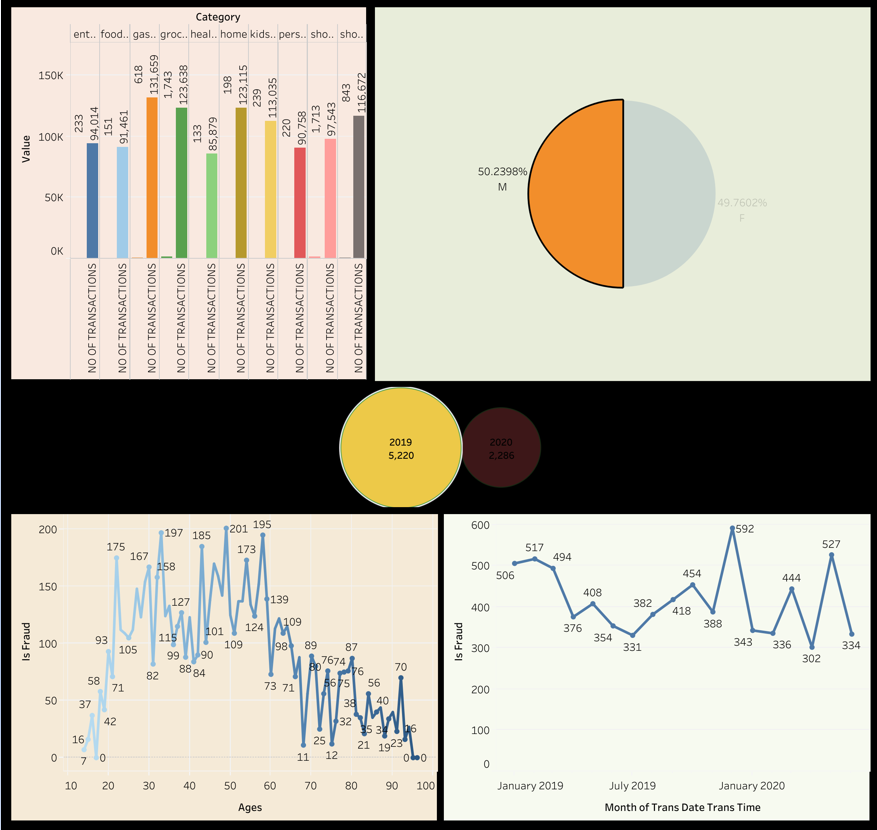
* **Amount**: This variable represents the transaction amount. Understanding its correlation with other features is crucial to determine if geographic or demographic factors influence transaction values.
* **Longitude and Latitude**: These geographical coordinates were included to assess any spatial relationships. High correlation between these coordinates and other features, such as city population, could indicate regional patterns in transaction behavior.
* **City Population**: This feature represents the population size of the city where the transaction occurred. Examining its correlation with transaction amount and geographic coordinates helps identify whether population density has a significant impact on transaction values.



**Fig.8 Correlation matrix**

**3.4 Data Visualization using Tableau-**

**1.**

****

**Fig.9 Dashboard 1**

* **Transaction Categories:** Most transactions are concentrated in essential categories like food, gas, and groceries, with a smaller volume in shopping.
* **Gender Distribution:** Transaction activity is almost equally divided between males and females.
* **Yearly Trends:** At the start of the year 2020 there was a significant reduction in the number of transactions as compared to 2019.
* **Fraud Analysis:** Fraudulent activities are more common in middle-aged groups, particularly between 20 and 50 years. The incidence of fraud fluctuates over time but shows a slight decrease toward the end of the period observed.

**2.**

****

**Fig.10 Dashboard 2**

 **Fraud by Category**: Fraud is most prevalent in "grocery\_pos" and "shopping\_net" categories, which might require enhanced monitoring and security measures.

 **Fraud by Job**: Certain professions, particularly "Materials engineer" and "Trading standards officer," show a higher likelihood of fraud involvement, which could inform targeted fraud prevention efforts.

