Credit Card Fraud Detection

Gayathri Sundareshwar, Keerthana Gopikrishnan, Deepasha Jenamani and Pallavi Jain

Department of Applied Data Science, San Jose State University

DATA 245 – Machine Learning and Tech

Shin Yu Chang

December 10, 2022

Table of Contents

1. Introduction

1.1 Project Background and Problem Definition

Advancements in technology have always been a driving force behind the rapid changes in lifestyle developments. The adaptation of innovative technology and software solutions has been prevalent across a wide range of industries. And financial services are not an exception to this change of digital transformation. The primary focus of finance-based organizations is to provide efficient, practical solutions and a more relevant and innovative customer experience. Thus, the influence of technology in these industries cannot be overlooked. Financial technology (FinTech) firms can develop many innovative solutions by utilizing technology and advanced software solutions to achieve faster, more convenient, and more reliable services and better customer experience. In addition, it has led to the development of solutions such as online transactions, cybersecurity, and improved risk management. And no field is devoid of issues poking through, such as fraudulent activities and compromised security.

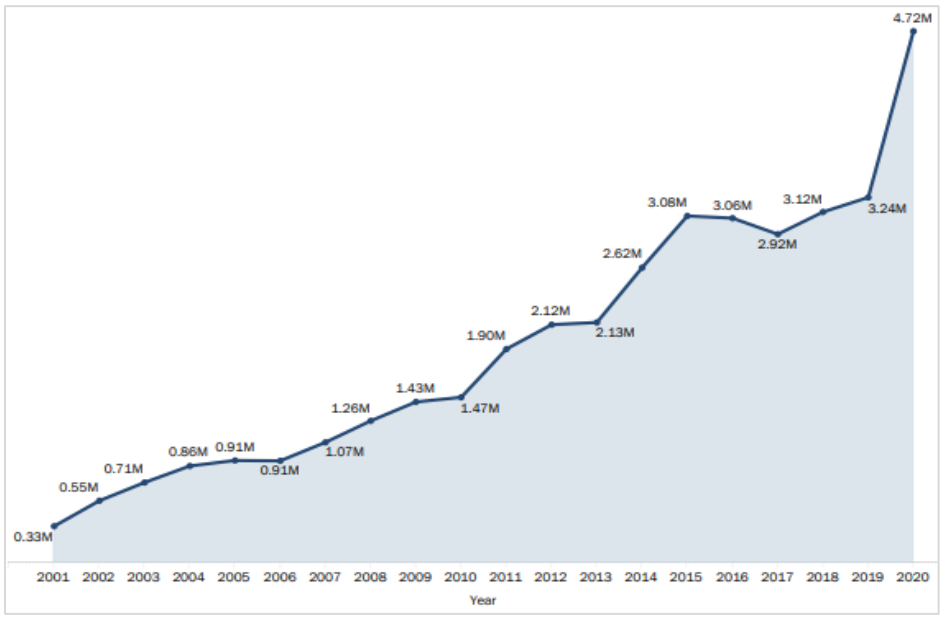
With such advancements in the services provided by the financial sectors, customers are more interested in shifting to digital banking, online transactions, and digital payment technologies because of the convenience. There are many ways to perform online transactions, such as debit or credit cards and digital payment apps like Apple Pay or Google Pay. The more people start using these FinTech solutions, the safety of their funds, personal data, and transaction details becomes a prime concern. The consumer can face several types of payment fraud, such as identity theft, phishing attacks, or wire transfer scams.

("Consumer Sentinel Network," 2020) showcased how in recent times, there has been a steep climb in the number of frauds reported annually, which can be found in figure 1. The report also stated that credit card fraud is the second most commonly reported identity theft type.

Figure.2 displays the total count of reports received, classified using the complaint type, and ranked based on quantity. Figure 3 also shows how credit card fraud is alarmingly rising as the most commonly reported identity theft type. These statistics raise a concern about how well-protected the consumers' financial information is. This acted as the reason behind the choice of this project.

Figure 1

Number of Fraud, Identity Theft and Other Reports by Year



Note: Consumer Sentinel Network Databook 2020 (ftc.gov)

Figure 2

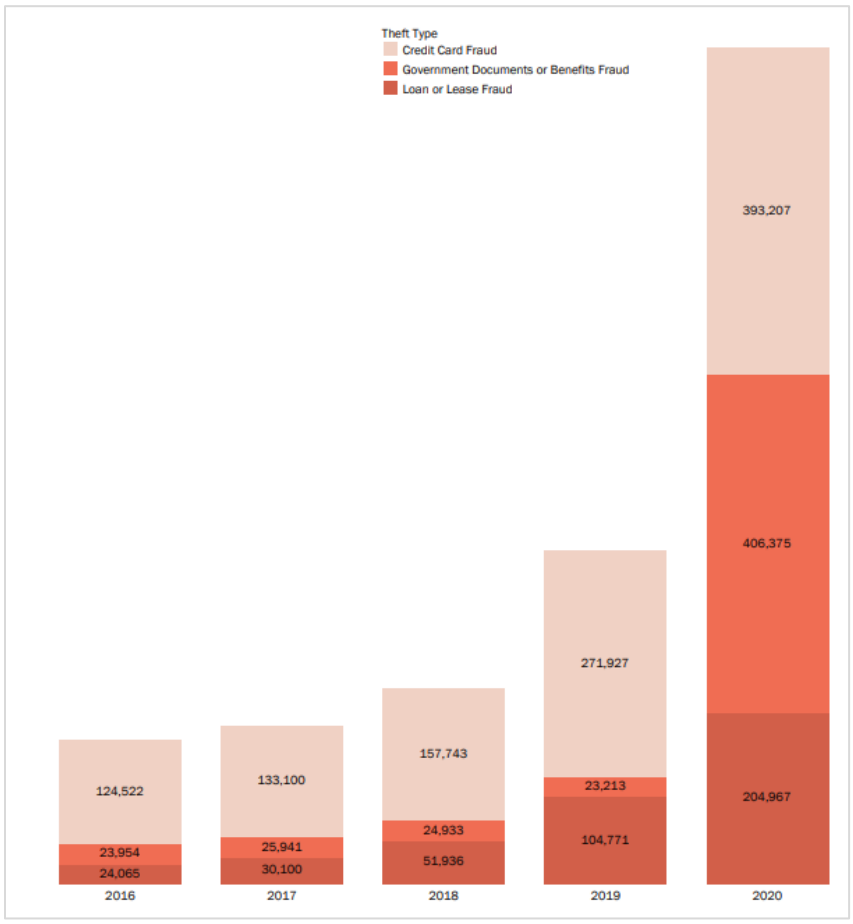
Top Identity Theft Types based on Number of Reports

Rank	Theft Type	# of Reports
1	Government Documents or Benefits Fraud	406,375
2	Credit Card Fraud	393,207
3	Other Identity Theft	353,152
4	Loan or Lease Fraud	204,967
5	Employment or Tax-Related Fraud	113,529
6	Phone or Utilities Fraud	99,539
7	Bank Fraud	89,476

Note: Consumer Sentinel Network Databook 2020 (ftc.gov)

Figure 3

Top Three Identity Theft Report Types by Year



Note: Consumer Sentinel Network Databook 2020 (ftc.gov)

In addition, credit card fraud, which falls under the category of identity theft, is also easy to target. Various online sites, including eCommerce platforms, have increased this risk with the availability of online payment methods. There are many ways in which fraudsters commit these frauds, such as stealing a cardholder's account, making fraudulent copies by skimming technique, and taking over card holder's account by phishing. According to the data published by FTC, the number of reported credit card frauds in 2020 was 393207. Thus, it has become essential to detect fraudster activity beforehand and prevent it from happening to ensure a secured transaction.

However, before devising any solution for this issue, learning, comprehending, and analyzing the historical pattern is advisable, which helps gain valuable insights for arriving at an optimal solution. The volume of historical data plays a crucial role in achieving a better solution, as more data aids in exploring the primary factors behind the problem. Machine learning models were considered more efficient than traditional fraud detection models while keeping the key idea in focus. The primary benefits of using machine learning are faster detection, higher accuracy, and improved efficiency with a more extensive set of data.

1.2 Project Objectives

The main objective of this project is to develop a machine-learning model which can detect and classify fraudulent credit card transactions. The recorded credit card transactions by various users were passed on to various ML models to classify the transactions as fraudulent or valid. In order to achieve this, the recorded transactions must pass through various stages of processing which turned out to be subset objectives that also have to be conquered for the resultant model to be fully functional. Some of the objectives are listed down below,

- Understanding the flow of transactions and identifying the possibility of fraud to happen,
- Collection of data with required features and exploratory analysis of the collected data,
- Cleansing the collected raw data by passing it through various wrangling and pre-processing methods,
- Transformation of features in case it is possible to quantify or aggregate it.
- Deriving additional features from the initially available features,
- Designing and Building the Machine Learning Models,
- Evaluation and Comparison of the built models and reporting of the performance.
- Documentation of the findings for future reference and maintenance.

As the initial step, the extracted raw data will be passed through various cleaning and transformation techniques, and the final cleansed data set passed to the model should be derived. An exploratory analysis will help decide which approach to the data works better. The analysis was done by understanding and analyzing the historical pattern in the data. It is necessary to understand which kind of data to prefer, keeping the project's goal in mind, which is achievable by performing appropriate business analysis.

Based on this project, a dataset's necessary elements are credit card and transaction details, the transaction's location details, and the card holder's certain demographic information. Once an apt data source is selected, the data is collected after devising a robust collection plan by complying with the policies enforced by the data owner. Once the desired data is collected, performing the necessary pre-processing steps is essential to handle any missing, incomplete, noisy, or inconsistent issues. These steps ensure that the accuracy and performance of the model are accurate. In addition, certain derived features can aid in attaining valuable insights; thus, required transformation steps are performed to achieve this.

After completing these above steps, it is necessary to understand the distribution of data and the existing correlation among the features; thus, the required exploratory analysis is performed. After analyzing the patterns and identifying the key features, appropriate machine-learning models are recommended, which will detect the frauds and classify those accurately. Nevertheless, it is essential to compare the selected models using various performance evaluation techniques to find an optimum model. The techniques used here are a confusion matrix, precision, F1 score, and ROC curve analysis. Simply put, the model's main objective is to devise an efficient model that can categorize fraudulent and non-fraudulent transactions, respectively. The sections below expand further on the path taken to achieve this model.

1.3 Project Requirements

1.3.1 Data Requirement

The dataset utilized in this project was retrieved from Kaggle, a vast repository of community-published data. The data source owner enforced specific policies, rules, and ethical guidelines, which needed to be agreed upon before downloading a dataset from the source. However, the dataset used in this project was under a public domain license; therefore, no specific policies were enforced during further use since it was not a mandate. According to the site, downloading a dataset requires a Kaggle account; thus, a zip file containing the raw data was downloaded after logging in with a valid account.

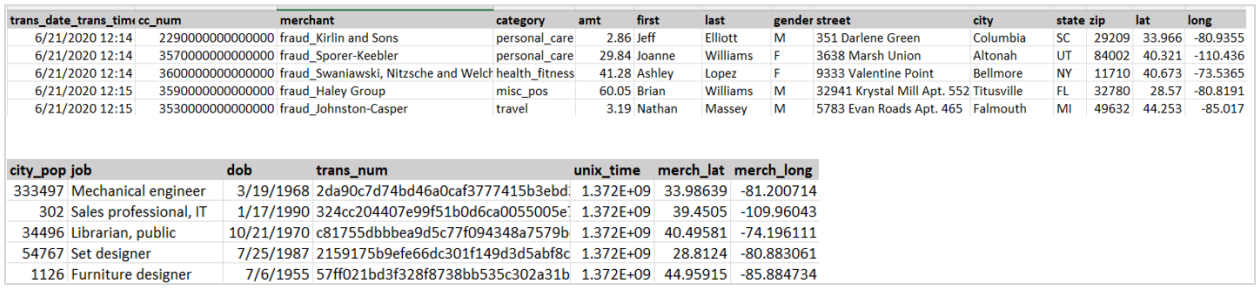
The model requires the extraction of each raw feature. Along with this, it also needs additional derived features which can play a vital role in the performance and accuracy of the recommended model. Additionally, it requires associated labels for fraud which will be required in the training phase to ensure better learning by the model.

The data source referred to for this project generates a simulated credit card transaction dataset. This dataset contains legitimate and fraudulent transactions from 1st January 2019 to 31st December 2020, which involves the credit card transactions of 1000 customers with 800 merchants. After the simulation, the files were combined and converted into a standard format. The source has two sets of raw data, which will be merged into a single file while processing. Due to the nature of the project, certain crucial features needed to be present in the dataset as a mandate. Some of those features were the transaction amount, location details, shopping category details, transaction timeline details, and consumer information. The raw data set may contain abnormalities, outliers, and redundant records, which were handled during the later parts. Figure 4 below displays a set of sample records from one of the raw source files. The format of

the source file was a comma-separated (.csv) file. Figure 5 displays the fields present in the source file.

Figure 4

Sample Records from One of the Raw Source File



trans_date_trans_time	cc_num	merchant	category	amt	first	last	gender	street	city	state	zip	lat	long
6/21/2020 12:14	2290000000000000	fraud_Kirlin and Sons	personal_care	2.86	Jeff	Elliott	M	351 Darlene Green	Columbia	SC	29209	33.966	-80.9355
6/21/2020 12:14	3570000000000000	fraud_Sporer-Keebler	personal_care	29.84	Joanne	Williams	F	3638 Marsh Union	Altonah	UT	84002	40.321	-110.436
6/21/2020 12:14	3600000000000000	fraud_Swaniawski, Nitzsche and Welch	health_fitness	41.28	Ashley	Lopez	F	9333 Valentine Point	Bellmore	NY	11710	40.673	-73.5365
6/21/2020 12:15	3590000000000000	fraud_Haley Group	misc_pos	60.05	Brian	Williams	M	32941 Krystal Mill Apt. 552	Titusville	FL	32780	28.57	-80.8191
6/21/2020 12:15	3530000000000000	fraud_Johnston-Casper	travel	3.19	Nathan	Massey	M	5783 Evan Roads Apt. 465	Falmouth	MI	49632	44.253	-85.017

city_pop	job	dob	trans_num	unix_time	merch_lat	merch_long
333497	Mechanical engineer	3/19/1968	2da90c7d74bd46a0caf3777415b3ebd	1.372E+09	33.98639	-81.200714
302	Sales professional, IT	1/17/1990	324cc204407e99f51b0d6ca0055005e	1.372E+09	39.4505	-109.96043
34496	Librarian, public	10/21/1970	c81755dbbbea9d5c77f094348a7579b	1.372E+09	40.49581	-74.196111
54767	Set designer	7/25/1987	2159175b9efe66dc301f149d3d5abf8c	1.372E+09	28.8124	-80.883061
1126	Furniture designer	7/6/1955	57ff021bd3f328f8738bb535c302a31b	1.372E+09	44.95915	-85.884734

Figure 5

Fields in the Source file

```
trans_date_trans_time     object
cc_num                   float64
merchant                  object
category                  object
amt                      float64
first                     object
last                      object
gender                    object
street                    object
city                      object
state                     object
zip                        int64
lat                      float64
long                     float64
city_pop                   int64
job                       object
dob                       object
trans_num                 object
unix_time                  int64
merch_lat                float64
merch_long               float64
is_fraud                   int64
dtype: object
```

1.3.2 AI Requirement

This section covers the needed AI features to achieve the project's goal. Six machine learning models were initially chosen to be evaluated and compared based on the dataset for prediction accuracy and performance. The first was a Support Vector Machine(SVM) for the classification task, which is considered well-suited to this project because of its efficiency in dealing with non-linear classifications. The second one was the Logistic regression(LR) model because it was easier to implement and interpret and had better efficiency while training. Thirdly, the Decision Tree(DT) model as it is purely deterministic, and controlling and explaining the classification result is simpler. The fourth was a Random Forest(RF) model, which is well-suited because of the straightforward implementation and interpretation of classifications. The fifth was a Gaussian Naive Bayes(GNB) classifier, which supports continuous-valued features while also modeling each feature into a normal distribution. The last one was the k-Nearest Neighbor(KNN) because it classifies the new data points based on the chosen similarity measure and gives better accuracy. Since the goal was to experiment with semi-supervised algorithms, the AI requirements extended to using Principal Component Analysis (PCA) and Silhouette Score.

1.3.3 Functional Requirement

Functional requirements include various processes, such as pre-processing chosen data and classifying transactions as fraudulent or not fraudulent. The dataset chosen already comes with a labeled flag denoting whether the transaction is fraudulent or not. Hence, in this case, the functional requirement would be to build a Machine Learning algorithm that gets trained with this existing labeled flag and the data, enhancing learning about fraud transactions. Once the model is trained, it should be able to predict the occurrence of fraudulent transactions in real-time financial transactions.

Another set of requirements that needs to be considered is the computational needs of the models, and the resources must be sufficient to execute the model with relevant data. Minimum requirements must be evaluated, such as system storage and memory size. The first one is system storage which must be a minimum of 3GB for efficient storing, acquiring, and processing of the raw dataset.

Secondly, minimum system storage of 8GB is required to load the input dataset for further pre-processing and the model's classification task. In addition, a GPU having a minimum memory of 8GB is essential for using the classification algorithm libraries that use GPU parallelization capabilities. The above-mentioned minimum requirements are essential for efficient modeling tasks.

1.4 Project Deliverables

1.4.1 Project Abstract

The project abstract provides a bird's eye view of the entire project and the outcomes. It can also be referred to as a concise review of the entire project explaining why the specific topic was chosen. The research goal, techniques, and implications will also be included. Additionally, it can be equated to simplifying complex process explanations and aids in quick comprehension of the subject matter. A good abstract is directly proportional to the consumer interest in the study.

1.4.2 Data Collection Plan

The data collection plan specifies keeping track of the findings observed during the brainstorming sessions related to data during the project requirement gathering discussion. It helps identify the crucial factors that play a role in selecting features, compatibility, and volume requirements. The plan specifies collection procedures, schedules, and persons responsible.

1.4.3 Data Exploration Plan

The data exploration plan consists of the steps involving analyzing the existing data. It also includes the pre-processing steps necessary while preparing the data to become an influential model input. The raw data chosen is often unclean. Hence, there exist a set of pre-processing and transformation techniques that could help attain insights. The data exploration plan includes the various cleaning, transformation, and pre-processing techniques that can be done so that the visualizations generated can help identify patterns.

1.4.4 Data Engineering

Data engineering can be explained as the second phase of processing. Any additional quantifiable information identified from the previous step's visualizations is brought to life. It deals with extracting derived features from the original set of features. The data engineering phase also results in the data being cleaner and more standardized. The data engineering phase helps to add additional summarizing statistical features. The data is brought to a form that the predictions tend to be of higher accuracy. The process ends with the test and train split of data.

1.4.5 Project Presentation

The project's goal, details about the dataset, the methodology employed, the results, limits, and potential future improvements are all presented in the presentation, which is a helpful deck of PowerPoint slides. As it explains, it serves as a guide for the project.

1.4.6 Final Group Report

The team and individual efforts will be included in the final group report. It is a paper that summarizes the whole procedure from beginning to end and comprises multiple chapters that give a whole picture of the project's realization. The report starts with the definition of the problem and ends with the study's results documenting each step along the way.

1.5 Technology and Solution Survey

Identifying credit card fraud has attracted a lot of research attention, and several strategies, with a focus on neural networks, data mining, and distributed data mining, have been proposed. This subject has been covered in a wide variety of projects, and most of them employed supervised machine-learning algorithms to predict whether a transaction would be fraudulent. The machine learning model used mainly is Random Forest, Linear Regression, Decision Tree, SVM, KNN, and Neural Networks. Imbalanced data needs to be dealt with in most projects, where fraud data is negligible compared to non-fraud data.

ML models can be categorized, such as; firstly, supervised ML models used in most research papers use a dataset with a label that will be passed to the model to perform predictions. Secondly, unlabeled data is clustered and analyzed using an unsupervised ML model. Thirdly, semi-supervised ML models process a large amount of unlabeled data with a small amount of labeled data to make predictions. Lastly, reinforcement learning uses a trial-and-error method; in this method, undesired and desired results are punished and rewarded.

The research papers helped attain insights into what worked well for the models and what could be improved for better results. Figure 6 below shows the comparison of various machine learning models used in the research papers for credit card fraud detection. A few similarities and differences between the works were found while doing the technology and solution survey and carefully examining the relevant reference papers. The next sections will cover a few of the observations.

1.5.1 Similarities between the Related Work

The background research provided insights into the algorithms, architecture, methodologies, data sources, and ideal ML models. Some similarities were observed between the

various works, which helped identify the common issues and how they were resolved. The data sources chosen by most of the paper include longitude, latitude, amount, and other essential features that can help identify if a transaction was legitimate or not. The second similarity was the use of supervised machine learning algorithms such as random forest, decision tree, and logistic regression to classify the outcome of each trade. Thirdly, the research paper also solved the issue of imbalanced records by up sampling, down-sampling, or SMOTE the labels to avoid bias in the train and test dataset. Lastly, the data contain credit card details of customers, which is a privacy concern; hence, the data is masked or hashed to protect the individual's identity

Figure 6

Observations from Existing Works

Existing Model	Advantages	Disadvantages	Performance
DNN	1. Commercially supported. 2. Identifies outliers	1. High cost for distance calculation. 2. Illicit activities show uneven nature	Accuracy: 99% (Sumanth et al., 2022)
RF	1. Works well for large datasets. 2. Suited for real-world applications 3. Firm estimate of the generalization error.	1. It Doesn't work well with semi-structured data. 2. Sensitive to noise. 3. Affects prediction accuracy when the data is imbalanced	Accuracy: 85.8% (Hussain et al., 2022) Accuracy: 99.95% (Kirar et al., 2021) Accuracy: 96.77% (Tanouz et al., 2021) Accuracy: 99.989% (Rathore et al., 2021)
DT	1. Handles imbalanced data.	1. Limitation of memory and computation requirement. 2. Overfitting of training data.	Precision: 85.11% (Khatri et al., 2020)
XGBoost	1. Works well on a large dataset. 2. Real-world application	1. Doesn't perform well in different domains	Accuracy: 99.95% (Jain et al., 2020)
KNN	1. Suitable for multiclass classification.	1. Training the model is expensive. 2. Not suitable for large datasets.	Accuracy: 99.947% (Shah and Mehta, 2021)
NN	1. Fast classification of unknown or new data.	1. It Doesn't work well when data has outliers.	Accuracy: 99.87% (Rai and Dwivedi, 2020)
LR	1. Use of cross-validation methods for selecting features.	1. Not suitable for large datasets. 2. It Doesn't work well for biased data.	Accuracy: 99.38% (Naveen and Diwan, 2022)
PK-XGBoost	1. Decomposition of data. 2. Privacy protection of data	1. Cost of training is high.	Accuracy: 78.5% (Wen et al., 2020)

1.5.2 Contrasts between the Related Work

The background research also showed the contrast apart from the similarities mentioned in the previous section. The papers showed how they handled similar issues using different solutions. Some disparities are associated with the algorithm, methodologies, and evaluation technique. Firstly, the contrast between the algorithm used to predict fraudulent transactions, such as Random Forest, Decision Tree, Support Vector Machine, Neural Network, XG Boost, Linear Regression, Deep Neural Network, and K Nearest Neighbor. It is also important to remember that, even though there may be similarities between a few works compared to other works outside the cluster, it can also be seen as a contrast.

The second discrepancy between the papers focuses on the evaluation techniques used, such as precision, F1 score, or ROC curve, instead of using only accuracy as the metric for evaluating the model's performance. The third contrasting point is the number of features extracted from the raw data. Some papers, for example, used features such as calculating user zone to identify the customer's buying patterns, age, hour, and amount. In contrast, others used feature selection to select the optimal feature set.

1.6 Literature survey of existing research

The objective of our project has been the subject of several earlier works. Some existing works catered as a base while establishing a foundational understanding of the approaches to reach the goal. A few of those existing studies are discussed in this section,

A research study by Sumanth et al. (2022) suggested using outlier mining using a distance sum metric to pre-process the data. A deep neural network (DNN), Support Vector Machine (SVM), and Navies Bayes (NB) were used to classify the transactions. The accuracy of

the DNN was 99%, followed by SVM having 97% observed. This study shows that it can be commercially supported for large-scale use and help identify outliers more efficiently.

Hussain et al. (2022) proposed a methodology in which, at a given node, samples were generated at random means to obtain a subset of the entire feature set that will be passed to the training set. Machine learning models such as SVM, Decision Tree (DT), and Random Forest (RF) out of these RF models had the best accuracy of 85.8%. This study showed that the model would firmly estimate the error caused by a generalization and ensure that the model does not overfit and is suitable for a large dataset.

Lucas et al. (2019) suggested agglomerative clustering, which divided the data into multiple clusters such that similar data points were grouped into clusters. The RF model showed an increase in precision-recall AUC by 2.5% by integrating clustering and features such as day, hours, and month. This study showed that using a distance matrix to detect the behavior and select the optimal feature improved the prediction result.

Khatri et al. (2020) addressed the limitation of imbalanced data by using different evaluation metrics. The model was designed using supervised machine learning algorithms such as DT, K Nearest Neighbor, Logistic Regression(LR), RF, and NB. The results show a precision of 85.11% and a sensitivity of 79.21% was acquired by DT, which was the best-suited model. Tanouz et al. (2021) suggested ways to handle imbalanced data and outliers, which will help detect anomalous activities. The classification ML models, such as LR, RF, and NB, were implemented to identify the model that works well when the data passed to the training test is under-sampled. The results show that the Random Forest classifier performs best, having 96.7741% accuracy, 100% precision, 91.1111% recall, 95.3488% f1 scores, and 95.5555 ROU-

AUC scores. This study shows that it is suitable for real-world applications as obtaining balanced data is quite impossible.

Jain et al. (2020) proposed combining machine learning and artificial intelligence (AI) to identify fraudulent transactions. The model implemented were RF, DT, and Extreme Gradient Boosting (XG-Boost). The results show that the XG-Boost model obtained an accuracy of 99.95%. The study shows that this model is suited for real-world applications and how AI can be incorporated during modeling.

Rai and Dwivedi (2020) suggested using an unsupervised machine-learning algorithm to handle outliers present in the transaction data. The models include Neural Network (NN), Local Outlier Factor (LOF), and K-means clustering that clustered the data by grouping similar data points and removing the outliers present. The results show that the NN model had an accuracy of 99.87% among the others. This study showed that when performing an unsupervised model, issues such as limitations in memory and computation requirements can arise, which need to be addressed to obtain good prediction results.

Shah and Mehta (2021) explored the problems that can arise while implementing a detection model, such as imbalanced data, misclassification significance, lack of identifying new patterns and standard metrics, and fraud recognition costs. The methodology, such as DT, Isolation Forest, KNN, LR, SVM, and RF, were implemented. The results show that the KNN model obtained an accuracy of 99.47%. This study provided insights into how balancing the data will help reduce misclassification errors and the cost associated with building a model.

Rathore et al. (2021) investigated feature extraction's importance, which helped create a new dimension to the modeling. The features such as amount, client habit, and user zone were extracted. The ML models were implemented, such as LR, KNN, DT, and RF. The accuracy of

the RF model was 99.989%. This study shows that feature extraction help scales the result better when working with a large amount of data.

Naveen and Diwan (2022) proposed building a detection model that uses predictive analysis to help determine the impact associations between features. Models such as LR, SVM, and Quadratic Discriminant Analysis (QDA) were implemented, and the LR model's accuracy was 99.38%. This study showed that the sampling technique could be performed to avoid bias in the data, and using the cross-validation method to select features will improve the model's performance.

Wen and Huang (2020) suggested combining the XG Boost algorithm with Kernel Principal Component Analysis (Kernel PCA) and implementing a hybrid model comprising unsupervised and supervised algorithms. The P - XG Boost outperformed all the other models with an accuracy of 78.5%. This study showed that the cost associated with training a hybrid model could be high due to factors like data decomposition and privacy of customer information.

2. Data and Project Management Plan

It is inevitable to carry out all the steps and procedures systematically to build an effective model. It is necessary to devise a proper plan for data and project management. A proper plan helps increase the preciseness of the model's prediction. The various procedures added to the data and project management plan will be discussed in this section, along with the plan.

2.1 Data Management Plan

A data management plan ensures that the entire data collection, management, security, and recovery process is covered. If any hiccup happens, the plan helps devise a way to deal with it. The various steps that were taken into consideration while devising the data management plan are discussed below,

2.1.1 Data collection Approaches

The dataset used in this project was a simulated credit card transaction dataset containing legitimate and fraudulent transactions from January 1st, 2019, to December 31st, 2020, acquired from Kaggle. The data collected relates to the credit cards used by 1000 clients who transacted with 800 retailers. Using Brandon Harris' Sparkov Data Generation tool, the simulation was produced. The simulator uses a python library called "faker" with a pre-defined list of customers and merchants. Weights can be added to parameters for the data that is being generated, such as age, demographic, amount, start date, end date, and minimum and the maximum number of transactions per day. The number of records in the dataset was around 1.85 million, consisting of data across all parameters, which were merged to create a realistic representation.

The copyright data simulator was accepted, which permits the use of the software and related documents. The guideline includes no restriction or limitation to use, modify, copy, distribute, publish, merge, or sell this software. Throughout the project, these requirements were upheld.

2.1.2 Data Storage Methods

Following the collection process, the credit card transaction information obtained from Kaggle was stored for use. The two storage methods used were Google Cloud Storage and the local memory of the individual team member's systems. The dataset's ethics and compliances were considered and respected during storage and development. Google cloud access, where the data was stored, was shared only among the team members.

The dataset collected from the source consisted of two files, "fraudtrain" and "fraudtest." As a consequent step, merging into a single file was necessary. The reason behind the merge can be attributed to the caution towards any bias that might exist in the default split. The merged file

was stored in google drive and local memory so that it shall be restored if any crash happens. In case of any updations in the data, versioning was enabled.

2.1.3 Data Management Methods

The data was stored in google cloud, and local machines were protected using a password. Access to download from the folder in the cloud was given only to the team members, ensuring it was maintained throughout the whole course of this project. The data management plan also includes timely backup of any new dataset derived after pre-processing steps, such as eliminating duplicates, handling nulls, and adding derived features. Timely backup after pre-processing also helps speed up further execution steps as the backup could be loaded directly into models instead of going through pre-processing again. Data management not only encloses the dataset, but any documentation work that was being done also needs to be backed up to avoid rework in case of loss.

2.1.4 Data Usage Mechanism

A PowerPoint presentation, evaluation results of machine learning models, and project report documentation were created throughout the project. It was crucial to track how the data was used during the progress of various steps, and doing so would be helpful in the future when dealing with similar situations. The precise procedures are documented in the project report, covering specific duties like comprehending the business, data collecting, preparation, modeling, and evaluation. It is also advisable to document any errors along with the fix made to rectify them. Documenting the errors will be helpful in quickly navigating similar scenarios in future projects. Future users may find these materials helpful in understanding how the data was used. +Figure 7 below shows the metadata of the credit card transaction dataset. The table showcases the initially assigned field type of the existing features.

Figure 7

Metadata of the Features with Data Types

Field Name	Description	Type
Field 1	Transaction Date and Time	String
Field 2	Credit Card Number	Float
Field 3	Merchant	String
Field 4	Category	String
Field 5	Amount	Float
Field 6	First Name	String
Field 7	Last Name	String
Field 8	Gender	String
Field 9	Street	String
Field 10	City	String
Field 11	State	String
Field 12	Zip Code	String
Field 13	Latitude	Float
Field 14	Longitude	Float
Field 15	City Population	Integer
Field 16	Job	String
Field 17	Date of Birth	String
Field 18	Transaction Number	String
Field 19	Unix Time Stamp	Integer
Field 20	Merchant Latitude	Float
Field 21	Merchant Longitude	Float
Field 22	Fraud	Integer

2.2 Project Development Methodology

For a project to be efficiently completed, there should be decorum in handling the various steps. The project development methodology determines how the work will be scheduled and prioritized while keeping the project's end goal in mind. A set of principles must be defined to plan, manage and execute the project properly. Thus, a framework known as Software

Development Life Cycle (SDLC) was followed and implemented. The framework can be explained as a series of steps that were detrimental to the various stages a project must go through to reach a successful completion. Singh (2009) described SDLC as a proper framework for software engineering that is critical for building reliable, cost-effective software that is easily maintainable (Software Development Life Cycle Section, Para.1). SDLC primarily consists of seven phases which are: requirement, analysis, design, development, testing, deployment, and maintenance. The development lifecycle updated over time, starting from a simple waterfall and growing upwards to Kanban and agile. Each model follows a series of specific steps to ensure development success. There are various ways to ensure successful completion, and especially the field of data science demands a solution-oriented approach to solving problems, and here data plays a crucial role.

Therefore, it is essential to look for valuable patterns and insights and find correlations. Hence, selecting a model that has worked exceptionally in similar circumstances was mandatory. For this project, a widely used methodology known as Cross-Industry Standard Process for Data Mining (CRISP-DM) was used. Its domain independence, flexibility, and iterative nature are the primary reasons behind utilizing this.

2.2.1 CRISP-DM

The CRoss Industry Standard Process for Data Mining (CRISP-DM) is a popular data mining and analysis methodology. This section explains the planning of the project development process and the steps incorporated based on the CRISP-DM methodology.

Business Understanding

The business understanding was the first stage, where an idea about the project's requirements was understood from a business perspective. It assists in gaining valuable insights

into the objective and focuses on determining the key factors that could influence the project's outcome. This phase involved multiple processes or tasks such as identifying project purpose, scope definition, background research, requirement gathering, and identifying viable data sources and documentation. Each process helped in gaining a more in-depth understanding of the project. Identifying project goals can be achieved through regular meetings with the stakeholders and having multiple brainstorming sessions among the members. After getting an idea about the project, it was essential to perform intensive research on various existing studies related to the topic.

Furthermore, the background research helped realize the existing shortcomings and how this can be improved and avoided in the current study. Following the background research, the next step was defining the scope, which helped draw a line between the items that would be part of the project's end product and those that were not. The business understanding phase also includes the step of requirement gathering, where the information regarding the required software, hardware, storage, and data requirements that were vital for the execution were identified. Constant communication with the stakeholders and associated teams helped condense the execution steps to the essential ones. Finally, a project plan document with updates regarding progress and achievements after each phase was prepared. Documenting the progress made so far could serve as future reference material. Once comprehensive insights were attained from the business understanding phase, the next one began.

Data Understanding

The resources identified in the previous stage aid in the initiation of data understanding phase. This phase involves the process of identifying potential data sources that could qualify to

act as the raw data source for this project, then deciding on one confirmed data source after reviewing the compliance policies, rules and guidelines. Choosing a good data source is a crucial factor that will influence the preciseness of the model. Hence, extra attention should be paid to what was chosen and whether it complies with all the regulations. Hence, the first step was to review the ethics and compliance policies laid out by the original authors of the shortlisted data sources. After this review, the dataset used throughout this project was chosen. The next step was to collect the data from the source without violating the ethics and compliance guidelines. Furthermore, it is essential to understand the data in depth to proceed further. Hence, an initial exploratory data analysis was performed. Any quantifiable information that could be derived and used was marked down. The analysis also helped determine the existing patterns that appear to be helpful in the future. The dataset's properties, such as the features present, data types, size, and format, were also examined in this phase.

Additionally, one more vital thing to keep track of is the distribution of values in features when compared with the target variable. Exploring the feature's distribution, performing statistical analysis, and exploring the correlation between the features also aided in gaining additional insights regarding the data. Developing the data quality assurance plan helped maintain the data's integrity and handle any mishaps—verifying the authenticity and quality of the data, which cannot be ignored. The verification process included various tasks, such as identifying incomplete, missing, or noisy data and eliminating duplicated values. The issues identified were dealt with during the data-wrangling process.

Data Preparation

The data preparation phase was initiated after conclusive knowledge regarding the chosen dataset was achieved in the understanding phase. The data preparation phase helps sculpt the

data into an even cleaner format while looking for an opportunity to perform further in-depth analysis. This phase played a significant role in deciding what data should be used for further analysis. The decision criteria included vital factors such as the relevance of the dataset to the project, the technical constraints accompanying it, a measure of how impeccable the data quality is, and the inclusion or exclusion of any attributes. The data quality was further raised by intensive data-wrangling operations, which aided in cleansing the data by performing tasks that eliminated corrupt data, redundancy and converting the data to be standardized.

Furthermore, additional calculated features that might be useful were identified. The selection of the derived features also depended on how quantifiable the data is and how it can enhance the model's performance,. Once decided, the additional features were derived by aggregation or merging operations. Merging happened when multiple sources were presented with a common object, whereas aggregation occurred when a new value was calculated by summarized information. Once a dataset that was clean enough to be passed to the modeling phase was generated, it was split into training and testing sets of the required proposition. The data was then passed on to the next modeling phase.

Modeling

After completing the data preparation phase, the obtained train and test datasets were used in various machine-learning techniques in this phase. Various models were identified based on the project's objective; however, only the six most suitable ones were identified for this task. It was required to devise an appropriate plan to determine the architecture explaining how to build the model. Based on this plan, models were built with the required parameter and hyperparameter settings. The models created were assessed to ensure they fulfill the goal from project and business perspectives. After comparing the outcome and quality of the model's

results, the parameters were tuned if required. And this ran iteratively until the best version of the model was identified. Moreover, a defect tracker was maintained to track any issues identified, and the fixes performed to handle the same.

Evaluation

An evaluation process was carried out to understand the degree to which the model fulfills and achieves the project's objective. The prediction accuracy and performance of the models were performed in this stage. The evaluation process involved various approaches, such as confusion matrix and ROC curve analysis. In addition, each model's performance was analyzed with the help of different dataset sizes. This performance evaluation also considered the model's memory usage, CPU utilization, and size. After performing the mentioned steps, a suitable model was identified, keeping the business success criteria and goal in focus. In this stage, a best-suited model was identified based on the performance and other factors as mentioned here. Based on the evaluation and assessment performed in this stage, a decision was made on whether to deploy the model or perform additional required iterations.

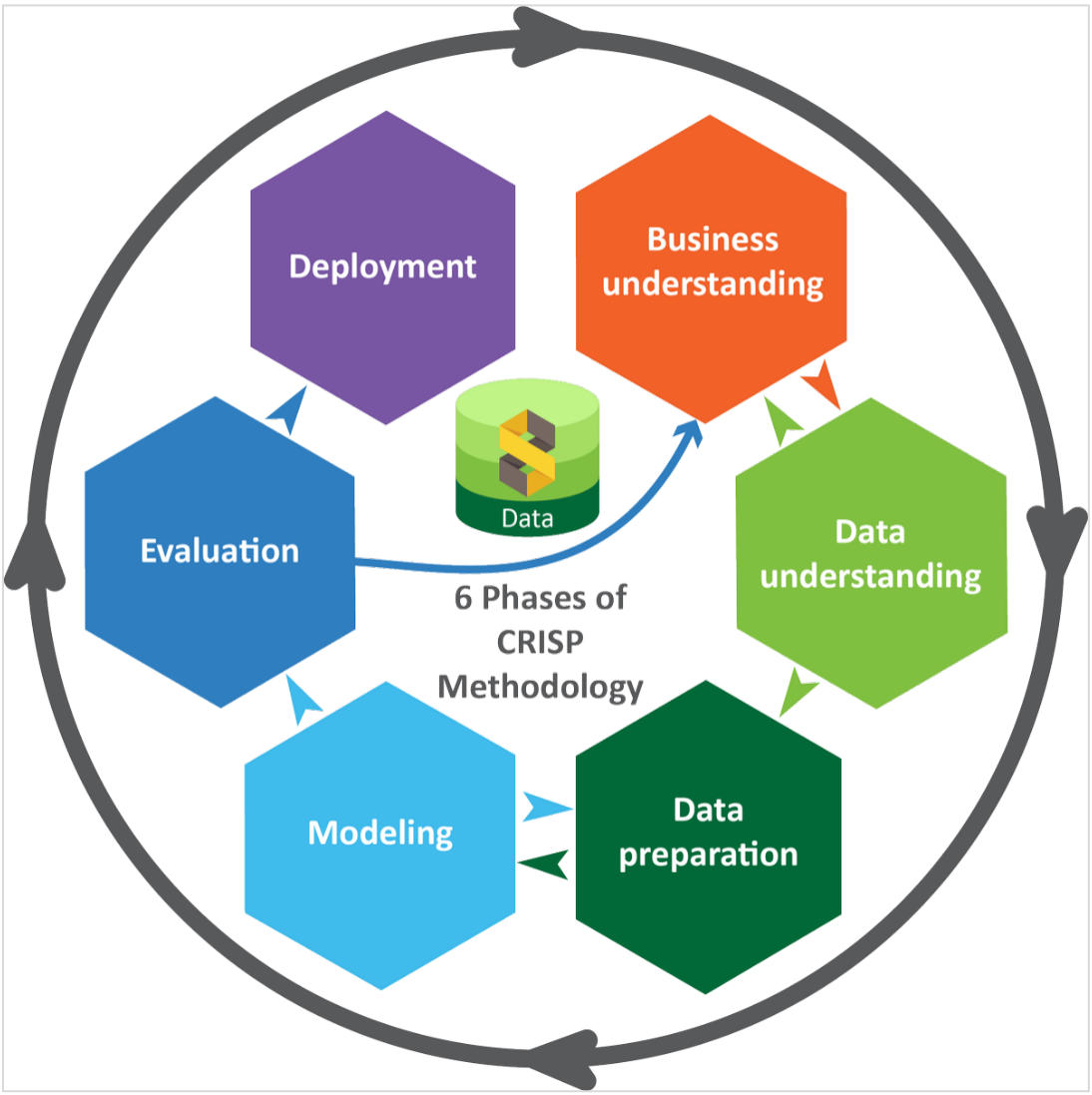
Deployment

After evaluating the models, a deployment strategy for the best-suited model was devised in this stage. This involved the necessary steps being performed and how to conduct those. However, it was essential to consider key factors in this stage, such as model size, packaging, versioning, and routing. This phase involved testing the model's performance in the production environment, which helped measure the performance under real and stressful environments as the production environment deals with a large volume of real-time data. In addition, a monitoring plan was also needed to be employed as the changes in the volume of data and its volatility could

affect the model's precision and performance. Figure 8 below depicts a pictorial representation of the various stages in CRISP-DM Architecture.

Figure 8

CRISP-DM Architecture



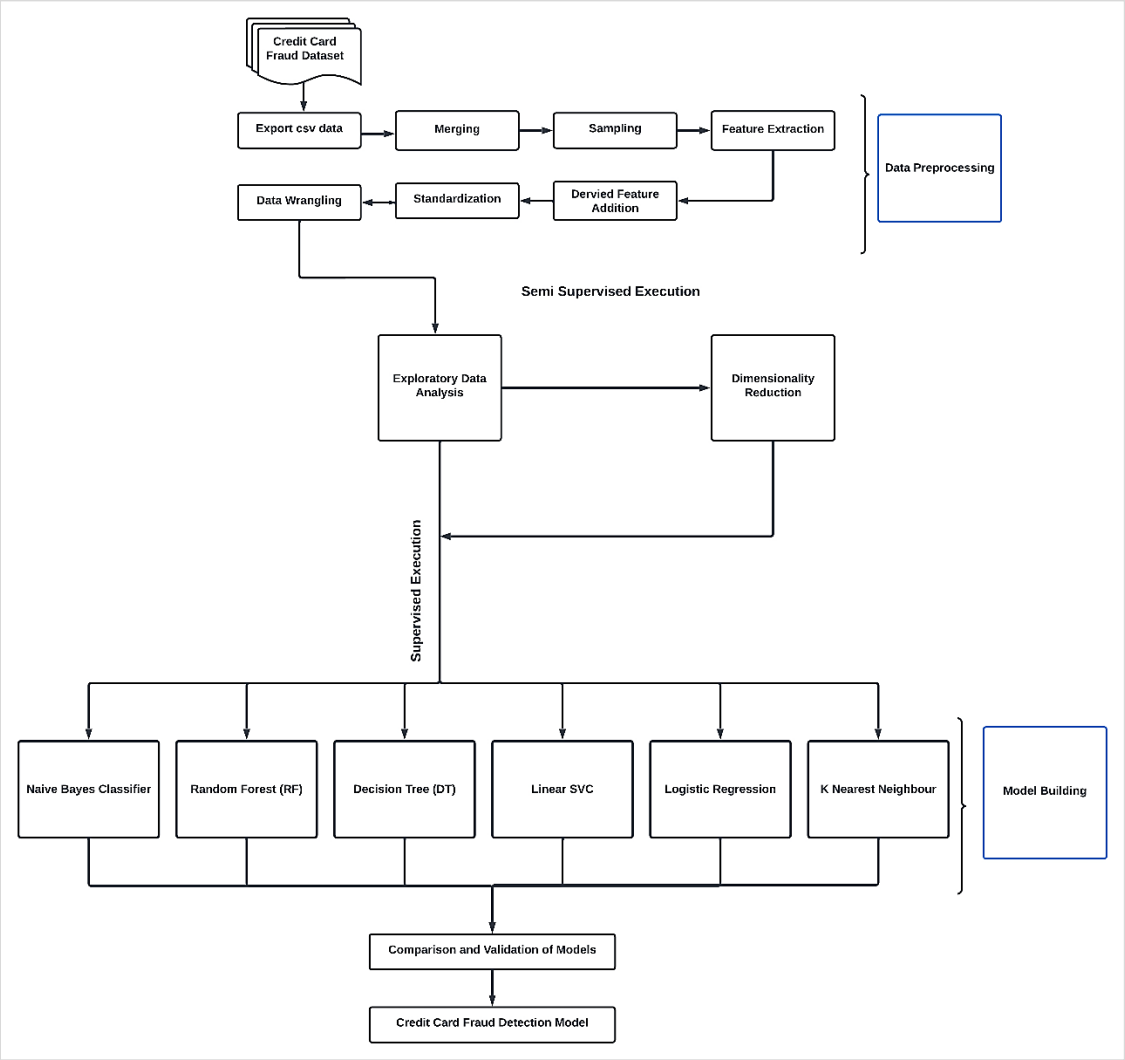
2.2.2 Project Workflow Development

A project flow diagram was developed to plot out the steps required for the study's successful completion. The workflow diagram will act as a base, creating the other organizational methodologies, such as a Work Breakdown Structure. The various stages depicted

in the workflow diagram can be split into the various phases of CRISP-DM. Figure 9 below depicts the workflow diagram.

Figure 9

Workflow Diagram



2.3 Project Organization Plan

A project organization plan comes in handy when there needs to be an order maintained. Devising an organizational plan at the initial stage can help track and check whether the project's progress is on course. Some of the project organization plans implemented were a Work

Breakdown Structure(WBS) and Project Evaluation and Review Technique (PERT) Chart, which will be discussed in detail in this section.

2.3.1 Work Breakdown Structure

A complex project plan represented through a Work Breakdown Structure (WBS) is always more understandable and easy to read. WBS can be defined as a diagram illustrating the breakdown of the project into multiple components and work packages to facilitate efficient completion and easier tracking. Incorporating scope, cost, and schedule baselines, also guarantee the coherence of project plans. The WBS is useful for outlining the general course that the project must take to achieve timely completion. It will also help identify potential roadblocks and aid in the creation of an advanced contingency plan. The six CRISP-DM phases were used to construct the WBS. The higher-level breakdown of the six stages covered in the previous sections was included in the WBS.