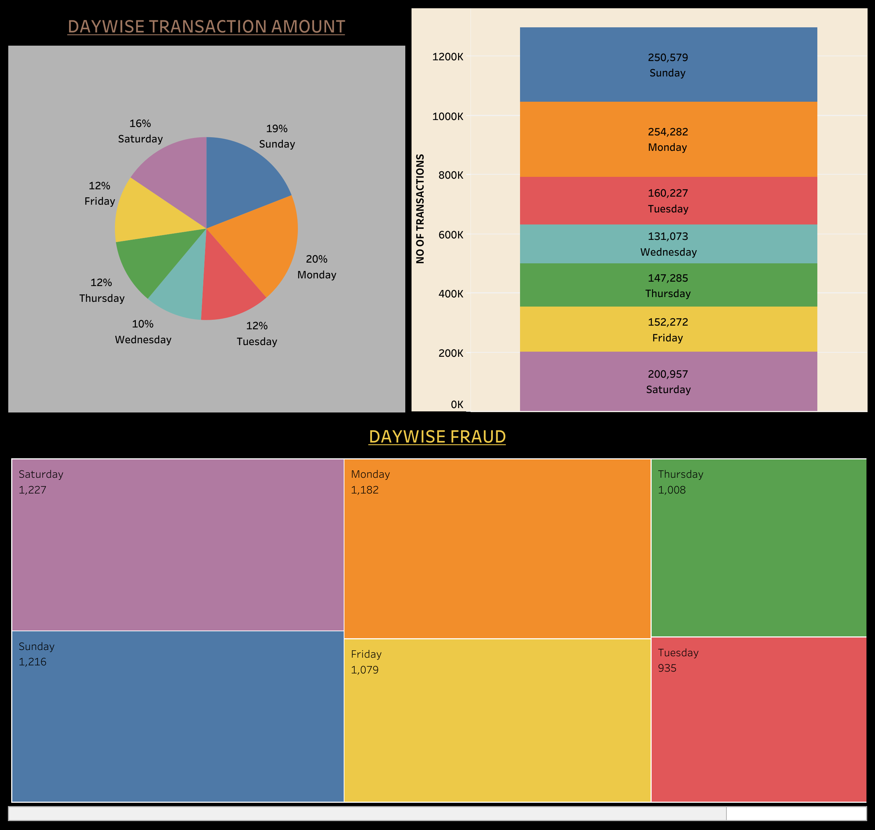
 **Fraudulent Merchants**: There are specific merchants that repeatedly appear in fraud cases, indicating the need for thorough scrutiny and potential blacklisting.

 **Credit Card Fraud**: A few credit card numbers are associated with multiple fraud cases, suggesting a need for heightened security protocols for those accounts.

 **Yearly Trend**: The reduction in fraud cases from 2019 to 2020 could indicate the effectiveness of fraud prevention strategies or other external factors affecting transaction behavior.

 **Gender Distribution**: Fraud activity is almost equally distributed between males and females, suggesting that gender-specific strategies may not be necessary.