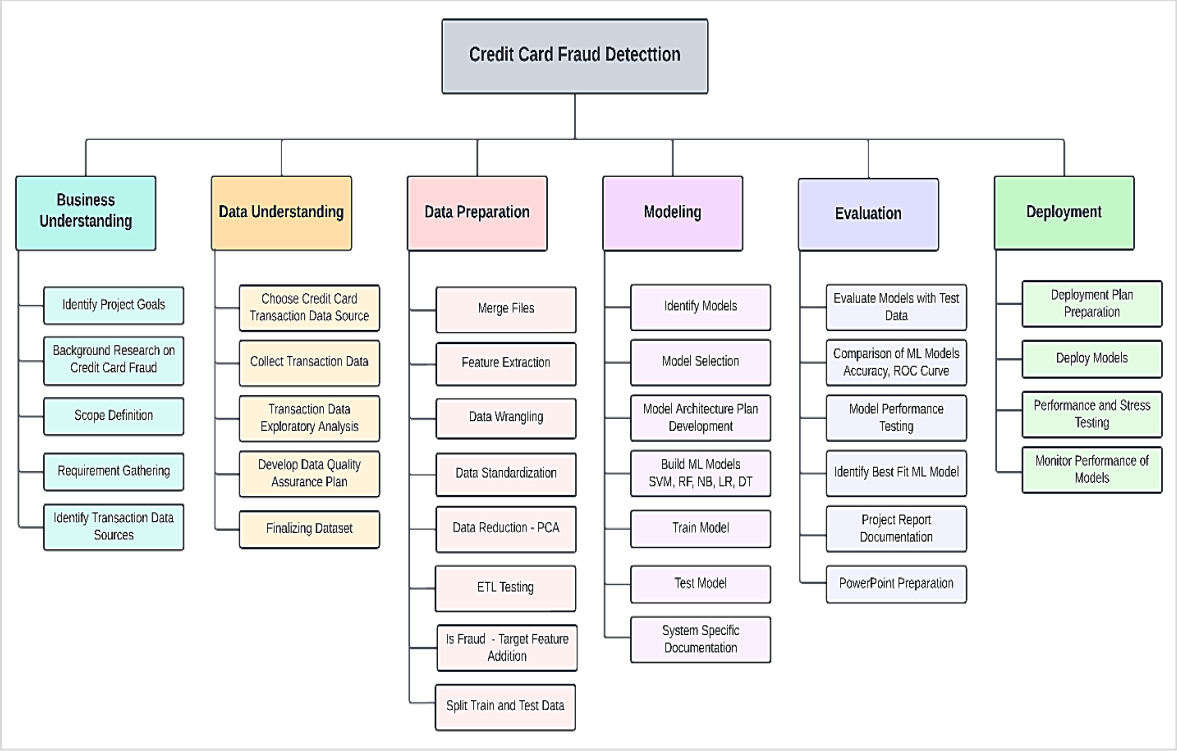
The CRISP-DM technique promotes the focus on business objectives, adaptability, and an iterative approach independent of any particular technology are some of its key features. It comprises six main stages: business understanding, data understanding, data preparation, modeling, evaluation, and deployment.

Figure 10 depicts the WBS created to represent the breakdown of the project. The top node shows the project's name, 'Credit Card Fraud Detection.' It has six phases as per its association with CRISP-DM. The phases are Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, and Deployment. Each phase discussed above has its section and tasks tagged under it with various dependencies. The tasks assigned to the phases give an overall picture of how the project must be carried out to reach a successful implementation. It also gives the customer a higher-level picture of how the development team is proceeding with

the requested integration. If any changes are required, or any tasks need to be added, it can be done after consultation between the clients, the project manager, and the scrum team. Since the WBS is the initial breakdown of the tasks, it can only sometimes be considered the perfect set way. Other methods dive even further deep into organizational planning. One such method is using a PERT chart which will be explained in the project schedule section.

Figure 10

Work Breakdown Structure



2.4 Project Resource Requirement and Plan

2.4.1 Hardware Requirements

This section provides an overview of the base hardware configurations and setup required to smoothly carry out the duties regarding the data collection, processing, modeling, and evaluation phases of the project. The project's foundation was laid using local macOS/Windows

client computers, which were used for the project's development. Data preparation and understanding duties were carried out on Apple and Windows Clients, which needed an X86 64-bit CPU and a minimum of 8GB of free system storage for unprocessed data, processed data, and processing software. Jupyter Notebook was used for modeling tasks, connected via macOS and Windows clients. The macOS/Windows clients were used for communication and documentation, and an additional 1GB of Google Drive storage was needed to store all project materials. The project's hardware specifications are shown in figure 11.

Figure 11

Hardware Requirements

Hardware	Configuration	Purpose
Windows Local Client (2x)	X86 64-bit CPU, minimum 8GB available system storage, Internet connection	Data Collection, EDA, ETL, Connecting to Jupyter documentation, communication
macOS Local Client (2x)	X86 64-bit CPU, minimum 8GB available system storage, internet connection	Data Collection, EDA, ETL, Connecting to Jupyter documentation, communication
Google Drive	3GB available storage	Model Input Data, Jupiter Notebook and Documentation Storage
Modem	DOCSIS 3.0 and 32x8 Channel Bonding Up to 1Gbps	Connecting to the internet

2.4.2 Software Requirements

All macOS/Windows clients utilized the most recent release of the corresponding OS and a browser that could accommodate Jupyter notebooks. Chrome, Edge, and Safari were the used web browsers. Models were created using Scikit-learn packages and Python 3.9.

Using NumPy and Pandas, data transformation, wrangling, and ETL testing were carried out. With the help of the Matplotlib and Scikit-learn packages, visualizations for the exploratory analysis and model accuracy validation were produced. The minimal software requirements are listed in figure 12.

Figure 12

Software Requirements

Software	Version	Purpose
macOS	macOS 10.15 (Catalina) or higher	Client OS
Windows OS	Windows 10	Client OS
Chrome Browser	106 or higher	Zoom, Jupyter Notebook
Safari Browser	15.6 or higher	Zoom, Jupyter Notebook
Microsoft Edge Browser	106	Zoom, Jupyter Notebook

2.4.3 Tools and Licenses

The tools that were required to carry out the project with their respective licenses are discussed in this section. Open-source licenses that were installed from Anaconda are Python and Jupyter Notebook. On the local client's system, Jupyter Notebook will be used for data preparation and interpretation. Free services used for the model building include Jupyter Notebook and Google Drive. Project documentation was also stored in Google Drive. Lucid Charts provided free services to help with project task management and visualizations. Zoom was used for video conferencing for direct communications, while Office365 programs with student licenses were used to record project documentation. These requirements are listed in figure 13.

Figure 13

Tools and Licenses

Tool	License	Purpose
Anaconda	Open Source	Python 3.9, Jupyter Notebook Installation
Google Drive	Free	Data, Notebook, Documentation Storage
Jupyter Notebook	Open Source	EDA, ETL Environment
Diagrams.net	Free	PERT Chart Creation
Lucid Chart	Free	WBS Creation
Office 365	Student	Documentation, Presentation Creation
WhatsApp	Free	Direct Communication
Zoom	Student	Meeting
Tableau	Student	Creating Visualizations

2.4.4 Project Cost and Justification

The project's cost implies the amount spent associated with the various technical requirements and other criteria. Most of the resources were either previously paid for before the project was commissioned or offered as free services or open-source software. Some commercial software, like Office365 and Zoom, were covered under student licensing. The minimum cost expected to complete this project would be around 1000 dollars. Figure 14 shows the cost breakdown for the necessary hardware and software. In Figure 13, the word 'Free*' represents an expense paid off through student privilege.

Figure 14

Project Cost and Justifications

Resource	Duration	Cost
8GB RAM	4 Months	$27
64-Bit Intel i7 processor	4 Months	$470
500 GB Hard Disk	4 Months	$50
Adobe	3 Months	Free*
Anaconda	4 Months	Free
Broadband Connection	4 Months	$80
Diagrams.net	3 Days	Free
Browsers (Chrome, Safari , Edge)	3 Months	Free
Google Drive	3 Months	Free
Jupyter Notebook	3 Months	Free
Lucid Charts	5 Days	Free
Modem	4 Months	$150
Python Libraries (Matplotlib, Numpy, Pandas, Scikit-learn)	3 Months	Free
Office 365	4 Months	Free*
macOS(11.0)	4 Months	$250
Windows OS(10)	4 Months	$130
Zoom	4 Months	Free*

2.5 Project Schedule

Proper scheduling of the tasks makes the tracking process more convenient. It also helps keep the project on track and ensures that any roadblocks that occur have ample time to handle

and do not end up as a risk. Hence it is necessary to schedule the tasks diligently along with the assignee tagged. A PERT chart was chosen as the apt representation and will be explained below.

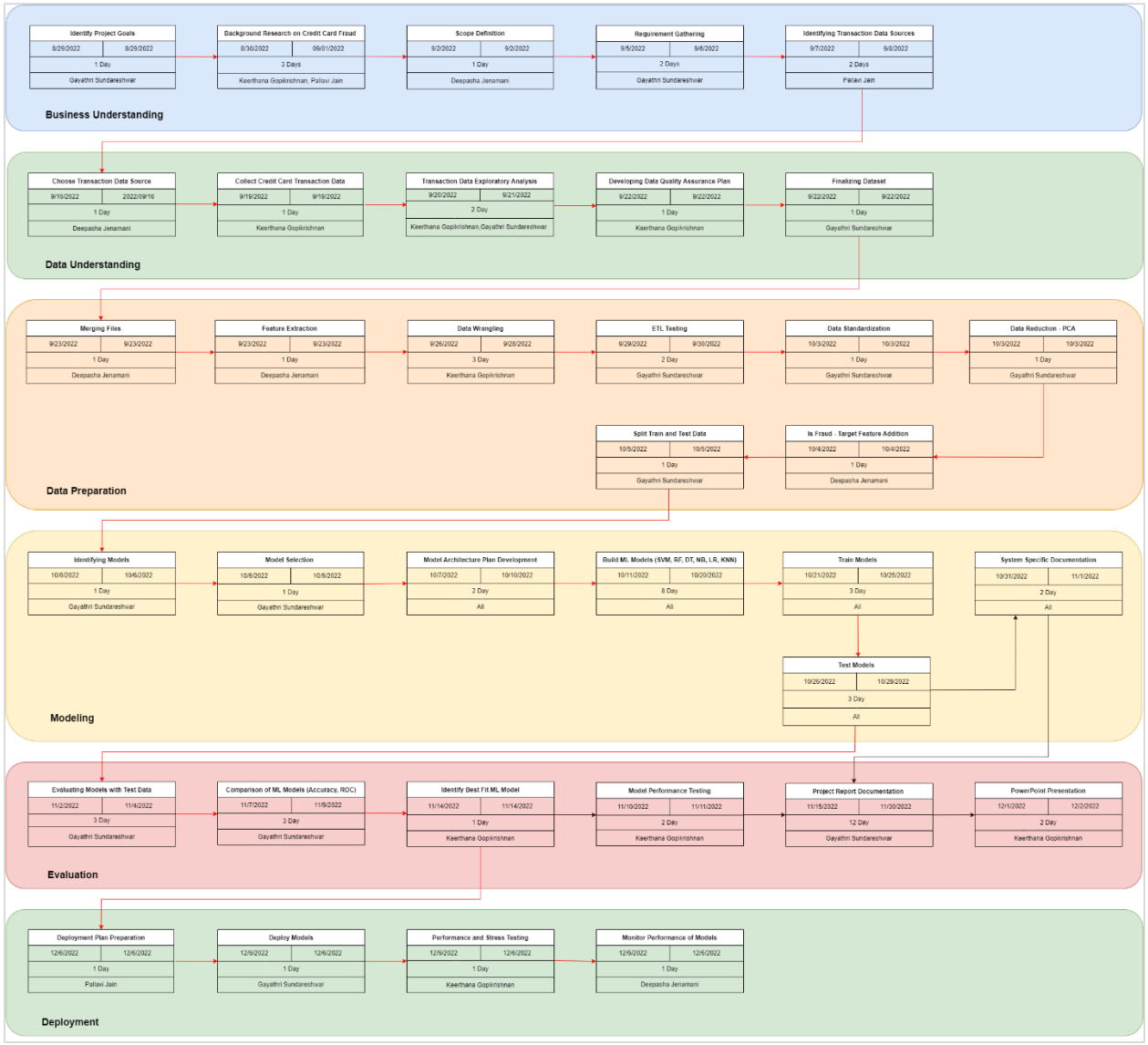
2.5.1 PERT Chart

A Project (or Program) Evaluation and Review Technique (PERT) chart is a visual project management tool. It can be explained as a more profound representation than a WBS as it breaks down even into smaller tasks with dependencies and timelines. This chart illustrates all the tasks and the time or duration required to complete the project. In addition, this chart assists in organizing, coordinating, and scheduling the tasks involved in a project. A PERT chart represents tasks and dependencies between the tasks. Each task was depicted in a box, and arrows illustrated the dependencies. In addition, the arrows were marked in a different color based on the criticality of the task. A PERT Chart also names the person associated with the task; this helps properly plan the workload distribution among the teammates. Figure 10 depicts the PERT Chart created for this project.

Figure 15 above depicts a PERT chart for the project "Credit Card Fraud Detection," and each lane shows six phases. PERT chart provides a birds-eye view of the entire project. Each box in the figure represents a task and other required details such as name, estimated start and end date, duration, and the assignee. The figure shows some arrows between the tasks in red color, and it denotes the critical path. This critical path depicts the route to take for project completion in a shorter span. In addition, specific tasks do not fall under the critical path, and these tasks do not hinder the continuation of the subsequent tasks. Also, tagging an assignee to the task helps keep track of the workload assigned to the team members. This way, extensive workload allocation can be avoided. Unlike a WBS, the PERT chart is even more condensed, which is why most of the tasks created fit under a small-time frame.

Figure 15

PERT Chart



3. Data Engineering

Data is a significant factor that can be persuasive while making business decisions. An sourced collection of information regarding the topic assists in identifying the patterns, leading to effective business decisions that can influence future performance. However, to handle or make

critical business decisions, vast amounts of data are available to refer to and analyze. Thus, data engineering comes in handy, which is designed to support this process. It makes the data usable for further analysis in a dependable, faster, and secure manner.

3.1 Data Process

An appropriate process was devised and followed throughout the data collection and preparation process, considering the importance of data. The efficiency of a solution is proportional to the volume of data collected. Thus, accurate and appropriate data, keeping the project's objective in mind, must be collected to maintain the task integrity and thus helps in achieving the desired outcomes. Consequently, an efficient collection plan must be devised and employed to achieve the goal. As the first step, a coherent strategy was devised and executed for the data collection process.

Therefore, a dataset with relevant features must be selected to achieve this goal, such as the card and cardholder's details, location details, and merchant information. An appropriate data collection plan was devised to collect data with these characteristics and generate derived features. In addition, this plan also aided in keeping track of the data's characteristics, such as the metrics, datatype, and size. Thus, a data collection plan was crucial as it involved steps such as identifying apt data sources, the time frame for the data to be collected, and collection methods. Additionally, transformations were performed to derive desired features that could influence the model's performance and prediction accuracy.

The collection process generates a dataset containing all the essential and optional features. However, this collected data is often raw and unclean, and thus, this data may have specific issues, such as missing values, inconsistency, or incompatible data types. If these discrepancies are not handled, they could influence the model's prediction and lead to an incorrect conclusion. Thus, a

cleaning process was performed before it was used in subsequent steps to handle such discrepancies.

Once the collection process was successfully executed, it was crucial to visualize, analyze, and summarize the essential characteristics of the dataset. An initial exploratory analysis helps understand the data better, and it also assists in uncovering any extremities in the dataset that were not identifiable through eyeball validation. Exploratory data analysis involved creating visualizations and displaying statistical information, which made it easier to discover the data distribution, patterns, correlation, and any anomalies if present. The analytical tasks include a statistical display, univariate or multivariate analysis, and dimensionality reduction. All these analyses lead to a better understanding and attain more valuable insights.

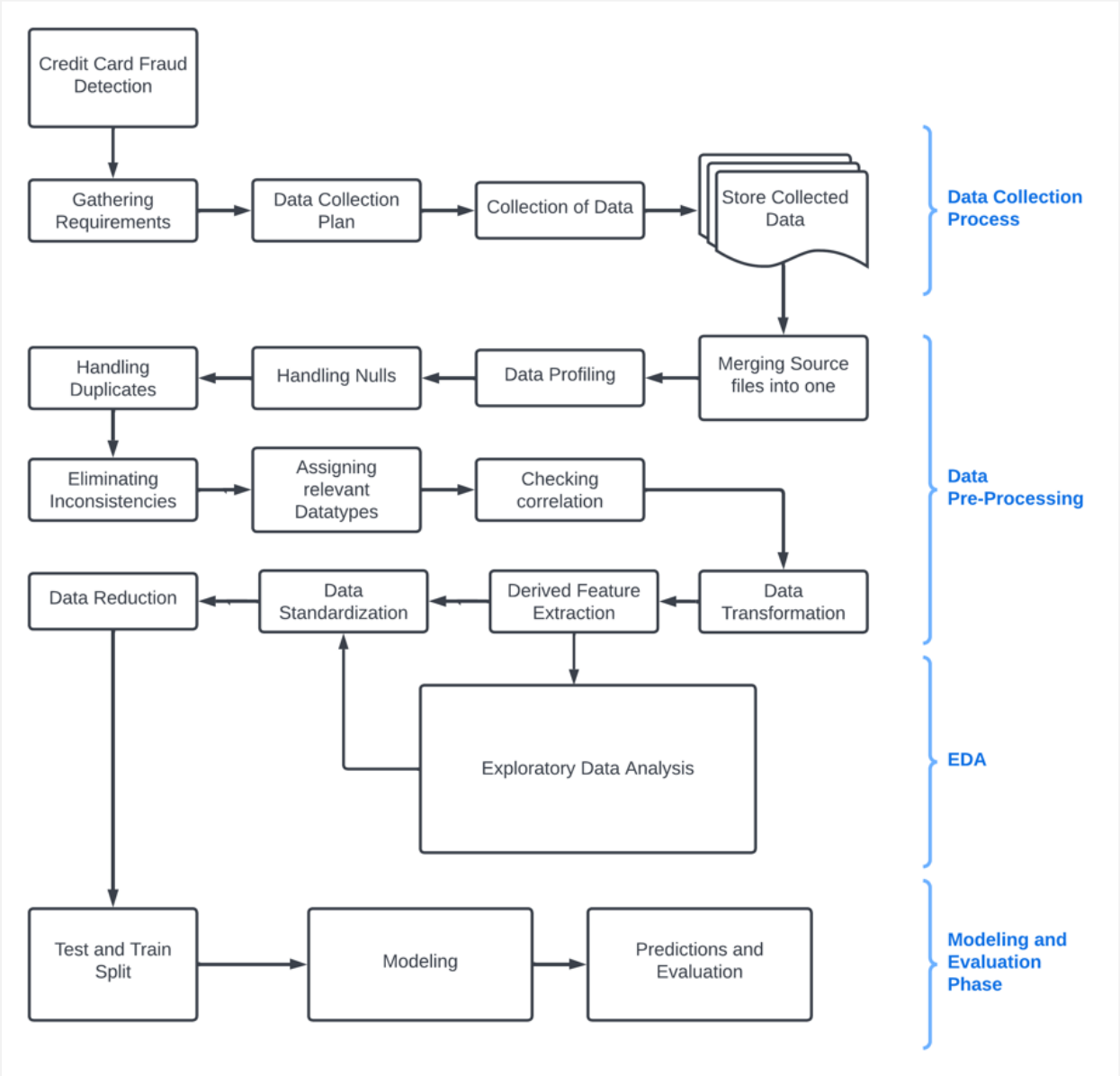
After finding out the extremities and abnormal values in the dataset by performing the steps mentioned above, all the data quality issues were handled appropriately. Although the data was processed and clean, it was necessary to identify if there could exist any quantifiable information that could be derived from the existing features, which would help enhance the model's prediction. However, before feature extraction, it was essential to look for ways to gain additional insight by performing some required transformations. It helped in gaining valuable insights that could help in the model's prediction and also in finding any mismatch in the data format, data type, and consistency between the features. In addition, it was also necessary to check if it needed any data standardization, normalization, or additional aggregation. Thus, an appropriate transformation strategy was planned and implemented.

After the data collection, cleaning, transformation, and analysis process, the resultant data will now be more concentrated. Though pure, it is not viable to say that this data was now qualified for further use in the subsequent modeling phase. Hence, an approach was designed to use the data

effectively to evaluate the model's performance. A technique known as the train-test split was used where each set suitably represents the problem domain. The division was done based on the project's goal and considering the computational cost. The optimal split size used here was 80% for training and 20% for testing. Figure 16 depicts the workflow diagram of the data process.

Figure 16

Workflow Diagram with Detailed Data Processing Steps



3.2 Data Collection

Establishing the data collection requirements is mandatory since it plays a crucial role in identifying the relevant features while collecting data. The critical factors and features were identified after having multiple brainstorming sessions with the business and among the team members. Based on these identified key factors, the requirement step was conducted. It is also necessary to stay compliant with the policies, compliance, and guidelines set by the data source owner during the collection process. Additionally, adhering to the privacy act was vital to ensure that a user's personal information was not put at risk and was adequately protected while using it. The data used for this project was retrieved from Kaggle, which is accessible under a public domain license.

The project aims to identify fraud in credit card transactions; thus, accurate and complete data was required to achieve the outcome. Therefore, the collected data must have certain features related to card details, card holder's demographic information, and merchant details. Emphasis was put on the quality of data, the measures, and the volume of the dataset with unbiased records. Hence, the data collected was sufficient and had the required features to make the model's prediction accurate. Additionally, the metadata of multiple collected datasets, such as the number of columns, and the type, could mismatch. Therefore, the metadata of multiple files associated with a source was checked and validated during the collection process.

The dataset selected for this project contains simulated transactions related to credit cards. The transactional data involved are of primarily two types: legitimate and fraudulent. The duration of this transaction showed that it started on 1st January 2019 and ended on 31st December 2020. The number of customers involved was 1000, and the number of merchants was 800—the resultant files generated after this simulation process were combined and converted into a standard format.

Two datasets were available to retrieve from the source; one was for training, known as the "fraudTrain" dataset, which has 1296675 records and 22 features. The other one was a testing dataset with the same set of 22 features and a 555719 record count. The multiple files were merged into one before processing to eliminate the possibility of bias and overfitting. The detailed associated steps will be explained in the forthcoming sections.

3.3 Data Pre-Processing

A crucial component of data management is data cleaning. The project's objective needs to be met for which data was gathered in various ways, including manually through surveys, interviews, and phone calls. Other techniques include tracking (online or transactional) and online media channels like social media and marketing. For this research, which involved detecting credit card fraud, data was gathered over a year using Sparkov Data Generator.

The acquired data may be unreliable, incomplete, redundant, or inconsistent, resulting in incorrect modeling predictions and the failure to meet the project's objectives. Data cleaning entails examining the data to update or eliminate outdated ones, ensuring a higher quality of the data utilized in the following phases. Numerous processes are used in data pre-processing to remove discrepancies in the extracted raw data. Thus, a thorough pre-processing procedure was carried out with extreme care, as described in the following section.

3.3.1 Merging and Original Features Descriptions

The raw dataset was merged into one file and converted into an apt format before executing the exploratory data analysis. Therefore, further pre-processing is applied to the data collection. The dataset's .csv files contain a list of each customer's credit card transactional information that was time-stamped. It was essential to merge the file to work on the subsequent stages without bias that might have previously existed and can be used to gain insights. Figure 17

below shows an example of this combined file, and Figure 5 shows the fields and associated types in the generated file.

Figure 5 depicts the fields present in the dataset; the first field, "trans_date_trans_time" represent the date and time the transaction happened. The second field, "cc_num," shows the customer's credit card number. The third field merchant lists the person involved in the sale. The field first, last, gender, street, city, zip, and dob represents all the details of the customer making the transaction. Lat and long fields record the customer's latitude and longitude from where the purchase was made. A similar set of fields was added for the merchant's coordinates. Lastly, the field "is_fraud" shows whether the given transaction is legitimate.

Figure 17

Workflow Diagram with Detailed Data Processing Steps

	Unnamed: 0	trans_date_trans_time	cc_num	merchant	category	amt	first	last	gender	street	...	lat	long	city_pop	job	dob	trans_num	unix_time	merch_lat	merch_long	is_fraud
454545	454545	2019-07-20 21:26:57	4364010865167176	fraud_Brown, Homenick and Lesch	health_fitness	11.91	Gary	Martinez	M	03512 Jackson Ports	...	39.5483	-119.7957	276896	Immunologist	1997-03-12	08098d93c85203bbe81333a65a9c52ff	1342819617	39.215426	-119.002679	0
170805	170805	2020-06-19 23:18:58	2712209726293386	fraud_Dooley Inc	shopping_pos	7.34	Jenna	Brooks	F	50872 Alex Plain Suite 088	...	30.4066	-91.1468	378909	Designer, furniture	1977-02-22	dd0844b6e5601f014720a28046432e80	1376954338	30.618718	-90.613614	0
876702	876702	2019-12-21 18:12:00	630451534402	fraud_Leffler-Goldner	personal_care	31.32	Rachel	Daniels	F	561 Little Plain Apt. 738	...	46.3535	-96.6345	765	Immunologist	1972-06-12	cf7db3740043c3b7e7d1ffcd71e22bcf	1356113520	45.852503	-95.740663	0
737297	737297	2019-11-11 02:53:00	3572976568630035	fraud_Buckridge PLC	misc_pos	7.10	William	Lopez	M	785 Kevin Walk Suite 237	...	41.1832	-96.9682	614	Associate Professor	1967-06-20	49c98c902ba30d92d99e9caff72010c8	1352602380	41.448385	-97.311391	0
1022208	1022208	2020-02-27 03:46:41	4149238353975790	fraud_Bins-Rice	gas_transport	70.23	Tanner	Carroll	M	494 Burke Ports	...	40.1008	-80.0652	632	Dealer	1989-04-08	ab83238f6fcbfcb92a2f6bd39c2582aa	1361936801	39.494226	-79.890573	0

5 rows × 23 columns

3.3.2 Handling Incomplete/Missing Data

Many factors, including human mistakes or mechanical problems, might result in missing or partial data. When data is gathered through surveys, a respondent may need assistance comprehending the questionnaire, there may not be a suitable response option, or the respondent may not be interested in responding to a particular topic. In other situations when using instruments is the collection method, there can be a problem with the machine recording the observation, the user not using it during the observation period, or a defective instrument that

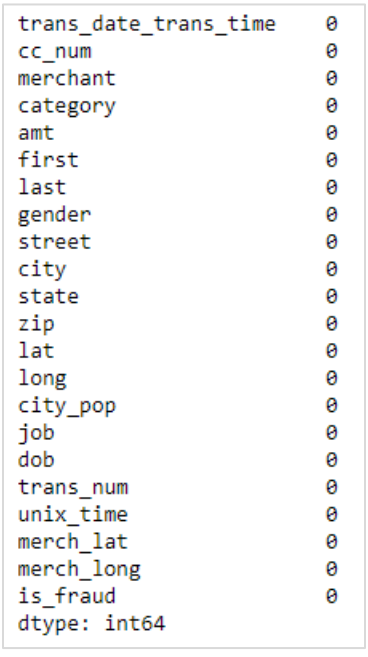
needs to be fixed. Therefore, a proper approach was implemented to handle the difficulties described in the next phase to prevent any misleading results brought on by these forms of data.