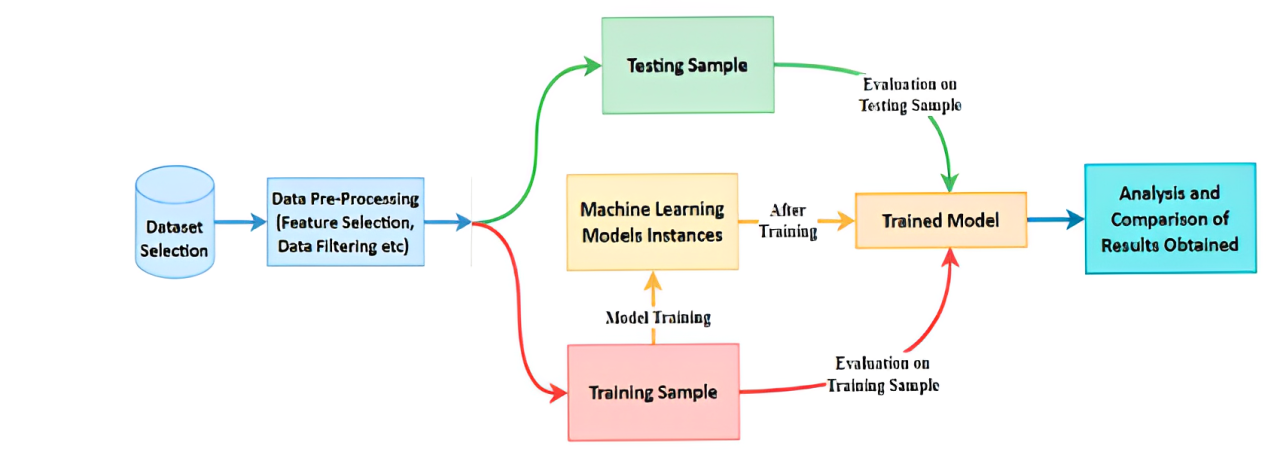
**3.**

 **Fig.11 Dashboard 3**

The dashboard reveals that Monday and Sunday are the busiest days for transactions, while Saturday experiences the highest number of fraud incidents. Fraud is more common on weekends, aligning with increased transaction activity. Mid-week sees fewer transactions and lower fraud cases, particularly on Wednesday and Tuesday. Strengthening fraud prevention on high-transaction days, especially over the weekend, could be beneficial.

**3.5 Model Description**

**Workflow Chart-**



**Preprocessing-**

Preprocessing is a crucial step to ensure that the data is clean, relevant and ready

For modeling. The preprocessing phase can be considered as the first stage of model building. The main goal of this step is to modify the dataset in such a way that will make it easier and faster for the different algorithms to build a model from it.

1. **Understanding the Dataset-**

* **Feature and Labels :** Identifying the features (input variables) and labels (output/target variable)
* **Data Types :** Checking the data types, is it numerical or categorical, of each feature.
* **Class Imbalance :** In this fraud detection dataset, the number of fraud cases are much smaller than non-fraud cases, this leads to class imbalance

1. **Handling Missing Values-**

* **Identify Missing Values :** “isnull()” funtion is used to check the missing values

Here “merch\_zipcode” column contains some null values, that null values are filled with “-1” by using “fillna()” function in python.

1. **Feature scaling-**

Here “get\_dummies()” function is used to scale the categotical features

1. **Feature Engineering-**

* **Age Calculation:** The 'Age' feature was created from the 'Date of Birth' column. We calculated age by subtracting the year of birth from the current year and adjusting for whether the birthday had occurred yet in the current year. This transformation provides a numerical representation of age that can be used in the model.

1. **Dealing with Categorical Data-**

* **One-Hot encoding :** In this dataset One hot encoding is used to create binary columns for each category.

1. **Outlier Detection-**

* **Identify Outliers :** Outliers are visualized using box plot

1. **Splitting the Dataset-**

* **Trainig and Testing Split :** Split the data into training and testing sets
* **Cross Validation :** k-fold cross-validation is used to ensure that the model performs well across different subsets of data.

**Model Classifiers -**

1. **Logistic Regression:**

* **Overview:** Logistic Regression is a linear model used for binary classification tasks. It estimates the probability of a binary outcome based on one or more predictor variables.
* **Key Characteristics:** The model uses the logistic function (sigmoid) to map predicted values to probabilities. It is particularly effective when there is a linear relationship between the features and the outcome.
* **Parameters:** We used lbfgs, sag, saga solvers to tune the model. The regularization parameter [C] was set to value using logspace function of numpy (np.logspace(-3, 3, 7)), and the penalty used was ridge regression [l2].

**2. k-Nearest Neighbors (KNN) Classifier:**

* **Overview:** KNN is a non-parametric, instance-based learning algorithm that classifies a data point based on the majority class among its k nearest neighbors in the feature space.
* **Key Characteristics:** It does not assume any underlying distribution and is simple to implement. The model's performance can be sensitive to the choice of k and the distance metric.
* **Parameters:** We selected k = [3, 5, 7, 9], and used [ Euclidean distance, Manhattan distance] as the distance metric. The choice of k was determined based on cross-validation results.