3.3.3 Checking for the Presence of Nulls and Duplicates

As indicated in the previous section, the dataset needed to be examined for nulls. Due to the vast number of people who have been identified and the enormous size of the dataset, some features may have null values. There are many ways to deal with null values, including deleting the associated rows or columns if more than 50% of the data is null, replacing the numbers with the central tendency factor (mean, median, or mode), or applying various interpolation techniques. The output of this function call is depicted in Figure 18rd below, showing that the dataset contains no null values in any place..

Figure 18

Null Check Result Showing the Number of Null Records in Each Feature

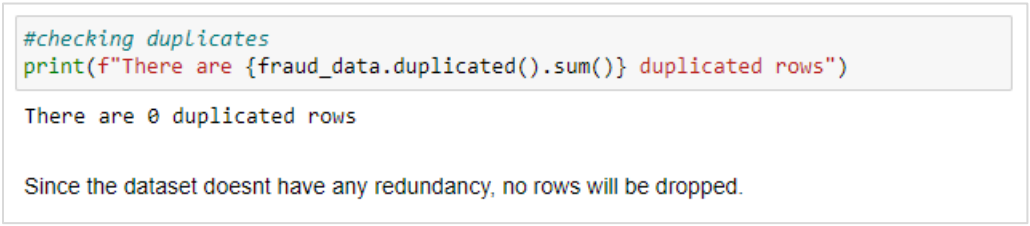


A frequent problem with data quality is redundant or duplicate data. It can happen for many reasons, including inaccurate entry of precise user details and several representations of the

same data. If this redundant information is not found and eliminated, the model will eventually become overfit. Although it will not generalize any better for the new test dataset, it will do well with the training data. Additionally, it can interfere with the validation process. Suppose the training set and validation set include the same data. In that case, the performance of the validation set will be better since the model has already learned about the validation set's data during training. In order to prevent the issue, it was essential to remove the unnecessary records during the cleaning step. Pandas library was used to check for duplicates, and the findings indicate none in the dataset, and this could be because the data was generated in a simulator. Figure 19 showcases that the data contained no duplicates.

Figure 19

Checking for Duplicates



3.3.4 Checking for Outliers in Latitude, Longitude and Amount

The collected credit card transaction contains each customer and merchant's location information that was part of a single transaction. A latitude is invalid if it does not fall between zero and 90 degrees, and similarly, a longitude that does not fit in the range of 180 or -180 degrees is considered incorrect. A test to single out the abnormalities in latitude and longitude returned no records. The reason for the inexistence of outliers could be attributed to the way the data was generated. The data points were generated through a simulator; hence there were no outliers in the dataset. There were cases where these abnormalities skewed the predictions; hence it is all right to be cautious even though the test yielded no change in the dataset.

3.3.5 Checking for Outliers in Amount

The field amount in the dataset logs the money charged by the merchant in each transaction. The values of this field can be negative or too high than usual. The negative values present in the dataset should be removed as it is not possible to have such values. The amount spent by specific customers can be higher than the usual range; hence specific criteria need to be set before any action is taken on higher data points. The chosen credit card transaction data does not have values greater than 23000 dollars. The data point in this feature had significantly fewer records where the transaction amount was above 1000 dollars.

The pre-processing phase did not affect the dataset much due to the precise way the data was collected. However, there is always room to see what derived attributes can be added to make the dataset even more optimal. An initial exploratory data analysis was necessary to identify such influencing factors, and the process involved will be discussed in the next section.

3.4 Initial Exploratory Data Analysis

The raw features from the source file have some correlations. The features can be utilized to the maximum of their potential once an in-depth analysis is performed to understand the existing patterns to a greater extent. Hence, an initial exploratory analysis was performed before using the raw features in the modeling phase. Through this process, the vital factors that can significantly skyrocket the model's performance could be identified. Some of the usually performed tasks associated with EDA are analysis of the distributions, statistical analysis, and visualization using various graphical features. Some of such visualizations created will be examined in this section.

Figure 20 below portrays a bar chart depicting the transaction category based on a percentage difference, which helped identify the category that posed the most fraudulent transactions. Figure 21 below shows a geospatial map representing the number of fraudulent

transactions in each state. This helped in understanding the concentration of transactions along with their fraudulent ratio. The age distribution of the recorded transactions categorized based on legitimate and fraudulent transactions is shown in Figure 22. A vital point worth noting was that older people tend to be more naive towards the transactions, which resulted in a staggeringly obvious ratio. It was also essential to understand the distribution of transaction amount, gender, and category of the fraudulent transaction, which is represented in Figure 23. The data used here had features related to the time of the transaction and understanding the distribution of these features could help gain insights into the trend of the fraudulent transaction. This trend analysis based on the time-related features is represented in Figure 24.

Figure 20

Fraud Vs Non-Fraud Transactions based on Spending Category

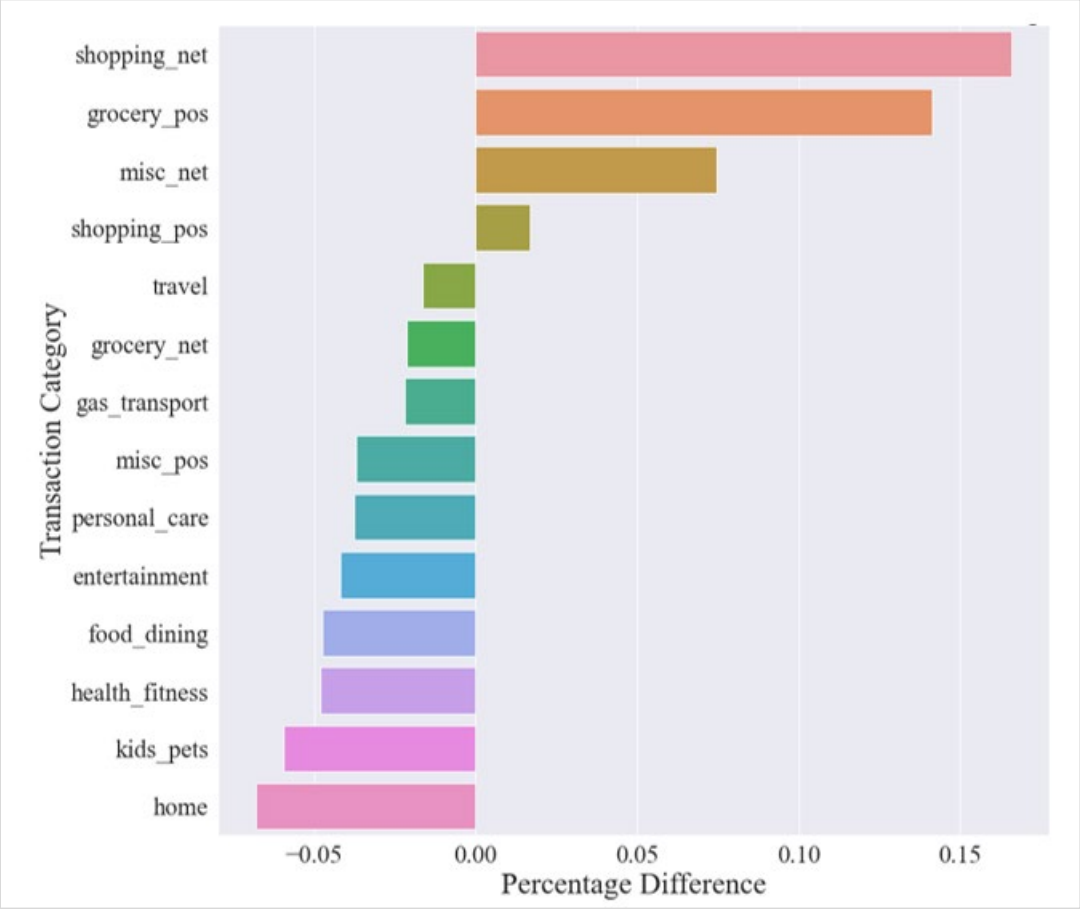


Figure 21

Geo-spatial Distribution of Fraudulent Transactions

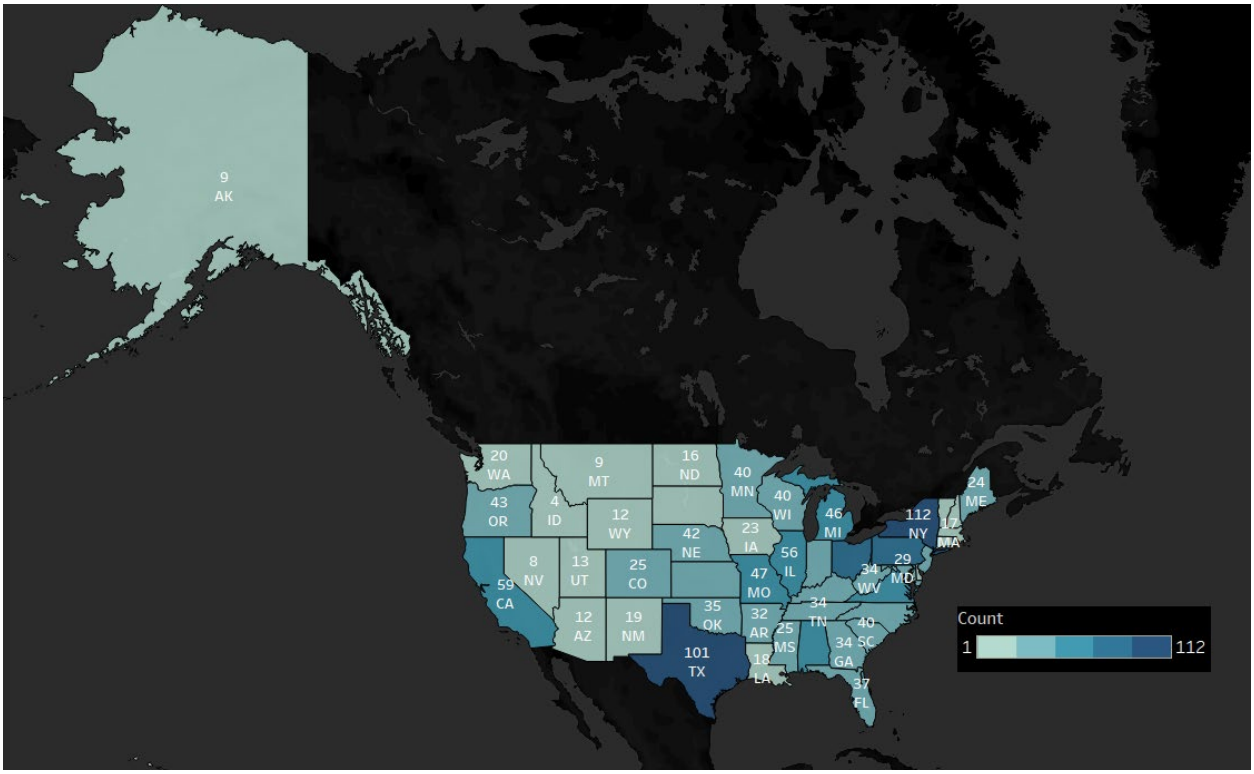


Figure 22

Age Distribution in Fraud Vs Non-Fraud Transactions

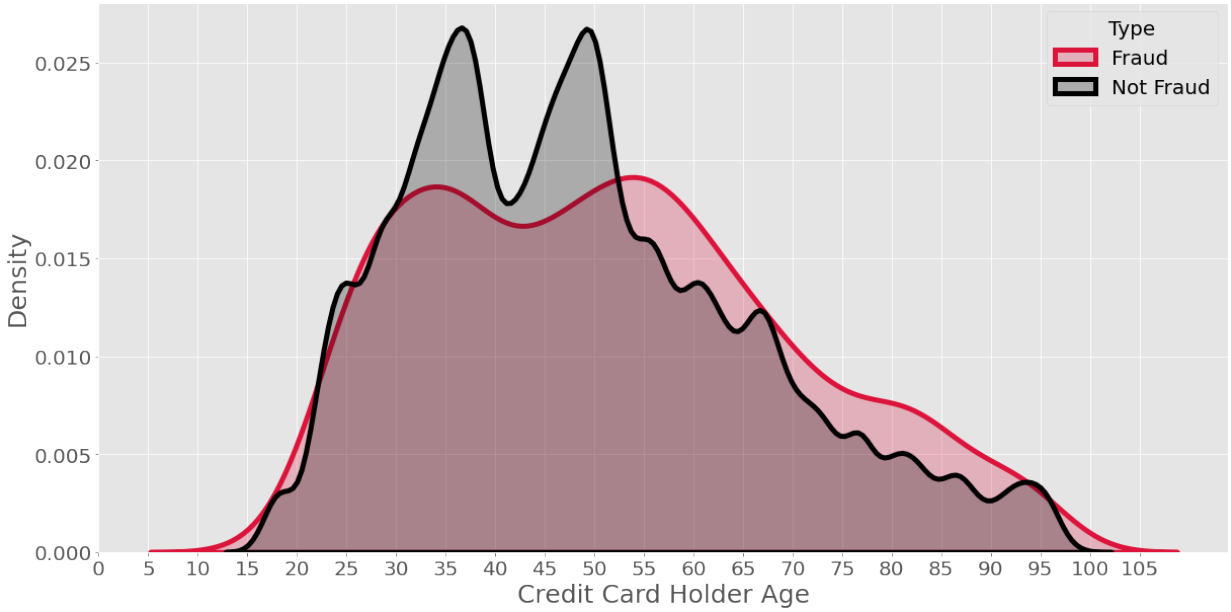


Figure 23

Distribution of Transaction Amount, Gender, and Category in Fraudulent Transactions

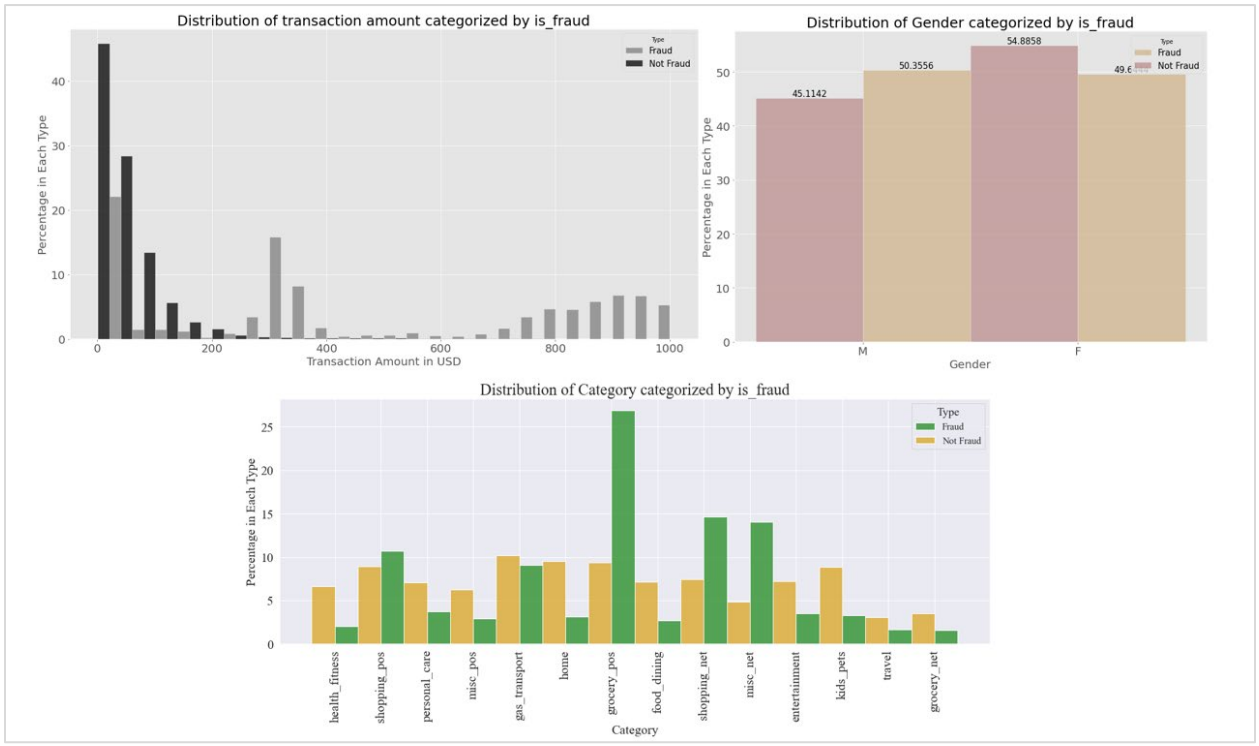
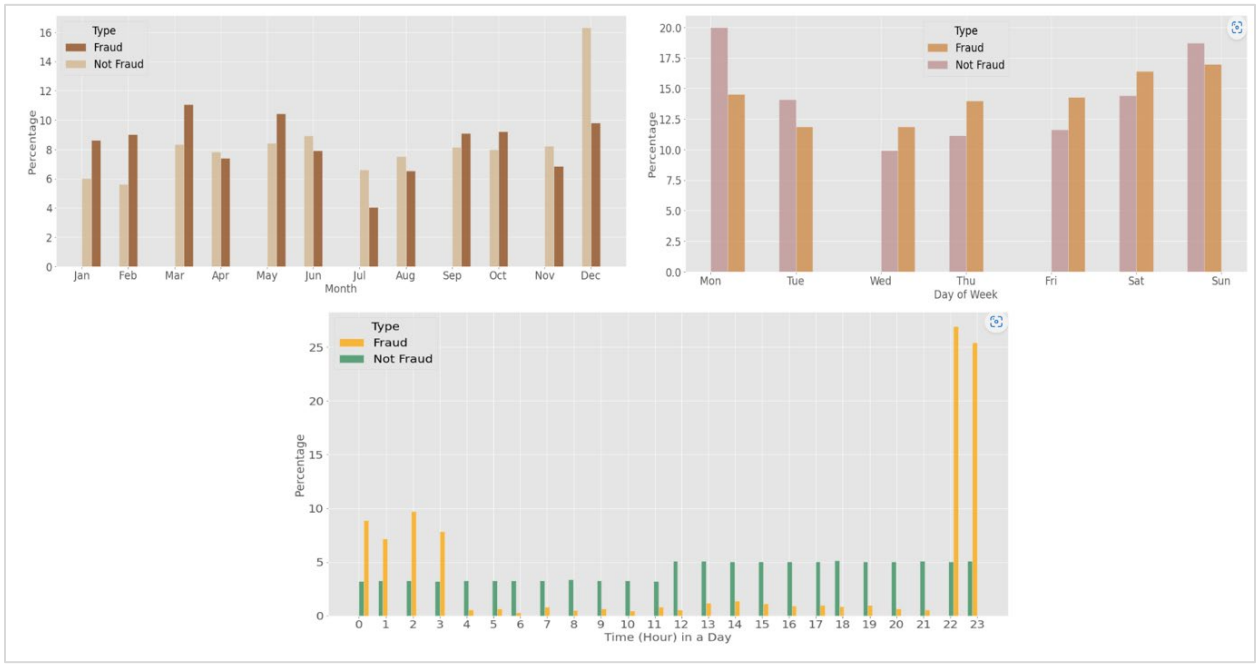


Figure 24

Trend Analysis in Fraudulent Transactions



3.5 Data Transformation

Identifying the essential features and deriving meaningful insights to make them usable across all systems is essential. Hence, designing and implementing an appropriate data transformation strategy becomes crucial to achieve this goal. The transformation process can be performed on the data's structure, format, or values. The steps involved in this process can vary from business to business based on the end goal. In some cases, a format conversion of a file is required, whereas others can utilize the same without any modification. In addition, the dependency on a transformation step is primarily because of the data volume. Therefore, a transformation process was implemented to maximize the data value and aid in managing and reducing the information overload, which the massive amount of data could cause. Thus, keeping the mentioned above points in mind, an appropriate transformation strategy was planned and executed, and the tasks performed are explained in detail below.

3.5.1 Checking the Consistency of Decimal Places

The number of decimal places for the decimal values can differ from one feature to the other. Data should be rounded off properly considering this characteristic. However, the rounding off should not be too much or too little. Because if it is not done correctly, some valuable information may not be captured. Thus, a strategy was followed to maintain the same number of decimal places for all the decimal numbers. There were five decimal numbers in the dataset which needed attention. The first three features are related to the amount spent, latitude, and longitude of the card.; the other two are related to the merchant's location information(latitude and longitude). Generally, the location-related information was stored with 15 decimal digits right to the decimal point. Thus, there was a possibility of a mismatch in the decimal places while recording these values, which could lead to inconsistency issues. The

resulting conclusion could have a different format if this inconsistency is not handled. Therefore, for both latitude and longitude, four decimal places were allowed. Similarly, two decimal places were allowed across the dataset for the amount spent.

3.5.2 Verifying Data Types:

The data type of the features specifies what kind of values are acceptable. Various data types can be used to represent each feature, such as string, integer, float, and DateTime. The dataset used for this project has different features containing information related to location, transaction time, date, and cardholder demographics. All of these features mentioned above have different types of values which serve a specific purpose. If a mismatch exists, that could lead to an incorrect feature being derived in the extraction process. Therefore, the location-related features, such as latitude and longitude for both credit card and merchant, and amount spent were maintained in float data type. The other required details, such as demographic details(gender, age, name, address) of a cardholder and category, were maintained as a string.

3.5.3 Derived Features Extraction:

The selected dataset contains the relevant features; however, it was crucial to determine what insights could be derived in addition to the existing ones for a better understanding of the data and to achieve a model with better prediction accuracy. Thus, deciding on the features that could be collected from the raw dataset was necessary. While deriving the additional features, some key factors needed to be considered, such as how it could make the data more quantifiable and how beneficial it was to enhance the models' prediction. The derived features were generated to aid in detecting fraudulent transactions effectively.

This section overviews such derived features. The first additional feature was the age which was calculated based on the date of birth data present in the dataset. Three features were

derived and added based on the transaction date and time. The first one was the hour of the transaction, the second one was the day, and the third one was the month of the transaction. Additionally, different categories were present in the dataset, which denote the category for which the transaction was made. However, many different categories could not help in the binary classification of fraud or not fraud transactions which is the goal of this project. Thus, different dummy variables or features were added to the dataset based on the type of categories present. This addition helped train the model better and get a better classification, leading to accurate prediction accuracy. Figure 25 depicts the derived features created in this step.

Figure 25

Sample Records with Dervied Features

	amt	zip	lat	long	city_pop	merch_lat	merch_long	age	hour	day	...	category_grocery_pos	category_health_fitness
454545	11.91	89512	39.5483	-119.7957	276896	39.215426	-119.002679	25	21	5	...	0	1
170805	7.34	70808	30.4066	-91.1468	378909	30.618718	-90.613614	45	23	2	...	0	0
876702	31.32	49895	46.3535	-86.6345	765	45.852503	-85.740663	50	18	5	...	0	0
737297	7.10	68626	41.1832	-96.9882	614	41.448385	-97.311391	55	2	0	...	0	0
1022208	70.23	15324	40.1008	-80.0652	632	39.494226	-79.890573	33	3	3	...	0	0

category_home	category_kids_pets	category_misc_net	category_misc_pos	category_personal_care	category_shopping_net	category_shopping_pos	category_travel
0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0
0	0	0	0	1	0	0	0
0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0

3.5.4 Dimensionality Reduction

As a part of this study, both the supervised and semi-supervised models were planned to be executed. Hence, Dimensionality Reduction was included as a part of semi-supervised learning. Dimensionality is described as the number of features present in a dataset. It is indisputable that too many features could skew the outcome of a model's prediction. Thus, it is necessary to identify and keep only the significant features for better prediction accuracy of the model.

Identifying these significant features helps decide whether to include or exclude any specific feature in the subsequent phase. Additionally, the reduction helps minimize the model's overhead when dealing with larger volumes of data. However, considering the dataset's characteristics, a decision should be made on whether to implement this technique. After carefully analyzing the dataset used in this case, it was decided to use a dimensionality reduction technique which could help in the modeling phase of semi-supervised execution. Figure 26 depicts the sample set after performing dimensionality reduction using PCA.

Figure 26

Sample Results of PCA

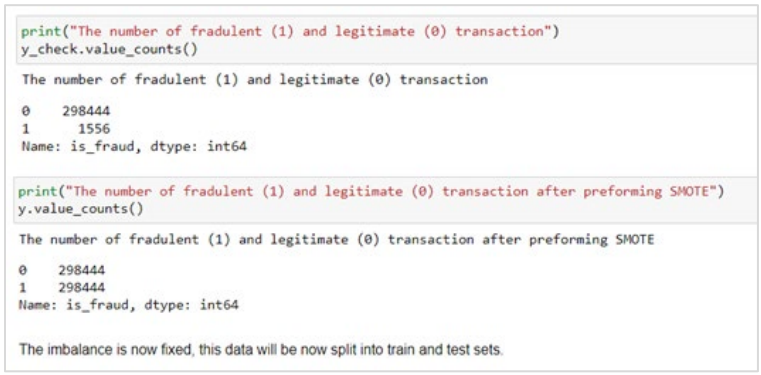
	Dimension 1	Dimension 2	Dimension 3
0	-87563.378808	-45255.416549	96.197083
1	-86653.502654	8512.379884	-22.588324
2	-80796.562355	37819.531487	-60.130184
3	-88166.512016	-15394.913020	42.467375
4	-88129.294397	-16415.675293	-13.871488

3.6 Data Preparation

The previous sections overview the essential pre-processing and transformation steps to ensure that the raw dataset containing inaccuracies was handled and required additional features were derived before passing the data on to the modeling phase. The dataset that was the outcome of all the previous phases was then passed on to the modeling phase. Before actually beginning with the modeling part, it was necessary to eliminate any bias that exists in the resultant clean dataset. Hence, a count check was performed to check for the bias presence. Figure 27 depicts the change in categorial record count after eliminating the bias using a Synthetic Minority Oversampling TEchnique(SMOTE). Figure 27 shows that before using SMOTE, the fraudulent record count was staggeringly lesser than non-fraudulent ones, which SMOTE rectified.

Figure 27

Record Count Before and After SMOTE



Following the SMOTE application, the next step would be to find the optimal test and train split of the data. This section describes the data ratio which needs to be considered for training and testing. For the prediction to be accurate, the modeling phase required an optimum test and train data split as the initial step. Appropriate machine learning models were identified before splitting the dataset depending on the data distribution and the existing pattern. Thus, keeping this idea in mind, six models were considered efficient and effective in fraud prediction, which was: Logistic regression(LR), Decision Tree(DT), Random Forest(RF), Support Vector Machine(SVM), Gaussian Naive Bayes(GaussianNB) and k-Nearest Neighbor(KNN). The possibility of other suitable models was also viable, but the model selection was made here considering the system compatibility. An optimal train and test split ratio were chosen, which was 80:20. The reason behind choosing this ratio was the learning process. Setting aside a higher percentage of records for the training phase ensures the model is well trained and the accuracy of the test could reflect the result.

3.7 Data Statistics

In this section, various characteristics of the features present in the dataset were statistically examined. As the initial step, an analytics base table was formed to identify the features that are cordial with the requirement. This is followed by formulating a cardinality table to analyze each feature's statistical distribution of the records. A data quality report helped identify the potential

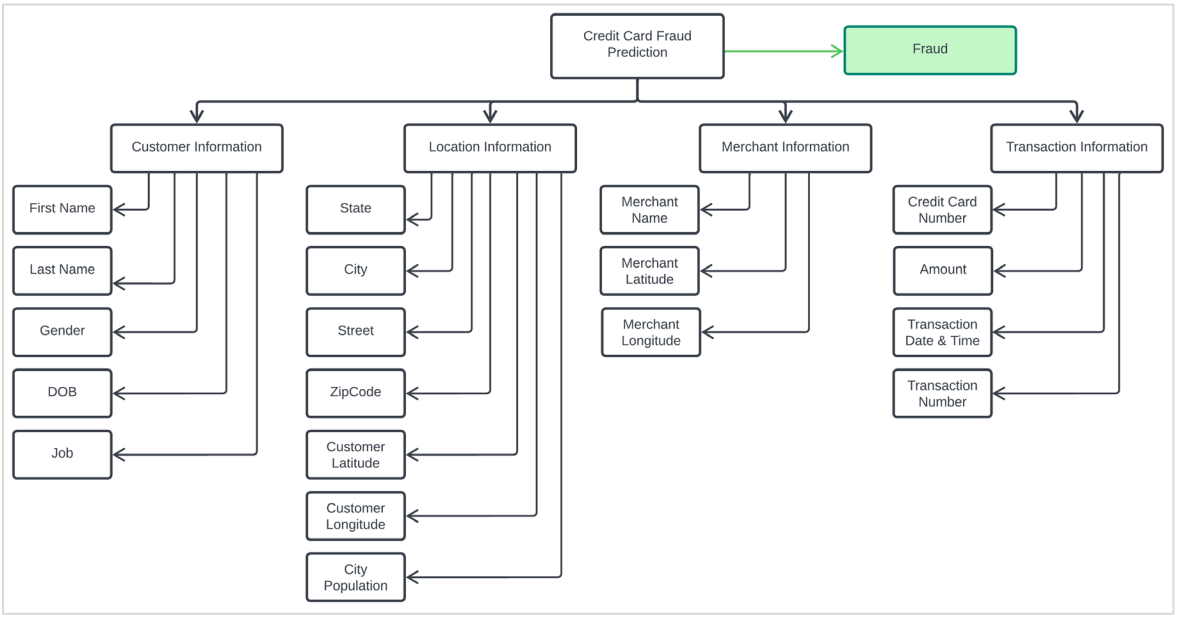
issues that could arise and the ways to handle them. These parts will be explained in detail in this section.

3.7.1 Analytical Base Table

The Analytic Base Table (ABT) represents the classification of features that would aid in increasing the preciseness of the models. An ABT helps realize the features that need to be looked out for either while collecting the dataset through manual means or while choosing an already existing dataset. A correctly done ABT can help with optimal feature extraction. Figure 28depicts the ABT created for the Credit Card Fraud Detection Project and shows that the ABT was initially classified into five sectors, where four represent the inputs and one represents the target.

Figure 28

Analytical Base Table



The five classified domains were customer information, location information, merchant information, transaction information, and target feature. This section will cover a detailed view of the domains.

The customer information domain encapsulated the features that were identified to represent this domain better, such as the customer's first and last name, gender, job, and date of birth, and aggregation can be performed to obtain additional features such as age. The characteristics such as job and gender can add dimension to the prediction model.

The following domain is location information which captures the features related to the geographic information, such as the customer's location details, such as latitude, longitude, city, state, zip code, and city populations. These features will help provide an idea of the customer's base location. Location details would play an important role in fraud detection; for example, if a customer's credit card made a transaction in California at a given time and date, and within one hour, it made another transaction on the east coast of the United States. In the scenario mentioned above, we can note that the second transaction was fraudulent as it is impossible to travel from the west coast to the east coast in under one hour. The features that could provide optimal information to enhance the predictions were added to each domain in the ABT.

The following domain merchant contains information regarding the merchant, such as name and location. Next is the transaction information includes details such as the customer's credit card number, amount, date, time, and transaction number. The amount provides the money exchanged between the customer and the merchant. Lastly, the target feature of the dataset is to predict if it is a fraud; this is a binary output given as 1 or 0.

3.7.2 Data Cardinality

The pre-processing steps were completed to ensure the data used in the model is clean; this will, in turn, increase the model's accuracy in predicting fraudulent transactions. The data cardinality will further understand the data and gain insights about their values. The figure 29 an 30 below shows the descriptive statistics of the numerical and categorical values. The statistics

show count, unique, mean, standard deviation, minimum, maximum, and interquartile range. The interesting points one can infer were, firstly, the feature amount has a minimum value of 1 dollar and a maximum of 22768.11 dollars, whereas the 75% quartile shows that value lies in and around 82.88 dollars. These are not considered outliers and are not removed. As previously mentioned, specific criteria were established not to remove these data points. Secondly, looking at the age feature, there are 42 unique values ranging from 17 to 98.

Figure 29

Descriptive Statistics of the Categorical Features

	count	unique	mean	std	min	25%	50%	75%	max
amt	300000	30525	69.8609725	149.333059	1	9.63	47.3	82.88	22768.11
zip	300000	963	48747.3925	26839.9055	1257	26237	48154	72011	99921
lat	300000	975	38.5491723	5.07738225	20.0271	34.6689	39.3716	41.9404	66.6933
long	300000	976	-90.194591	13.7215953	-165.6723	-96.798	-87.4769	-80.158	-67.9503
city_pop	300000	873	88601.3714	301453.004	23	741	2443	20328	2906700
unix_time	300000	300000	1358678499	15889387.1	1325376018	1347484722	1357058828	1370382813	1388534276
merch_lat	300000	297367	38.5496375	5.11086514	19.033288	34.746879	39.370665	41.9587923	66.835174
merch_long	300000	298924	-90.195139	13.7324859	-166.67124	-96.875379	-87.421291	-80.237806	-66.955996
is_fraud	300000	2	0.0056	0.07462345	0	0	0	0	1
age	300000	82	48.6926	2.15498	17	25.458	50.497	80.444	98

Figure 30

Descriptive Statistics of the Analytical Features

	count	unique	mode	Frequency	2nd mode	2nd mode frequency
first	300000	351	Christopher	6135	Robert	5134
last	300000	482	Smith	6668	Willams	5520
gender	300000	2	F	164057	M	135943
street	300000	983	4034 Smith Avenue	781	0069 Robin Brooks Apt, 695	764
city	300000	894	Birmingham	1280	Utica	1186
state	300000	51	TX	22058	NY	19317
category	300000	14	gas_transport	30185	grocery_pos	28648
job	300000	495	Film/video editor	2221	Surveyor, land/geomatics	2105
dob	300000	1862	3/23/1977	1067	8/29/1981	864
trans_num	300000	300000	9f02854a2cbbbb414e5f03facd55493a	1	20ce54dc46aaf8e27fe5f4c24bded46	1
trans_date_trans_time	300000	291066	11/30/2020 21:15:00 PM	6	12/14/2020 14:47	5
merchant	300000	693	fraud_Kilback LLC	1048	fraud_CormierLLC	880

Figure 30 shows the descriptive statistics of categorical features. The statistics show information on the mode, frequency, count, unique values, and second mode and its frequency of

the features. Some interesting insights were that the feature day and month have the correct number of unique values. Furthermore, features such as job and category have 495 and 15 unique values, respectively, and using these features during modeling will add dimensions.

3.7.3 Data Quality Report

Features extracted from the raw data can sometimes be impure and corrupt. There were so many pre-processing and transformations that were discussed in the previous sections. At the forefront of all of those steps, there exists a need to prepare a data quality report. This report acts as the foundation for how the dataset's quality assurance checks will be carried out. A quality assurance plan or a data quality report can be explained as the steps that are recorded with the intent of being a reminder to be cautious during validations. Each feature might have its own list of pitfalls, and being aware of these flaws will help handle them during the more extended run. Figure 31 and 32 below depicts the data quality report created for the categorical and numerical features

Figure 31

Data Quality Report Containing the Categorical Features

Feature	Data Quality Issue	Potential Issues	Handling Strategies
first	None	Might contain special characters and numbers.	The value must contain only character, check if there are any special characters or number, then remove such values.
last	None	Might contain special characters and numbers.	The value must contain only character, check if there are any special characters or number, then remove such values.
gender	None	The cardinality of this can be off the charts due to the multiple ways the logging was done.	The cardinality of this feature should be only 2 that is male and female. Check and replace different representation of the same gender to one value.
city	None	Invalid city names might be entered and special characters and numbers can be entered	The field contains characters. Check for special characters and remove such values. Should check for the validity of the city name through geo mapping
state	None	Invalid state names might be entered and special characters and numbers can be entered	The field contains characters. Check for special characters and remove such values. Should check for the validity of the state name through geo mapping
category	None	Might contain too many unique categorical values and values that don't make sense.	Too many Categories can make the field messy. So the number of allowed categories must be examined and limited. Special character allowed should also be limited to only a few such as an underscore.
job	None	Might contain invalid information and special characters or numbers	Check for valid jobs should be done and invalid values must be considered for either elimination or replacement
merchant	None	Might contain a string or number but no special characters will not be allowed	We can check for special characters and avoid such data values.

Figure 32

Data Quality Report Containing the Numerical Features

Feature	Data Quality Issue	Potential Issues	Handling Strategies
trans_date_trans_time	None	This field might contain the different time and date formats.	This can be handled by setting the format of the datetime field to 'YYYY-MM-DD HH:MM:SS'.
cc_num	None	Might contain non numerical values of length greater than 12	The maxmimum allowed character limit should be set to 12 and numerical.
amt	None	Might contain a negative value or non numerical value.	The value must be Non-negative, check if all the values are non-negative, if there is a negative value then replace the amount using mean or median.
street	None	Invalid addresses might be keyed in	The values in this field can contain numbers, special characters, and strings. Check if the data has the proper word.
zip	None	Might contain non numerical values of length greater than 6	Contains six-digit numbers. Check to ensure the numbers are non-negative and do not contain characters. If so they need to be removed.
lat	None	Abnormal values out of the latitude range could be entered	The values should be between -90 to 90 in decimal-degree format. Check for values outside this range and need to be r
long	None	Abnormal values out of the longitude range could be entered	The values should be between -180 to 180 in decimal-degree format. Check for values outside this range and need to be r
city_pop	None	Non numerical values can be entered and decimal values can be entered	Should ensure that the field contains only integers
dob	None	Each record could be of a different date format, the date range might not make sense as a date of birth	Valid date of birth should be entered, only date format should be allowed and all records should be off same date format
merch_lat	None	Abnormal values out of the latitude range could be entered	The values should be between -90 to 90 in decimal-degree format. Check for values outside this range and need to be r
merch_long	None	Abnormal values out of the longitude range could be entered	The values should be between -180 to 180 in decimal-degree format. Check for values outside this range and need to be r

The fields in the figures depict four columns: feature, data quality issue, potential issues, and Handling strategies. The feature column contains nothing but the name of the features, and the data quality issue field tracks the issue in the dataset. The potential issues field lists the flaws that could corrupt the data, and the final column lays down the potential solutions for the noted flaws. The data quality report comes in handy when dealing with too many features

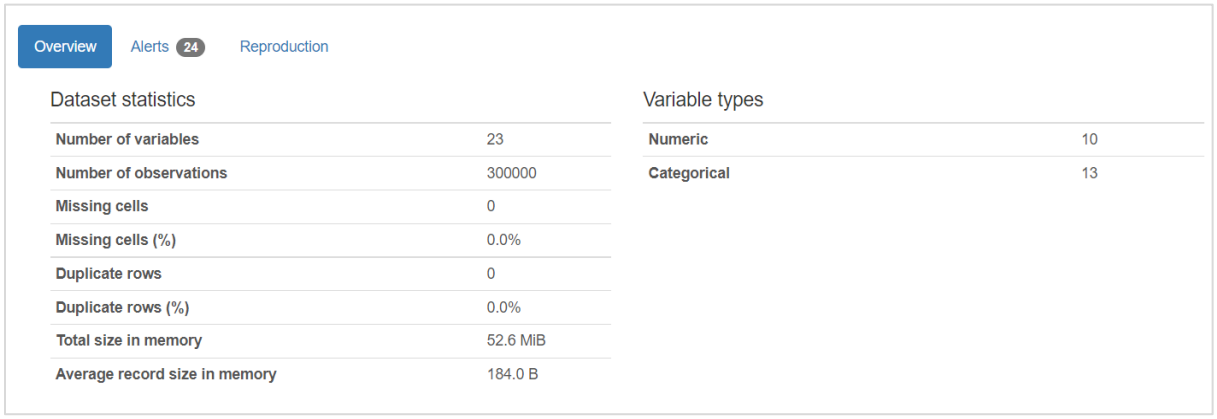
3.8 Data Profiling

In addition to the previously discussed steps, a data profiling step was performed to attain a more in-depth understanding and correlation between the features. This goal was achieved by using the Pandas profiling tool. This tool helped generate an interactive report in a web format that was easy to comprehend. Additionally, it generated a univariate and multivariate report for better

data understanding. The essential information for each feature, such as the quantile statistics, descriptive statistics, number of unique values, missing values, and correlation metrics, was generated and presented by this tool in an HTML report. Figure 33 portrays an overview of the data profiling analysis result for the features present in the dataset. It shows the size of the number of variables divided into numerical and categorical types, record counts, and some other statistics, such as missing values, redundant records, and size.

Figure 33

Data Quality Report Containing the Numerical Features



Visualizing and analyzing the transactional features could help in understanding the statistics better. Data profiling helps understand the feature's statistical distribution and generates an interactive section where the user can select the features for which the interaction needs to be visualized. Figure 34 represents the profiling analysis result of transaction date and time, the amount spent, credit card number, and expense category. It was also essential to gain additional insights from the statistics related to merchants. Figure35 depicts the data profiling result of the merchant and the related location information. Similarly, figure36 shows the customer-related features' profiling result. Figure 37 shows the customer's latitude and longitude's profiling. The essential statistics obtained by data profiling of location-related features are shown in figure 38.