**3**. **Support Vector Machine (SVM):**

* **Overview:** SVM is a powerful classification technique that finds the hyperplane that best separates classes in the feature space. It can handle both linear and non-linear classification tasks using kernel functions.
* **Key Characteristics:** SVM aims to maximize the margin between classes. The kernel trick allows SVM to perform well on complex datasets with non-linear boundaries.
* **Parameters:** We used linear, rbf kernel with C = [np.logspace(-3, 3, 7)] and gamma = ['scale', 'auto']. These parameters were tuned using grid search and cross-validation.

**4. Decision Tree:**

* **Overview:** The Decision Tree classifier splits the feature space into regions based on feature values, leading to a tree-like structure of decisions. It is intuitive and easy to interpret.
* **Key Characteristics:** The model makes decisions based on feature values to partition the data into classes. It can handle both numerical and categorical data and is prone to overfitting.
* **Parameters:** We used [Gini, impurity/entropy] for splitting criteria, and the maximum depth was set to [None, 10, 20, 30, 40, 50] to control overfitting. Pruning was applied to avoid overly complex trees.

**5. Naive Bayes Classifier:**

* **Overview:** Naive Bayes is a probabilistic classifier based on Bayes' theorem with the assumption of feature independence given the class. It is particularly effective for text classification and problems with large feature spaces.
* **Key Characteristics:** It assumes that features are conditionally independent given the class label, which simplifies the computation. It performs well even with a limited amount of data.
* **Parameters:** We used the Gaussian model variant of Naive Bayes, depending on the feature types. For Gaussian Naive Bayes, we set ['classifier\_\_var\_smoothing': np.logspace(-9, -4, 6)] for calculation stability

**Chapter 4**

**Implementation**

1. Use of Python Platform for writing the code with **Pandas, Numpy, Seaborn, Matplotlib.**
2. Hardware and Software Configuration:

Hardware Configuration:

* + CPU: 16 GB RAM, Quad core processor
  + GPU: 16GB RAM **Nvidia's GTX 1080Ti**

Software Required:

* + **Anaconda**: It is a package management software with free and open-source distribution of the Python and R programming language for scientific computations (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify deployment.
  + **Jupyter Notebook**:

Jupyter is a web-based interactive development environment for Jupyter notebooks, code, and data.

Jupyter is flexible: configure and arrange the user interface to support a wide range of workflows in data science, scientific computing, and machine learning.

Jupyter is extensible and modular: write plugins that add new components and integrate with existing ones.

* + **Spyder**: Spyder, the Scientific Python Development Environment, is a free integrated development environment (IDE) and open-source scientific environment that is included with Anaconda written in Python, for Python, and designed by and for scientists, engineers and data analysts.

It includes editing, interactive testing, debugging, and introspection features with the data exploration, interactive execution, deep inspection, and beautiful visualization capabilities of a scientific package.

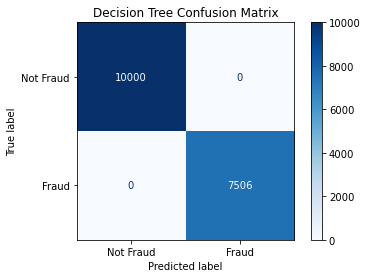
**Chapter 5**

**Results**

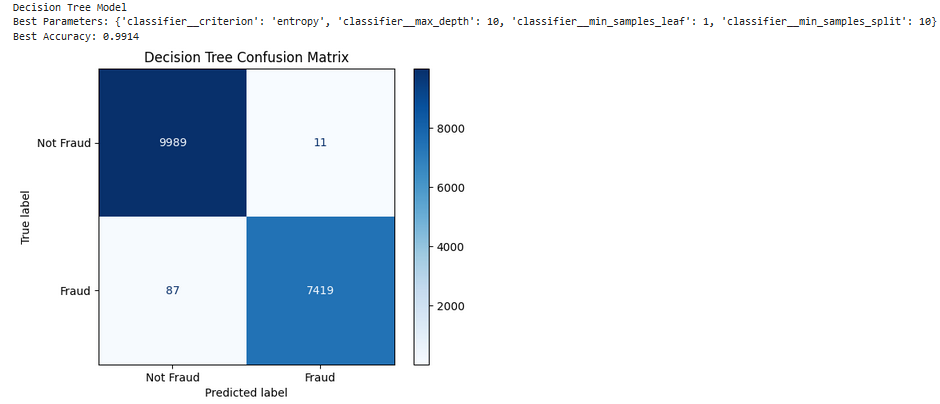
**Predicted Results –**

Best Model: Decision Tree with Accuracy: 0.9914

**Without Parameter Tunning :**



**With Parameter Tunning :**



**Accuracy Score –**

**With Parameter Tunning :**

Logistic Regression Model  
Best Parameters: {'classifier\_\_C': 1.12, 'classifier\_\_penalty': 'l2', 'classifier\_\_solver': 'sag'}  
Best Accuracy: 0.8959

K-Nearest Neighbors Model  
Best Parameters: {'classifier\_\_metric': 'manhattan', 'classifier\_\_n\_neighbors': 6, 'classifier\_\_weights': 'distance'}  
Best Accuracy: 0.9692

Decision Tree Model  
Best Parameters: {'classifier\_\_criterion': 'entropy', 'classifier\_\_max\_depth': 10, 'classifier\_\_min\_samples\_leaf': 1, 'classifier\_\_min\_samples\_split': 10}  
Best Accuracy: 0.9914

Support Vector Classifier Model  
Best Parameters: {'classifier\_\_C': 1.0, 'classifier\_\_gamma': 'scale', 'classifier\_\_kernel': 'rbf'}  
Best Accuracy: 0.9780

Naive Bayes Model  
Best Parameters: {'classifier\_\_var\_smoothing': 1e-08}  
Best Accuracy: 0.9584

**Chapter 6**

**Conclusion**

**6.1 Conclusion**

* In conclusion, the main objective of this project was to find the most suited model in credit card fraud detection in terms of the machine learning techniques chosen for the project, and it was met by building the five models and finding the accuracies of them all, the best model in terms of accuracies is Decision Tree Classifier which scored 99.32% without parameter tunning.
* The best model in terms of accuracies is \_\_\_\_\_\_\_\_ which scored \_\_\_\_\_% with parameter tunning.

**6.2 Future Enhancement**

* The model's performance could potentially be enhanced by employing more advanced techniques. For instance, ensemble methods such as Random Forests, Gradient Boosting, or stacking can be explored to combine the strengths of multiple classifiers. Additionally, optimizing the model using techniques like gradient descent can further refine the model’s accuracy and generalization capabilities. Future work could involve experimenting with these methods to improve the predictive performance of the model.

**Chapter 7**

**References**

[1] Zareapoor, M., Seeja.K.R, S. K. R., & Afshar Alam, M. (2012). Analysis on credit card fraud detection techniques: Based on certain design criteria. International Journal of Computer Applications, 52(3), 35–42. <https://doi.org/10.5120/8184-1538>

[2] Gupta, A., Lohani, M. C., & Manchanda, M. (2021). Financial fraud detection using naive Bayes algorithm in highly imbalance data set. Journal of Discrete Mathematical Sciences and Cryptography, 24(5), 1559–1572. <https://doi.org/10.1080/09720529.2021.1969733>

[3] Dheepa, V., & Dhanapal, R. (2012). Behavior based credit card fraud detection using support vector machines. ICTACT Journal on Soft Computing, 02(04), 391–397. <https://doi.org/10.21917/ijsc.2012.0061>

[4] Adepoju, O., Wosowei, J., lawte, S., & Jaiman, H. (2019). Comparative evaluation of credit card fraud detection using machine learning techniques. 2019 Global Conference for Advancement in Technology (GCAT). <https://doi.org/10.1109/gcat47503.2019.8978372>

[5] Alenzi, H. Z., & Aljehane, N. O. (2020). Fraud detection in credit cards using logistic regression. International Journal of Advanced Computer Science and Applications, 11(12). <https://doi.org/10.14569/ijacsa.2020.0111265>

[6]<https://www.kaggle.com/code/muhammadehsan000/credit-card-transactions-insights-visualization>