Figure 34

Data Profiling Result – Transactional features

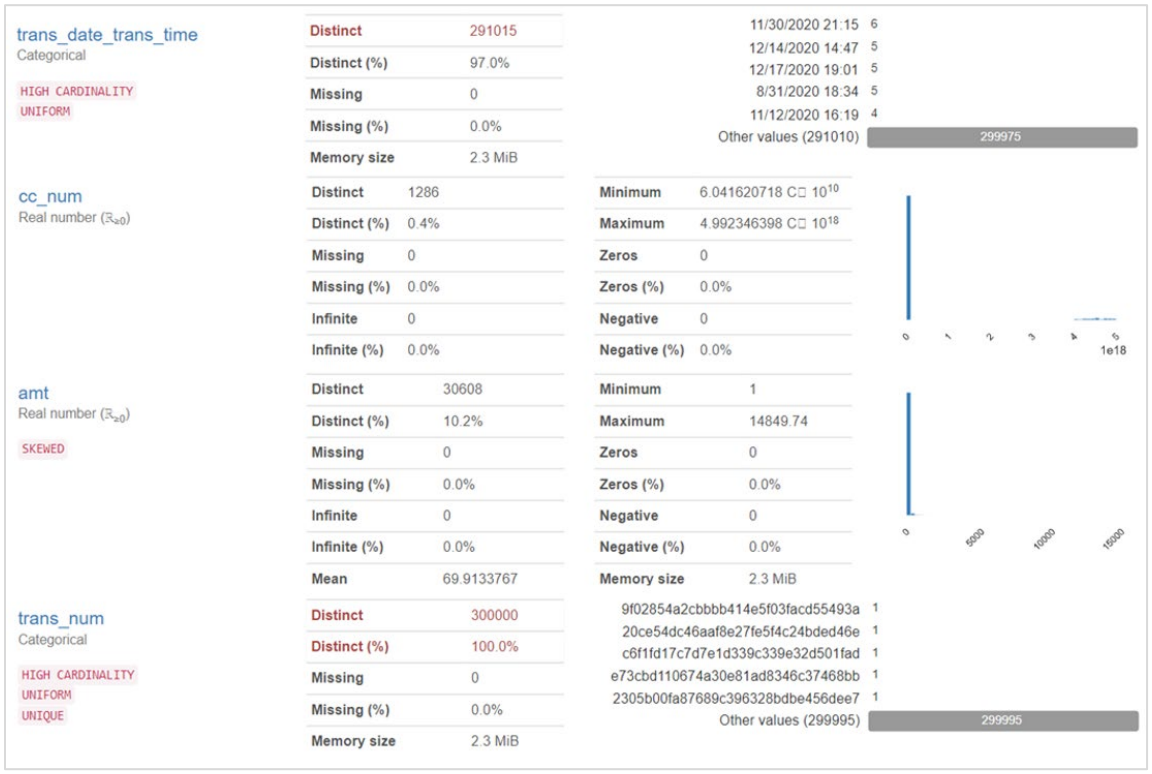


Figure 35

Data Profiling Result – Merchant Features

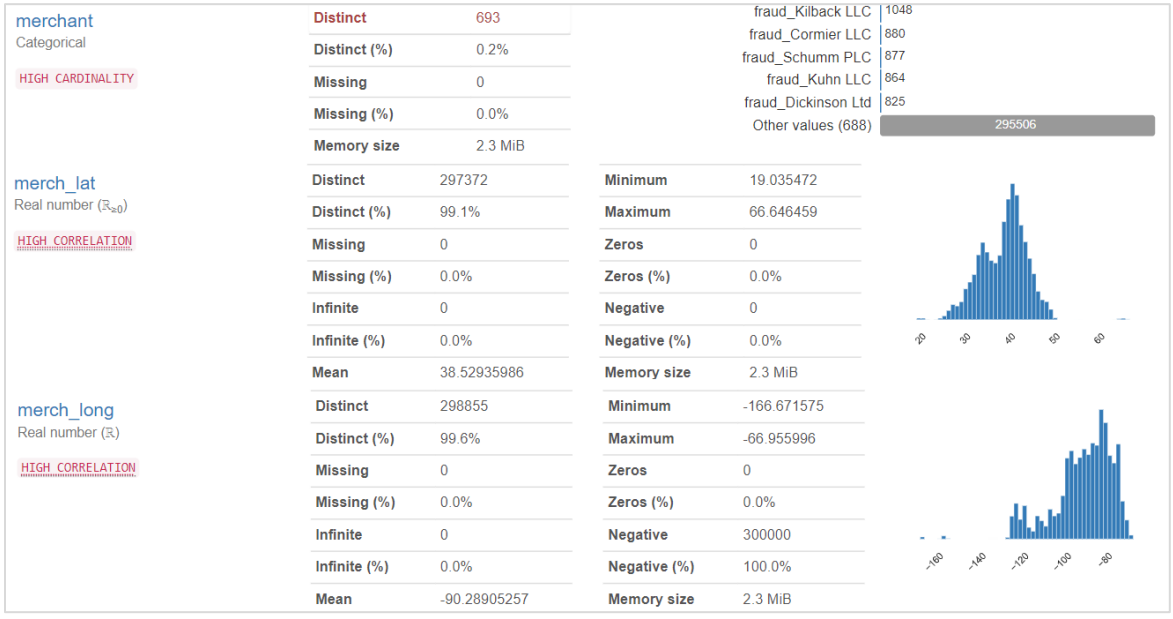


Figure 36

Data Profiling Result – Customer Features

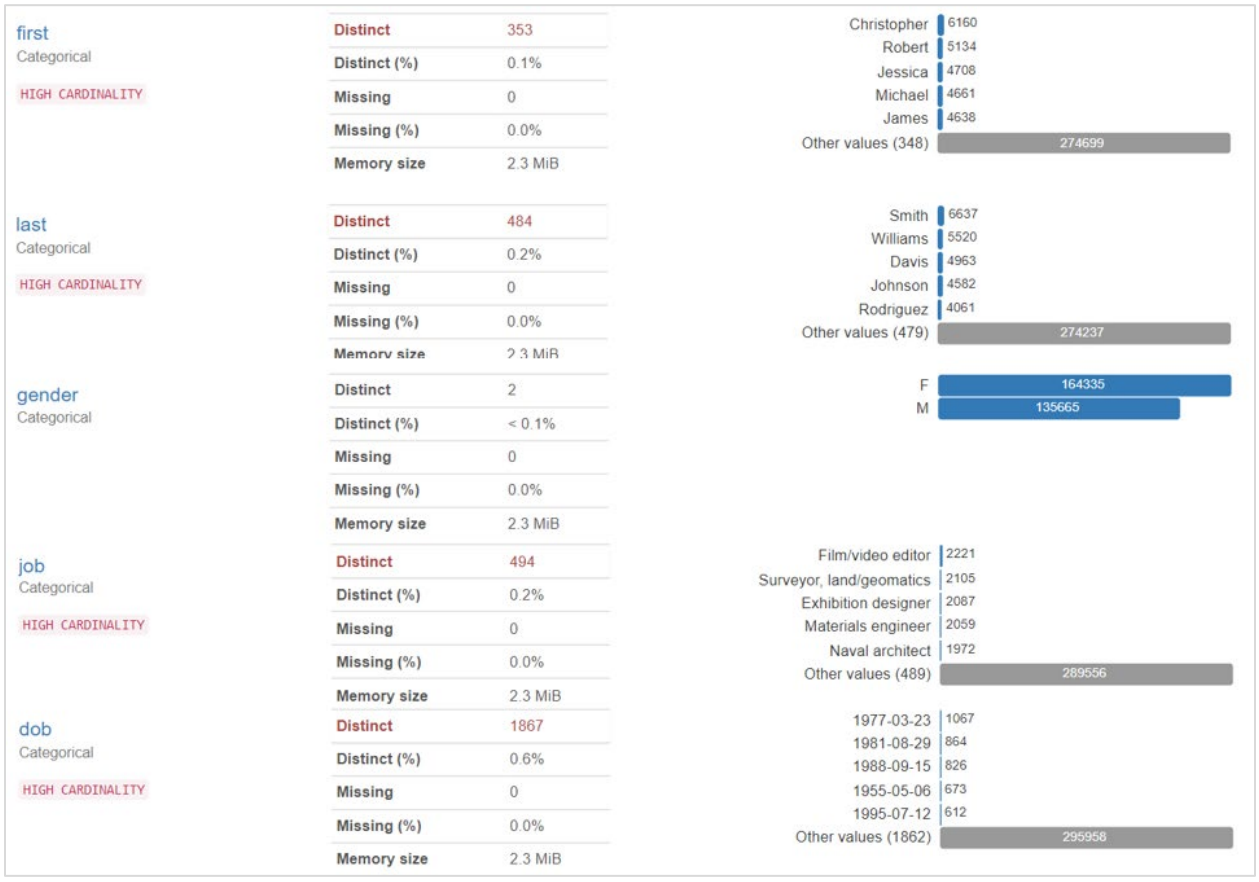


Figure 37

Data Profiling Result – Latitude and Longitude Features of Customer

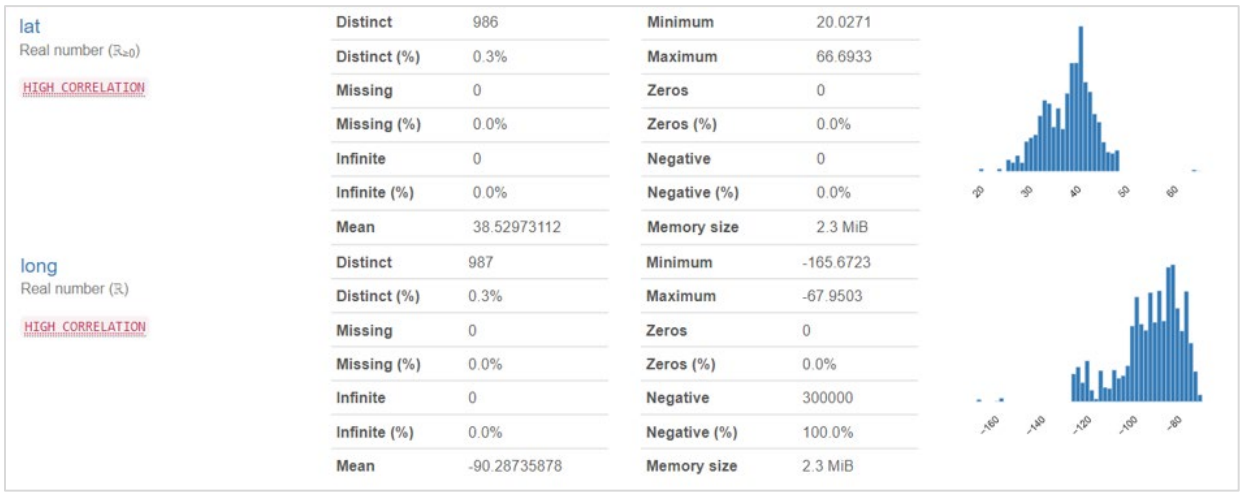


Figure 38

Data Profiling Result – Location Features

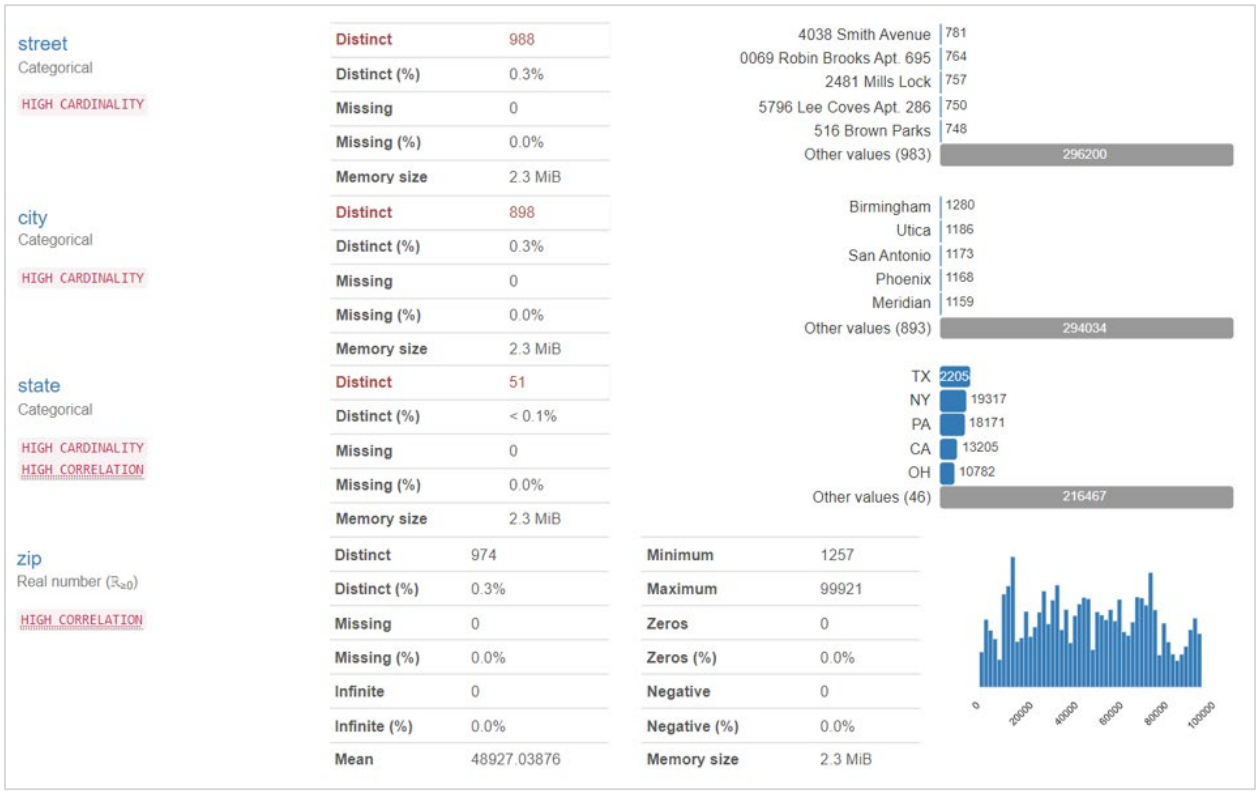
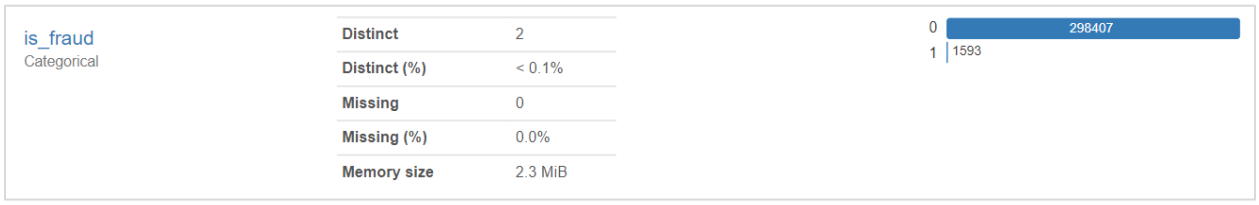


Figure 39 shows the profiling results of the target feature.

Figure 39

Data Profiling Result – Target Feature



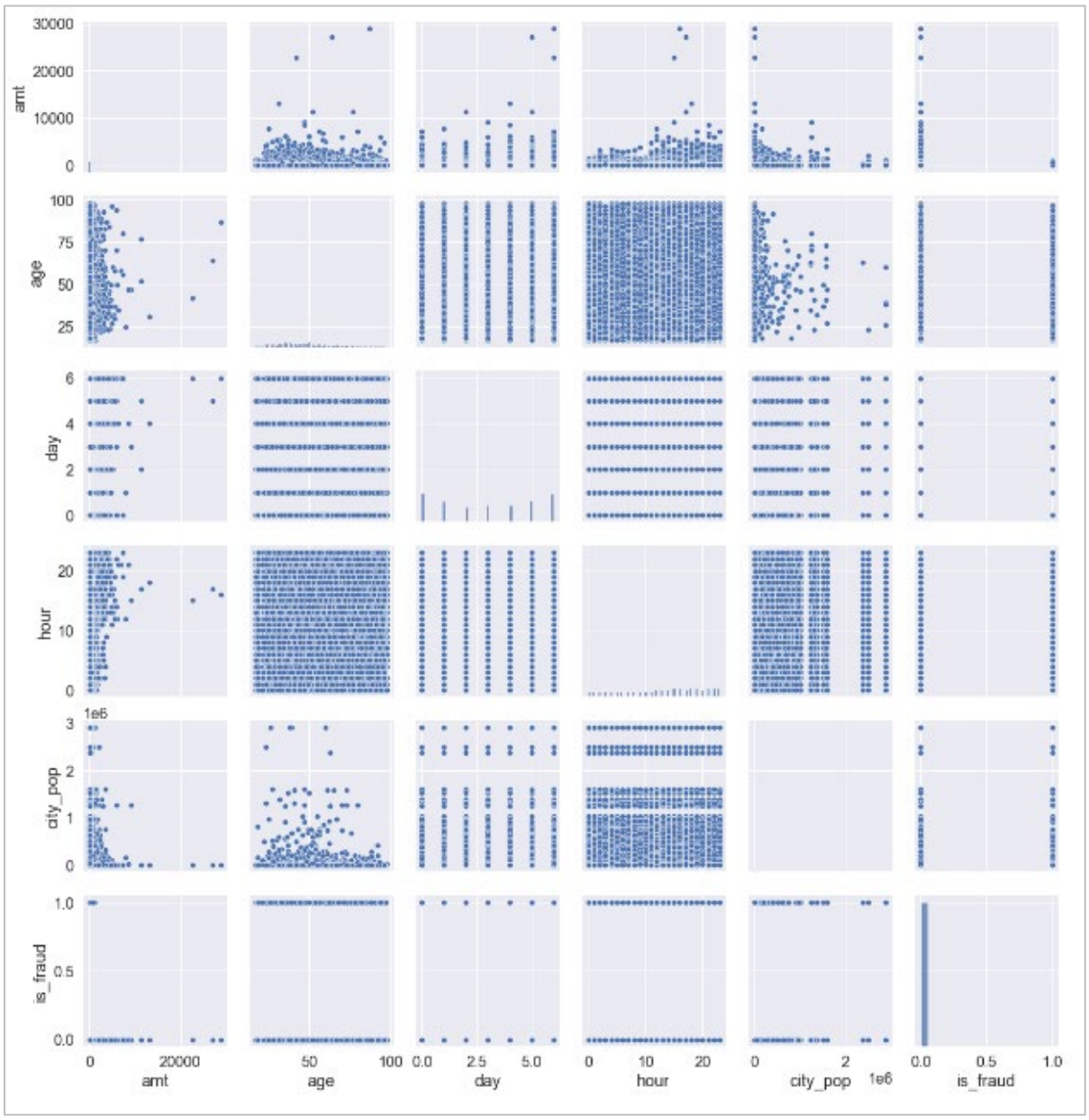
3.9 Data Analytics Results

Data analytics is critical to understand the characteristics and statistics of the features and how they relate to one another. Data analytics can help understand the relationship between the features, which can either have a causation factor linked to it or purely coincidental.

Thus, finding out the correlations between the features could help to a greater extent in the prediction accuracy of the model. There are several ways to identify and understand these correlations, such as pair plots, KDE plots, and heat maps. A data analytics step was performed to examine the features' correlation further. Figure 40 portrays the pair plot of some significant features, and it shows the correlations between age, transaction amount, day and hour of the transaction, city population, and is_fraud feature.

Figure 40

Pair Plot

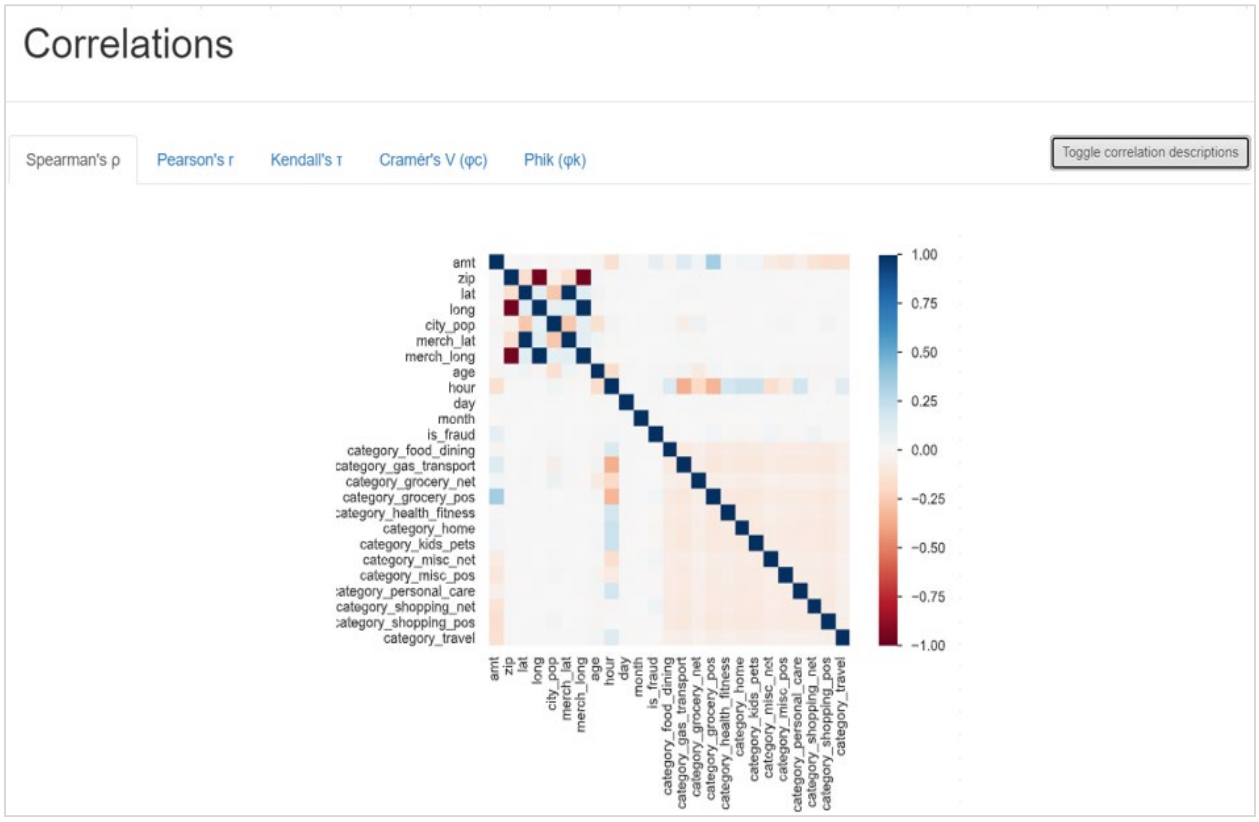


Each colored dot on the plot depicted in figure 40 shows the distribution of the data points for each associated feature and their relationship with the other features shown in the plot. It was observed that the relationship between the additional feature age is uniformly distributed in relation to the transaction hour feature and vice versa. Additionally, this plot showed a few outliers in features like age, transaction day, or city population.

Figure 41 represents the correlation heatmap of the features. It helped in identifying the potential relationships and understanding the strength of each of these relationships. It was observed from the heatmap that the fraud feature has a better correlation with the transaction amount rather than other features. In addition, figure 41 implied that the location information of a merchant and a credit card holder had a better correlation and vice versa. Data profiling helped with the generation of four different kinds of correlation heatmaps. One of those is depicted here.

Figure 41

Correlation Matrix



4. Model Development

4.1 Model Proposal

Modeling is the next phase in the CRISP-DM, following all the tasks mentioned in the previous section to ensure the data is clean. The data coming to this phase is split for training and testing, and SMOTE has been performed to avoid under-sampling of the target features. The model that needs to be chosen for performing prediction was decided based on the background research done in the project's early stages. The understanding gained from the research classification models was best suited for predicting fraudulent transactions. Supervised and Semi-supervised ML models were chosen to be performed to get a better idea of which type of machine learning model will yield a better prediction result.

A supervised Machine Learning model uses a training set with labels passed to the model. The model is provided with the input and target feature for the model to learn from it. This model measures the loss function in this model is used to measure precision, and it is adjusted until the error is suitably minimized.. A supervised learning algorithm generates an inferred function from the training data and can map new instances. The data needs to generalize the model to predict the outcome of unseen situations.

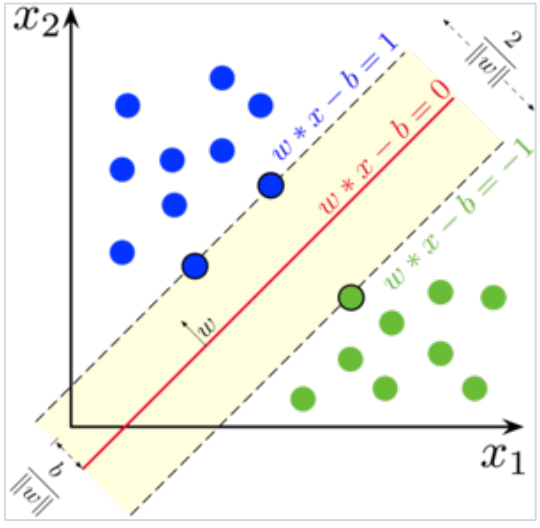
A semi-supervised machine learning model uses a small number of labeled data and many unlabeled data being passed during the training phase of the models. It combines supervised learning and unsupervised learning. The model predictions add dimensions and establish a structure to the learning problem, which clusters the data class. The unlabeled data points expose the model, generalize, and accurately estimate the whole distribution. The model obtained through this machine learning model will give an accurate close distribution. This section helps explain the chosen models and the reasoning behind the choice briefly.

4.1.1 Support Vector Machine (SVM)

The first model that was identified as a potential for execution is the Support Vector Machines (SVM). SVM employs both linear and non-linear enactments and is particularly beneficial when dealing with data that contains extremely complicated datasets. When dealing with margin separations, SVM typically performs well and is less prone to overfitting. Numerous papers already published pertinent to the topic at hand attest that SVM is one of the most popular algorithms for classification problems. SVM uses a hyperplane to assist in generalizing the distinction between the classes. A line that separates and establishes the boundary between the classes is known as a hyperplane. Assuming the data is linearly non-separable, the SVM employs kernel methods to potentially make it separable. One benefit of SVM is that it does not skew the results due to the presence of outliers. Using both linear and non-linear technology will depend on the complexity of the data. SVM often has good accuracy and performs well in complex domains with distinct margins. Figure 42 retrieved from Wikipedia contributors (2022), depicts the fundamental SVM algorithmic structure.

Figure 42

Support Vector Machine

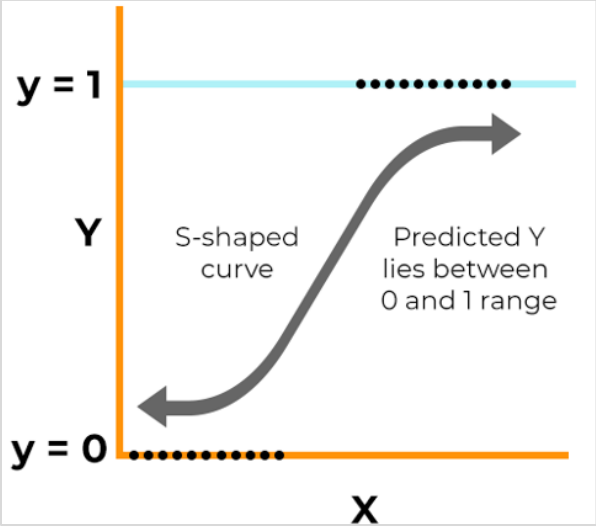


4.1.2 Logistic Regression

The second such chosen model is Logistic Regression. The logistic regression model used here can be classified as binomial due to the nature of the target field. Basic linear models for classification, such as logistic regression models, are used to forecast discrete or binary dependent variables. To reduce error, regression also uses a penalty parameter. Even if the correlation of features is not regular, new data can be supplied, the model can continue to be updated in the ongoing process, and the model functions effectively. The fundamental concept of logistic regression is shown in Figure 43, retrieved from Kanade (2022).

Figure 43

Logistic Regression



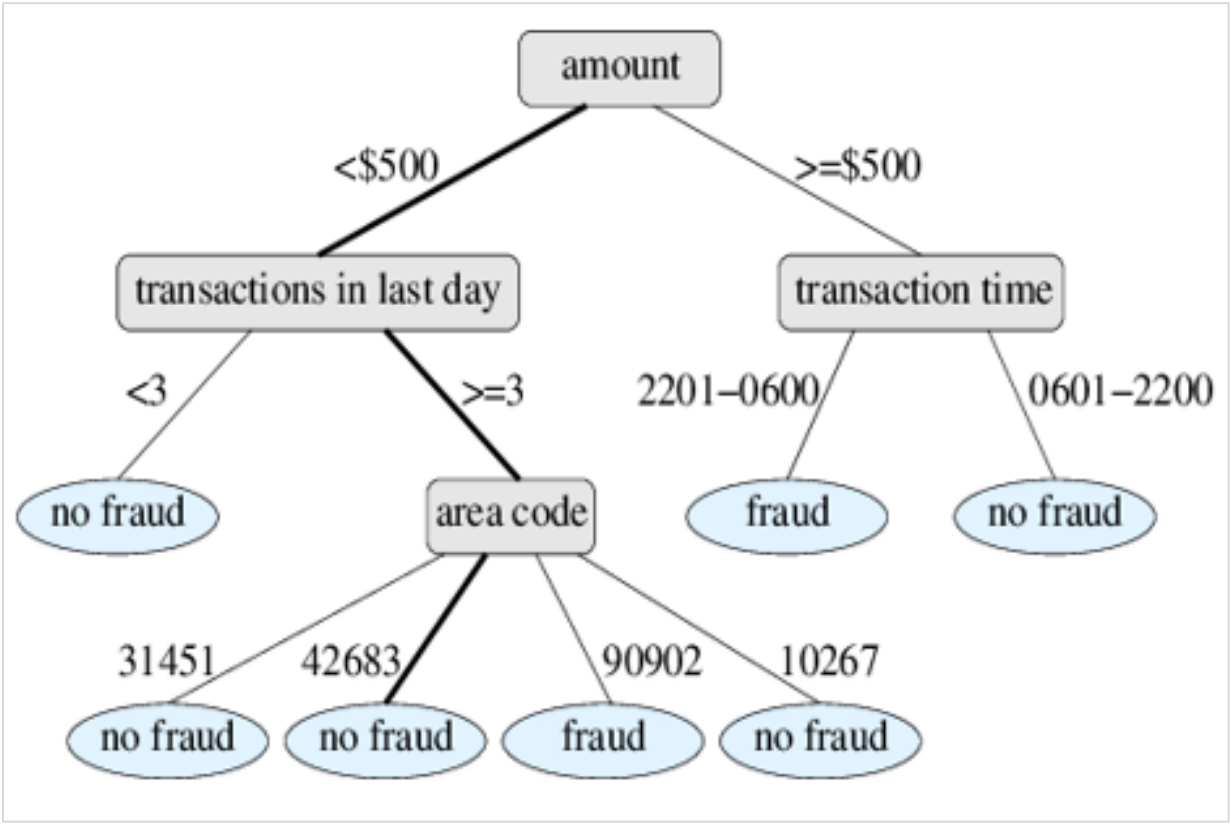
4.1.3 Decision Tree Classifier

A Decision Tree Classifier is the third chosen model, which uses a rule-based methodology to solve classification and regression issues. This kind of classifier divides the dataset into groups of data points belonging to the same class by using the values in each feature. The class discriminator recursively partitions the training data until one class dominates each partition. A reliable turnover prediction model uses both old and new data.

The decision tree is one of the most common and fundamental models for categorization issues. It has different parts, including a root and leaf node, branches, and internal nodes, and has a hierarchical tree structure. It begins with a root node, moves through internal nodes, and then arrives at leaf nodes, which stand in for possible outcomes. Because of their non-linear properties, decision trees are often more adaptable in their exploration. When dealing with numerous possible outcomes from a range, decision trees are beneficial. A decision tree is also picked as one of the models to be investigated and contrasted with the other probable models because the research is a classification problem with various potential outcomes. Figure 44 retrieved from Kalyanakrishnan (2014) shows the fundamental design of a decision tree classifier.

Figure 44

Decision Tree



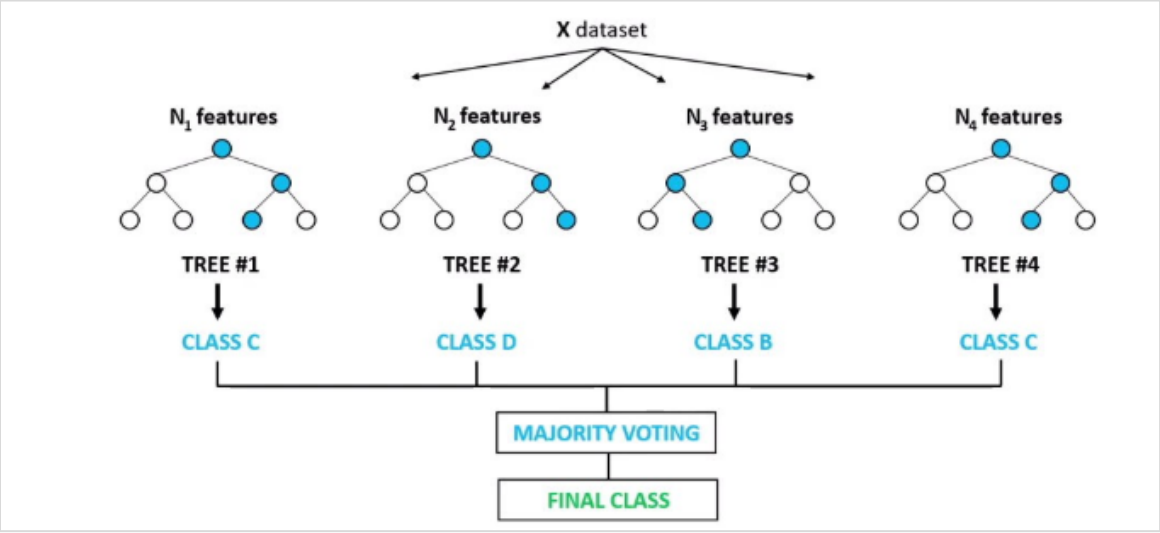
4.1.4 Random Forest

Next, the Random Forest model was identified to be chosen. The random forest model, which consists of a collection of distinct decision trees, can take into account features that are not linear. It works well in high-dimensional environments with lots of training data. Calculating the maximum features per tree, the ability to bootstrap samples, and splitting performance are the best working solutions. This algorithm can carry out the warm start operation.

The method promises unique benefits, like the capacity to produce many classification trees while avoiding noise. The model randomly chooses data and variables from the source set to make a forecast. When working with varied data sets with a high degree of dimension, a random forest classifier comes in helpful. Additionally, random forest algorithms appear to have faster training rates, leading to lower computation costs. While conducting background research on the subject, it was intriguing to discover how frequently studies used the Random Forest Classifier as their choice model. Figure 45, retrieved from Chauhan (2021), depicts an example of how the Random Forest Model persists.

Figure 45

Random Forest



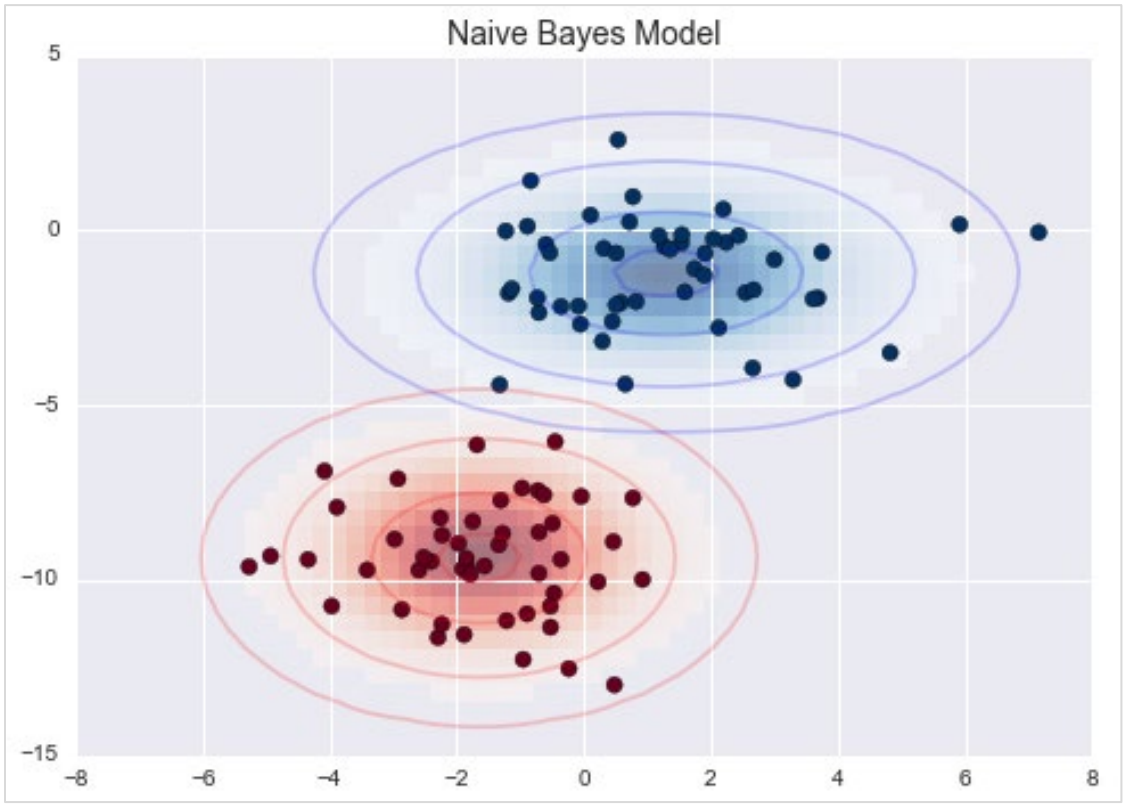
4.1.5 Navies Bayes Classifier

A machine learning model called Navies Bayes (NB) classifier that distinguishes between various objects based on attributes was chosen as the fifth model. Probabilistic machine learning is utilized when performing a classification problem, utilizing the Bayes theorem with an intense independence assumption between the features.

For naive Bayes classifiers, the number of parameters is linearly proportional to the number of variables (features/predictors) in a learning problem, making them highly scalable. Instead of using an expensive iterative approximation, maximum-likelihood training can be used for various other classifier types performed in linear time by evaluating a closed-form expression, which is faster. Figure 46 retrieved VanderPlas, n.d., shows the naïve Bayes model.

Figure 46

Naïve Bayes Model

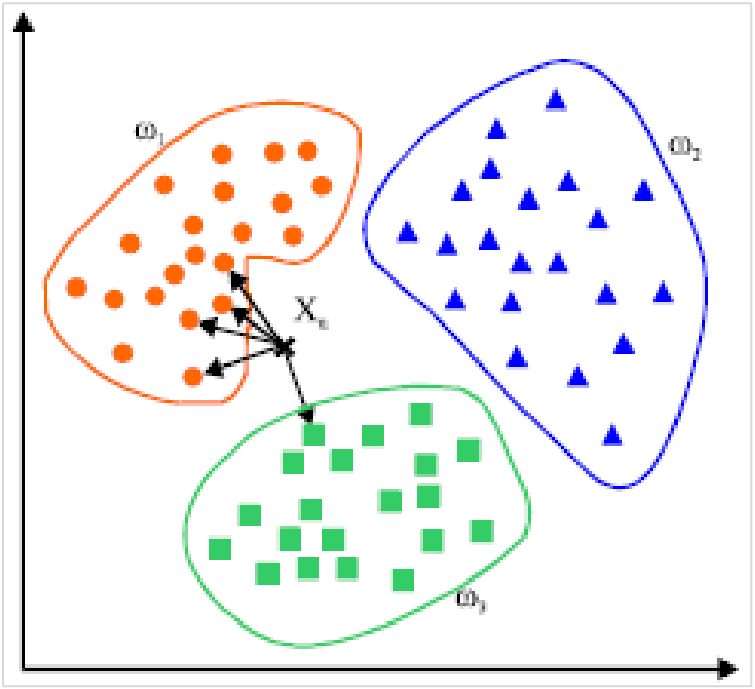


4.1.6 K- Nearest Neighbors

KNN, a supervised non-parameter machine learning model used for regression and classification, was also identified as one of the models that could be implemented. The function is approximated locally, and the computation is deferred in the classification model. With k-NN, all estimation is postponed until after the function has been evaluated and the function is only locally approximated. Since this technique relies on distance for classification, normalizing the training data can significantly increase accuracy if the features reflect several physical units or have distinct sizes. The closest neighbor will contribute more to the average than the further neighbors because the model assigns weights and distances based on the nearest neighbor. The critical aspects of the KNN model are that its easy to interpret the output and the calculation time is high during testing compared to training. Figure 47 below, retrieved from K-Nearest Neighbors (KNN) - PRIMO.ai, n.d., shows a sample of the KNN model.

Figure 47

K-Nearest Neighbor

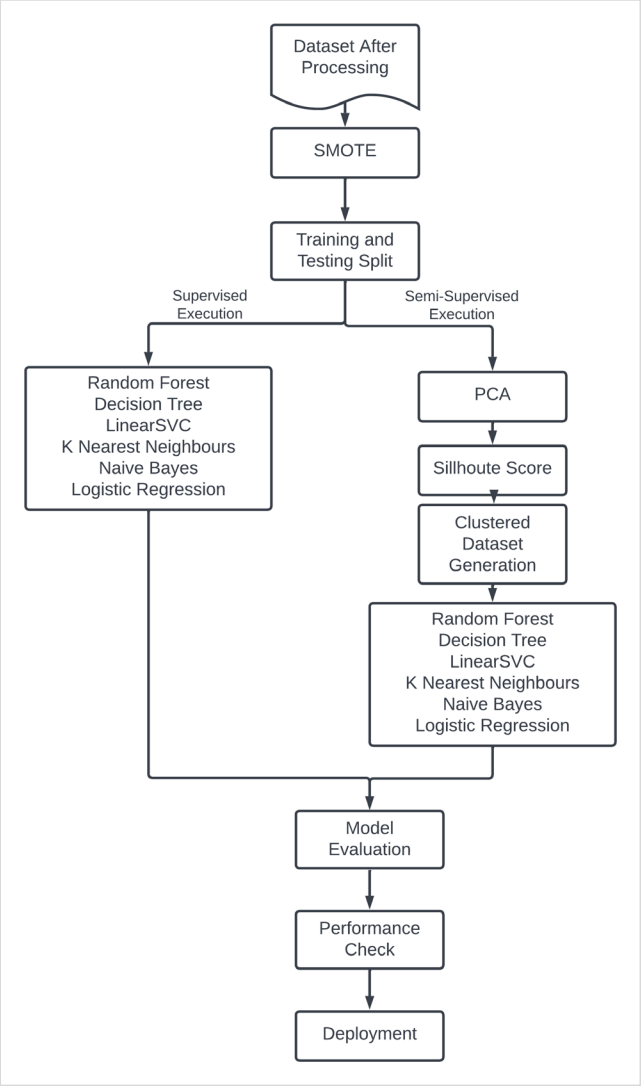


4.2 Modeling Workflow

The chosen models were built during this phase, supporting both supervised and semi-supervised execution. All models were tuned with similar hyperparameter settings while building for the model results to be comparable. PCA and Sillhoute scoring were used while building semi-supervised models to generate the clustered unlabeled feature. The overall Modelling workflow is presented in the figure 48 below.

Figure 48

Modeling Workflow



A detailed description of the various modeling and evaluation techniques involved in the model development phase will be explained briefly in the forthcoming sections. Once the desired chosen models were built with similar parameters, the training was done with 80% training data, followed by testing with 20% testing data. The comparison and evaluation of the modeling results will be explained briefly in the evaluation phase. Cross-validation plays a significant role in the training, testing, and evaluation.

4.3 Model Building and Training

Insights gained from analyzing the various background studies involving the chosen models played a major role in the smooth sailing of the model development phase. Any hiccups that arose during the modeling phase were resolved through further examination of the hyperparameters. For the comparison standard to be the same across all models, parameter settings were chosen considering the uniqueness of the models in mind. Once the architectural design was complete, the building phase took off. Various techniques such as SMOTE, Cross-Validation, Ensemble, Dummy creation, and ROC analysis were considered for both the predictions and performance evaluation phase. Supervised and semi-supervised models were built and trained with similar data so that the results could be comprehensively evaluated. Once the training was done, the testing phase began, and cross-validation was chosen as an alternative to regular modeling.

5. Evaluation

5.1 Model Evaluation Methods

As discussed in the previous section, a handful of models were elected to be the chosen ones. These models were then trained and tested, and accuracy was obtained. In order for the model to be of pristine quality, the evaluation method that was used to obtain the results should be selected after careful consideration. Evaluation of models involves various steps such as bias elimination,

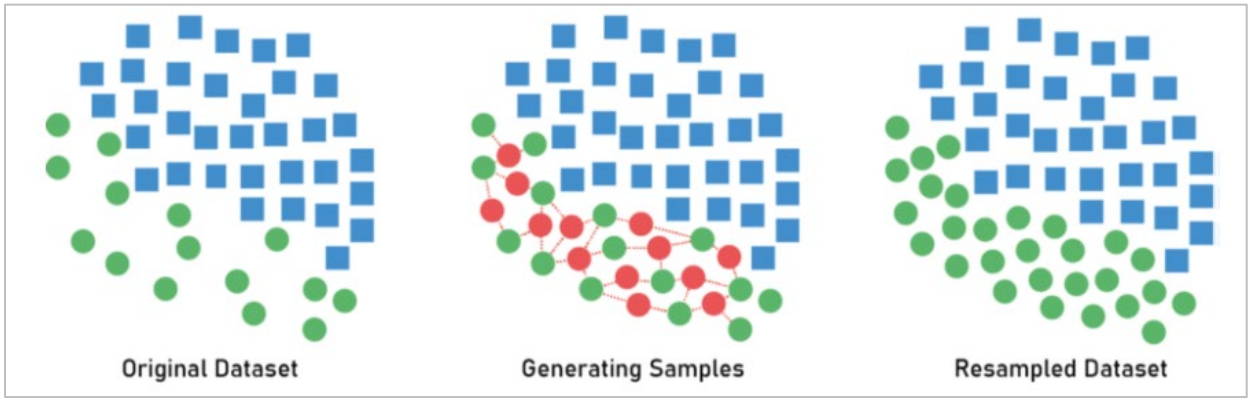
hyperparameter tuning, ample training of the models, and identifying proper depth for the tree-based classifiers. Careful consideration and evaluation of previous studies have helped narrow down the possibilities of how these can be carried out. Hence, this section will discuss the various evaluation methods incorporated while the model building.

5.1.1 SMOTE

The dataset used in this project had more transaction records related to non-fraudulent ones rather than fraudulent ones. This issue was handled using an oversampling technique known as SMOTE. This method created a balanced proportion for the target labels, which resolved the imbalance issue. An imbalanced class could lead to incorrect predictions; thus, resolving this ensured that the model's training was done appropriately. This ensured that the model's accuracy was enhanced than the initial attempts. Figure 49 depicts how SMOTE method works which is retrieved from Natural Language Processing (NLP) Project Example for Beginners (2022).

Figure 49

SMOTE



Once the imbalance in the target label was fixed using SMOTE, finding the optimal training and testing split ratio is mandatory. Ideally, quite a few split parameters can be considered defaults, and one among them is the 80:20 ratio. After careful consideration, a train

test split was performed with a ratio of 80:20. The split ratio was done in such a way that a large proportion of data was assigned to the training phase. Likely, the smaller was assigned to testing. As the dataset is huge, this splitting ratio ensured that the learning phase of the model could be intensive. Intensive training often leads to better predictions. The 20 percent assigned to the test set was also ample, as the overall sample size is humongous. The train and test split performed were done in such a way that the split happens only once. This method is ideal for some scenarios when the dataset is not large enough, and restricting the splitting to one time only works well. On the other hand, when the dataset is vast, splitting the set only once and using the training and testing data for training the model and validating using the test set can not help in getting better prediction accuracy. Therefore, an effective approach should be implemented to ensure that the model is trained well and the testing is done efficiently. The technique used in this project is cross-validation. This will be explained in detail in the next section.

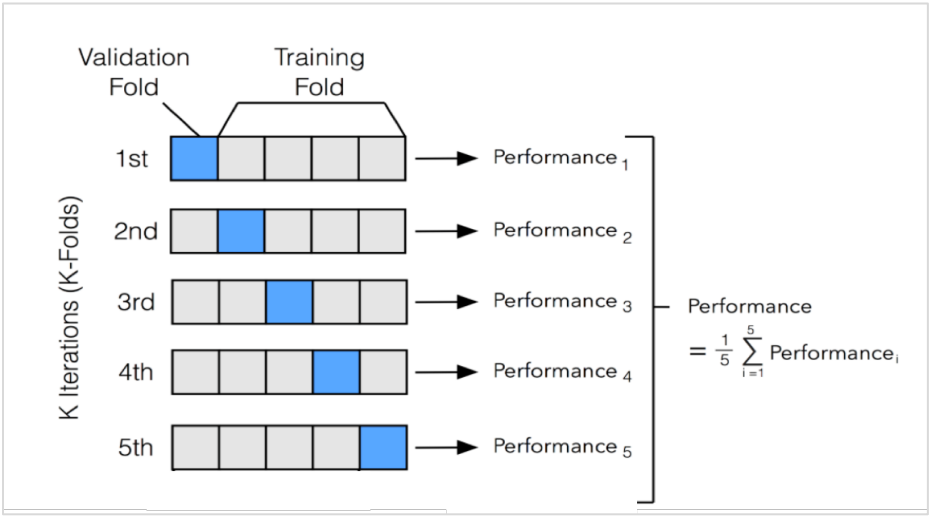
5.1.2 Cross-Validation

Ideally, in a typical scenario of model training, the split happens only once because the training is done with only one set, not multiple. Splitting the dataset once into train and test might seem like a perfect scenario but consider the implications that could arise when the volume of the dataset is not exceptionally high—in such a scenario, splitting the dataset once restricts the learning capacity of the model. Similarly, inconsiderate of the data volume, it is common knowledge that the iterative learning technique yields better predictions than one-time learning. Hence, instead of splitting the set only once and using the training and testing data for training the model and validating, an alternative approach was proposed. The proposed alternative approach should be implemented to ensure that the model is trained well, and the testing is done efficiently. The technique used in this project is cross-validation. As per this

method, the dataset was split multiple times, and each split could be denoted as a fold. The optimal number of folds in this process depended on the available data volume and the prediction accuracy goal. In this project, five was considered an optimal number of folds. Based on this factor, the dataset was divided randomly into five parts during the implementation of cross-validation. Out of these five parts, four were used for training and the fifth for testing. This process was repeated five times, and in each iteration, a different split was performed, resulting in extensive learning, and a distinct fifth test was used every time. The training of the model was expected to be improved in each iteration and lead to better performance and prediction accuracy. In addition, a performance metric is needed to evaluate the model's prediction effectively. Several metrics are available to perform this, and the method opted for while execution was using accuracy. The metric was selected based on the goal of the project's requirements. In this project, the goal was to get a better prediction result in fraud detection. Hence, an accuracy metric is used while doing the cross-validation process. Figure 50 depicts how cross-validation works,

Figure 50

Cross – Validation Framework



5.1.3 Ensemble

The performance of the proposed model may be better in the current context, but the accuracy can be abnormally high. The abnormally high accuracy could be attributed to the overfitting problem, which is quite common with the tree classifiers. Random Forest(RF) and Decision Tree(DT) models are quite prone to overfitting. Hence, finding an alternate hypothesis that could be used when needed is necessary. Here comes the ensemble technique, which can help mitigate the overfitting issue. In the ensemble technique, rather than relying on only a single tree classifier for decision-making, it takes into account a sample of tree classifiers and makes a prediction based on the aggregated result from this sample set of classifiers.

There are several methods to implement the ensemble technique, and the one used in this project is the bagging or bootstrap aggregating method. Several bootstrapped samples were pulled from the dataset, and a DT or RF was formed on each of those. A voting mechanism was used to find the most effective prediction, which calculates the end prediction result based on the outcomes generated by each classifier model. In this way, the overfitting of the models was handled with a proper approach.

5.1.4 Cluster Analysis

In the semi-supervised model, PCA was performed to reduce the dimensionality of data and find the most significant features with an optimal dimension. In this project, the PCA reduction task was performed, and it presented the data in an optimal dimension of three. In addition, a cluster analysis was performed to group the identified data in these three dimensions into different clusters. This goal was achieved using a Silhouette score analysis, and based on the score, four were identified as the optimal number of clusters. Figure 51 and 52 represent the cluster analysis results in two and three dimensions, respectively.

Figure 51

Cluster Analysis – Two Dimensional

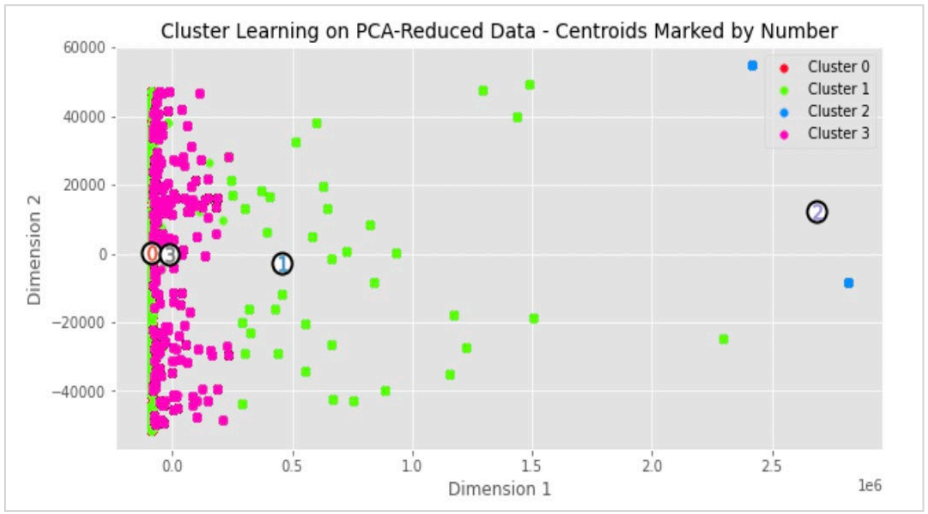
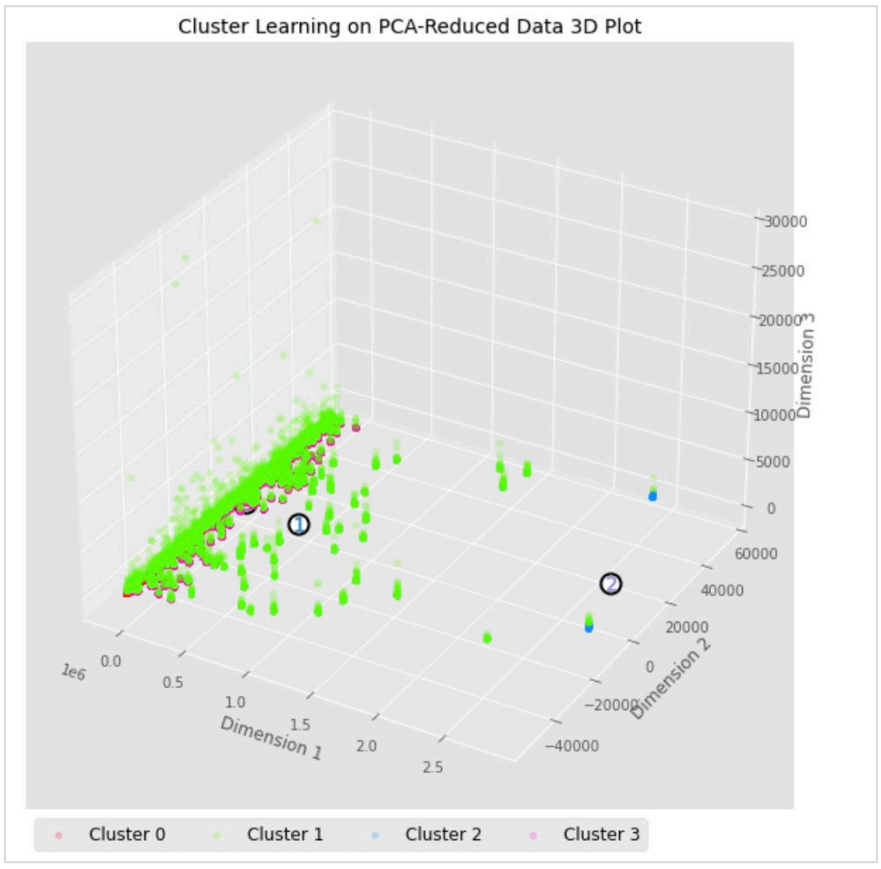


Figure 52

Cluster Analysis – Three Dimensional



5.2 Model Comparison

The model comparison will shed light on various aspects of how well machine learning models perform under different parameters—contrasting the models using ensemble methods, the hyperparameter of the selected machine learning model, and training duration and accuracy. This section compares the machine learning model RF, LR, DT, LinerSVC, NB, and KNN. The following part will examine each model's training time in supervised and unsupervised ML models.

5.2.1 Comparison based on Performance Time

Training time represents the time taken for a model to train on the dataset. When considering the supervised machine learning, Gaussian NB took the least training time, i.e., 0.1633 seconds; this could be because the model calculates the probability of each class, and there are no coefficients that need to be fitted to optimize this process. The logistic regression model had the second-lowest training time, clocking in at 1.1315 seconds; this could be attributed to the model's simple probabilistic interpretation and less complicated algorithm. The complexity of the model during the traversal down to construct the tree with the dataset may be why the decision tree model required 4.8187 seconds to train. KNN model takes a time of 14.8942 seconds to train could be the result of the data size passed; otherwise, this model takes no time to train as all the processes in carried out during the testing. SVM's training time of 51.4051 seconds could be because too many instances are passed to the training kernel function. The training time by Random Forest of 63.7831 seconds was the highest; this could be the result of its complexity which uses randomness in selecting features to build individual trees, which later will create and forest that will be used to predict the outcome. The time taken by each model was recorded during execution, and these records helped a great deal in assisting with the comparison. Figure 53 depicts the time taken by

each model and the cross-validation accuracy which will explained in forthcoming sections.

Figure 53

Supervised model Performance Time

	classifier	time
0	LinearSVC	51.140518
1	LogisticRegression	1.131529
2	DecisionTreeClassifier	4.818788
3	RandomForestClassifier	63.783102
4	GaussianNB	0.163382
5	KNeighborsClassifier	14.894248

The implementation of unsupervised models also resulted in a similar training time as supervised expect a few contrasts. Firstly, SVM took a train time of 114.1286 seconds; this increase in time compared to supervised is due to the addition of a large number of unlabeled data and the size of the train set. Secondly, The training time of the KNN model increased drastically to 166.1287 seconds; this was due to the model getting a large amount of unlabeled data and adding features to the training data. Figure 54 depicts the outcome recorded during the semi-supervised execution.

Figure 54

Semi-Supervised model Performance Time

	classifier	time
0	LinearSVC	114.128689
1	LogisticRegression	3.627432
2	DecisionTreeClassifier	9.665615
3	RandomForestClassifier	107.466418
4	GaussianNB	0.489584
5	KNeighborsClassifier	166.128751

5.2.2 Comparison of Results After Ensemble

The resultant accuracy attained during the Random Forest, and Decision Tree's Cross-validation is overfitting. Hence, to tackle this issue, ensemble techniques were used. The comparison of results attained before and after using ensembles will be discussed in this section. Predictive models called ensembles combine the results of two or more models. The reason for using an ensemble was to create a more accurate classification model, which will help reduce false predictions made by the model. A bagging classifier was implemented on Random Forest and Decision Tree model as the training accuracy of the model was above 98%; this could mean that the model has been overfitted. This ensemble technique employs a meta-estimator that fits base classifiers one at a time to random subsets of the original dataset. Then it aggregates the individual predictions (either by voting or by averaging) to provide a final prediction. A hyperparameter was set to the base model's maximum depth to 10, which will restrict the depth of the tree built from the training data.

The bagging classifier had different hyperparameters, such as a base estimator, bootstrap, number of jobs, and random state were set to help generalize the tree models. The Ensemble technique helped reduce the training accuracy of the model by six percent, which was then passed to the test data, where the prediction could be made on new instances. This was implemented in both supervised and semi-supervised models. In order to obtain a simplified dataset from the original data, a considerable amount of the labeled data can be converted to unlabeled data that can be passed through the semi-supervised model. As a part of this process, the original data is first passed through the principal component analysis to identify trends, jumps, clusters, and outliers, representing a multivariate data table as a smaller number of variables. While maintaining patterns and trends, the relationship between the features will be revealed, reducing the complexity

of the data. The performance of the models before and after enabling bagging is depicted in figures 55 and 56.

Figure 55

Performance of Models Before Ensemble

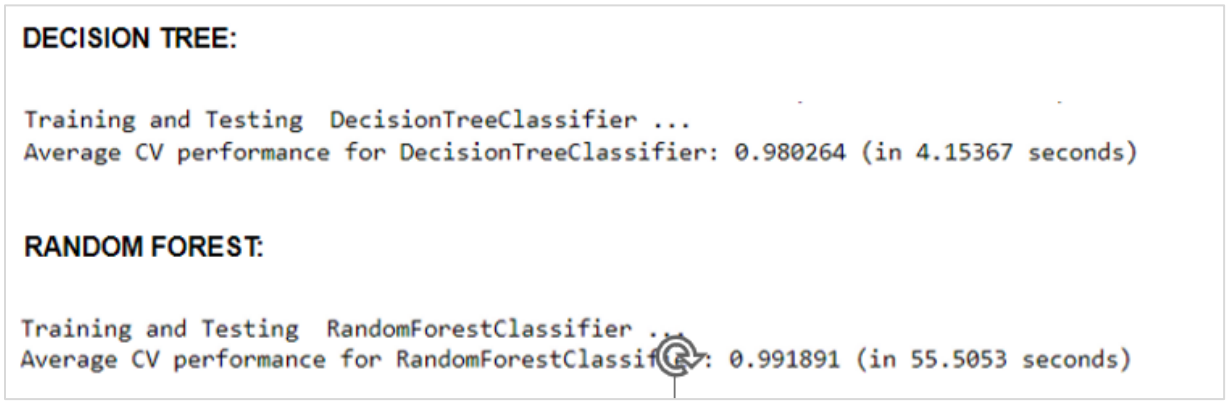
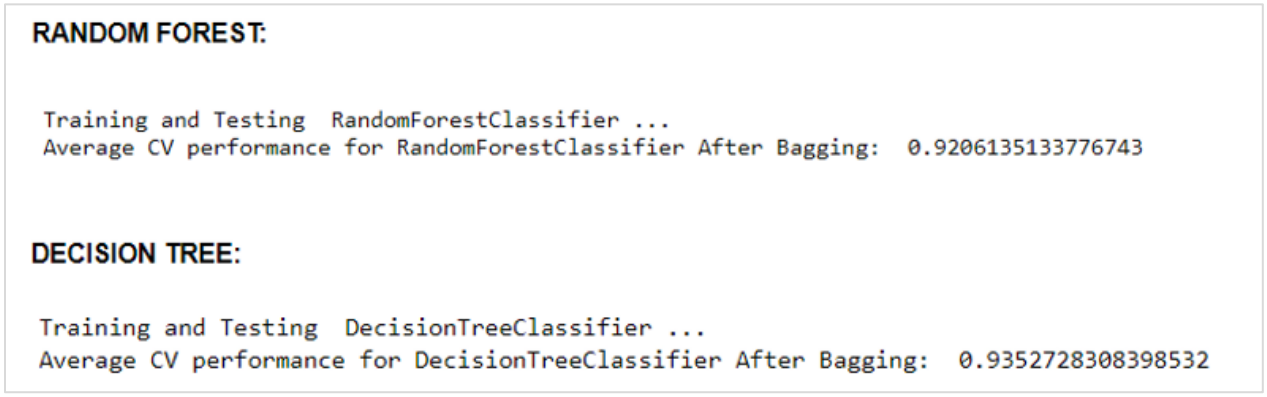


Figure 56

Performance of Models After Ensemble



5.3 Model Validation and Evaluation Methods

Once the models are developed, and all the necessary steps are performed for evaluation, it is time to validate the result achieved by each model. This section demonstrates the result achieved in the model using some performance metrics. A performance metric aids in identifying the performance of the model and how good or bad the result is. This metric signifies if the model is calibrated enough to identify and classify fraudulent transactions. There are several

ways to implement these performance metrics, such as accuracy, F1 score, precision, and Area Under the ROC(Receiver Operating Characteristics) curve(AUC). The metrics used in this project were accuracy and AUC, which are explained in detail below.

A simple way to estimate the prediction performance of a model is to get the total count of correct predictions yielded by a model. The correct prediction proportion can be calculated using the accuracy metric. Figure 57 shows the formula for calculating the accuracy. The numerator depicts the count of True Positive(TP) and True Negative(TN). Figure 58 depicts the accuracy result of the supervised models whereas figure 59 shows the accuracy result achieved by semi-supervised models. It was observed that the accuracy score for the random forest classifier is higher than the other ones in both the supervised and semi-supervised approaches. This shows that random forest had better performance and thus the classification would be accurate.

Figure 57

Accuracy formula

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + FP + TN + FN)}$$

Figure 58

Accuracy -Supervised Model

classifier	accuracy
LinearSVC	0.709989
LogisticRegression	0.870780
DecisionTreeClassifier	0.980935
RandomForestClassifier	0.991766
GaussianNB	0.842215
KNeighborsClassifier	0.967967

Figure 59

Accuracy -Semi-supervised Model

	classifier	accuracy
0	LinearSVC	0.745764
1	LogisticRegression	0.870655
2	DecisionTreeClassifier	0.983246
3	RandomForestClassifier	0.992813
4	GaussianNB	0.842408
5	KNeighborsClassifier	0.967783

5.4 Model Results Discussion

The previous section covered the various customizations to the parameters and the evaluation techniques that aided with the model's exceptional performance. A few factors remained constant during the multiple iterations of evaluation involved. Some of those constant factors were tested, and the observed results will be discussed in this section. Based on the analysis from the previous section top 3 models trained in supervised and semi-supervised were chosen: Random Forest, Decision Tree, and K-Nearest Neighbor are concluded based on the scores in the previous section for both supervised and semi-supervised models. The models were implemented using a cross-validation technique having five folds. Average performance was recorded, ensemble techniques were used in the tree model, and the results obtained were more accurate once the models were generalized. Some of the issues tackled during the execution of the model were the exhaustive running time of specific models, overfitting, and the prevalence of biases in data. Each issue and the fix were documented thoroughly to facilitate ease of access in future scenarios. Tree classifiers were more prone to overfitting and were tackled using bagging, which is an ensemble technique. Detailed evaluation of the true positive, false positive, true negative, and false negatives will be discussed in detail in the next section.

6. System Evaluation and Visualization

6.1 Visualization of Results

The results attained by the various models were compared using visualizations for easy readability. Figure <> below depicts the accuracy attained by supervised and semi-supervised models, along with a line chart representation of the execution time of both. The common characteristic of both is that random forest performs the best in both scenarios. However, this visualization represents the results before the ensemble, which means reducing the overfitting. When the ensemble is applied, though random forest still maintains its higher accuracy position, it drops to 93 from 99 percent; similarly, in the decision tree, the accuracy drops from 98 to 94 percent. Both results are depicted in figure 61.

Figure 60

Summary of Results

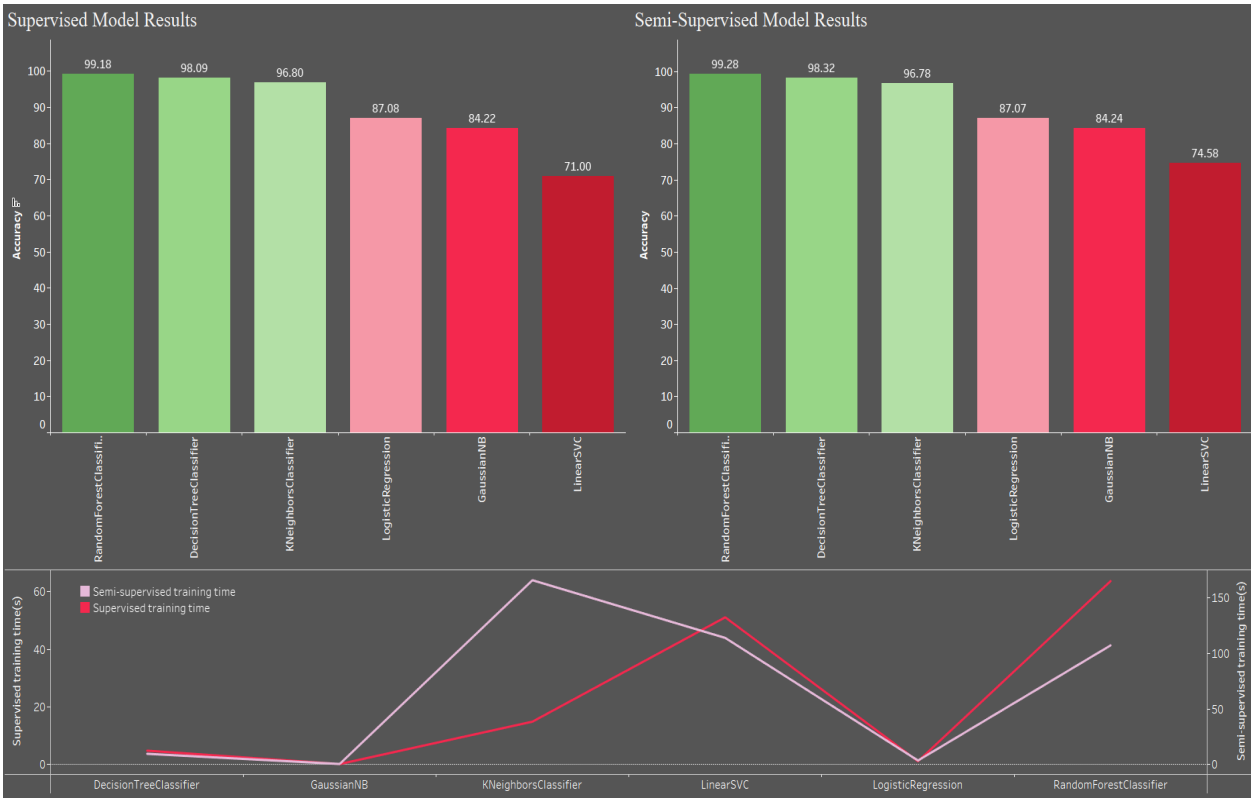
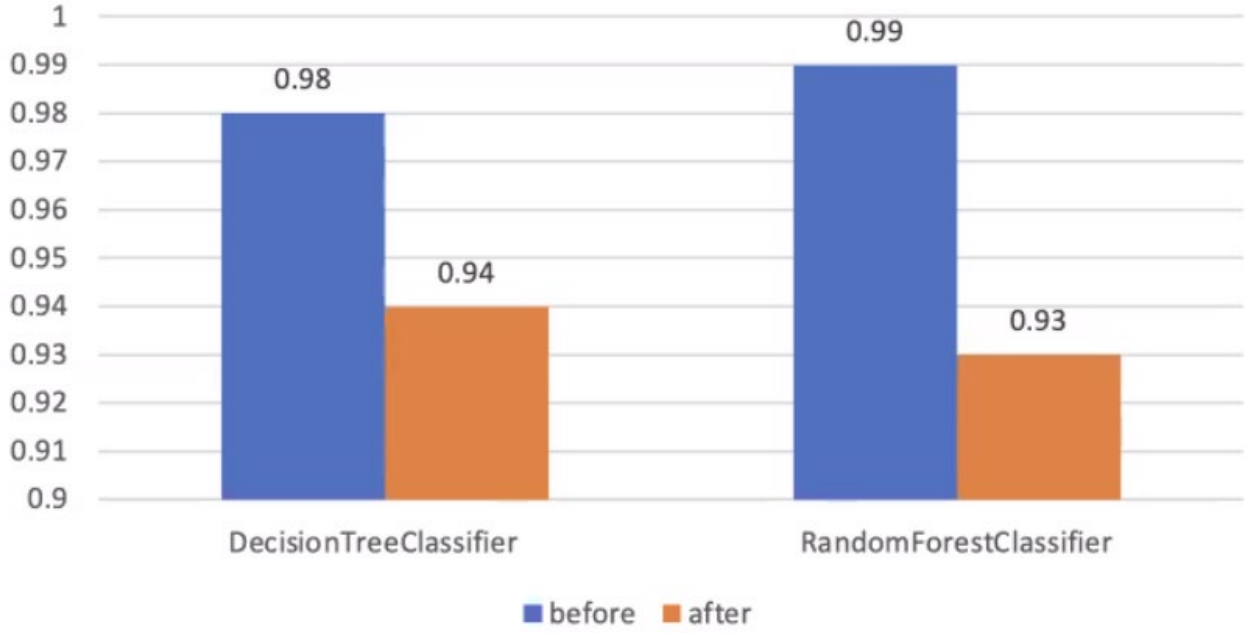


Figure 61

Results before and After Bagging



6.2 ROC Curve Analysis

It is essential to validate the performance of a model by using multiple metrics as it ensures that the model is robust enough in terms of performance and prediction accuracy. Hence, another approach that was used is the AUC metric. The AUC is calculated from the area under the ROC curve. ROC provides a way to visualize the qualification quality and it allows the assessment of the performance of binary classifiers which is the case here. It shows the dependency between the True Positive Rate(TPR) and False Positive Rate(FPR) where TPR is nothing but the proportion of positive actual labels and FPR denotes the proportion of negative actual labels. The ROC curve plots the TPR or sensitivity against an FPR or 1-specificity. The formula for both the TPR and FPR is shown in figure 62. The Area under this ROC curve measures the degree of separation between the positive and negative classes, and it provides a summary of this curve. If the AUC metric is high the performance of the model is considered

better as it can distinguish the positive and negative classes better. The ROC curve analysis for supervised models is depicted in Figure 63. And figure 64 represents the same metric result for semi-supervised models. It was observed that in both the case of supervised and semi-supervised approach, for the random forest classifier the ROC curve went higher than the other models and thus led to an AUC metric value of 98.6%.

Figure 62

TPR and FPR formula

$$\text{TPR (sensitivity)} = \frac{TP}{TP + FN}$$

$$\text{FPR (1-specificity)} = \frac{FP}{TN + FP}$$

Figure 63

ROC Curve analysis – Supervised Model

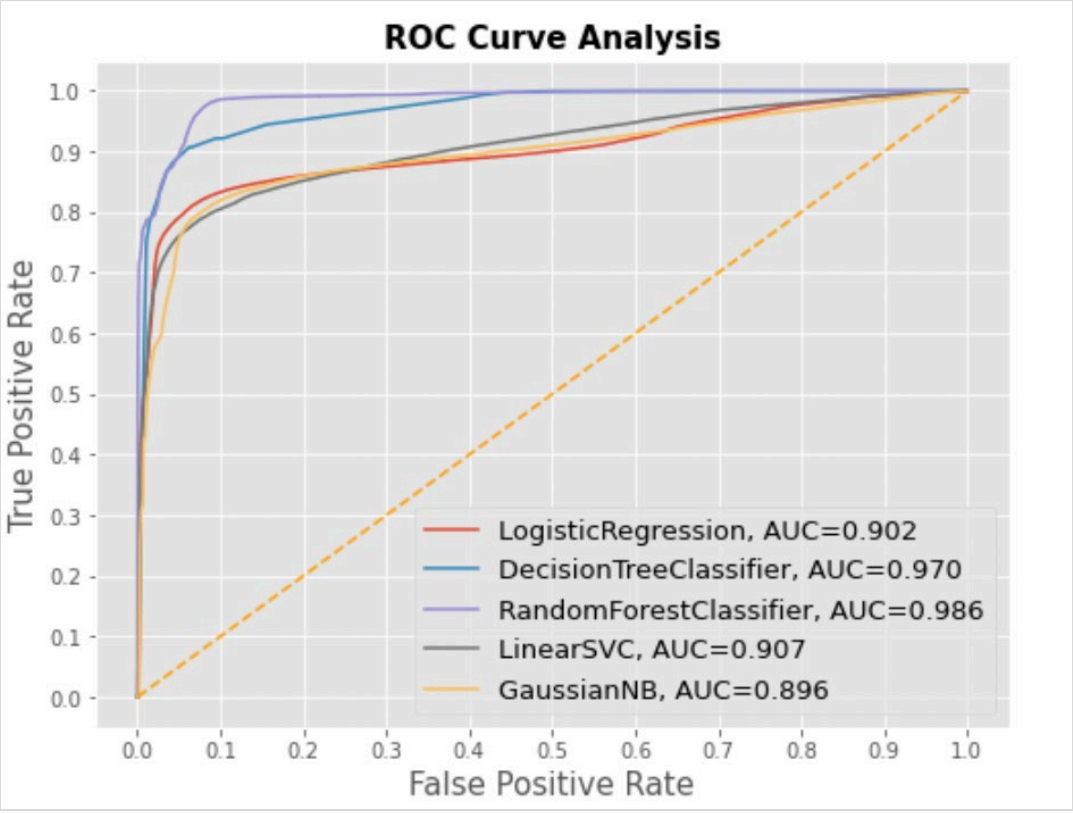
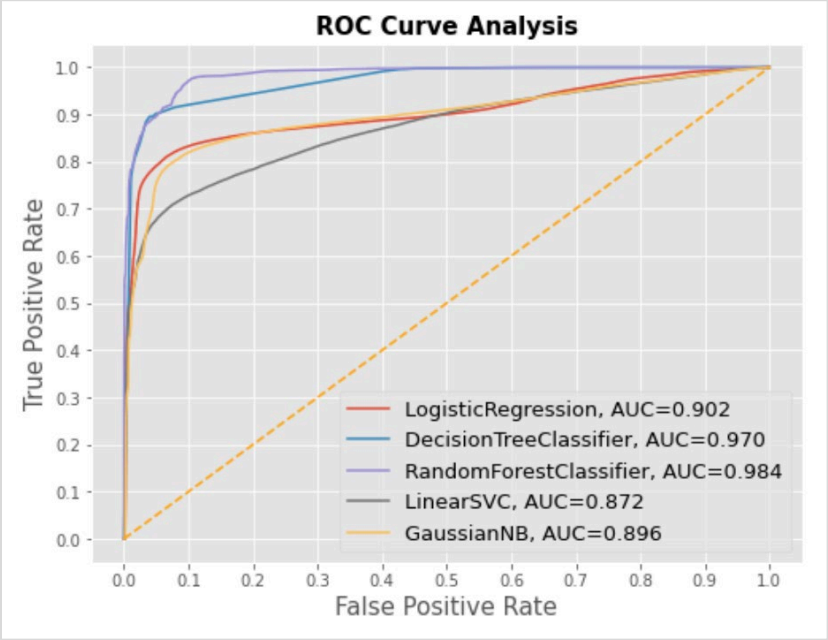


Figure 64

ROC Curve analysis – Semi-supervised Model



7. Evaluation and Reflection

This section covers the knowledge and shortcomings discovered during the tenure of this project. The main objective of this project was to detect fraudulent credit card transactions using machine learning models. Six algorithms were implemented using supervised and semi-supervised models, Random Forest, Support Vector Machine, Decision Tree, Navies Bayes, Logistic Regression, and K-Nearest Neighbor. In order to increase the performance of the models, each model was assessed using a cross-validation technique. The performance of the models was evaluated using a ROC curve and confusion matrix. The detailed insights attained and how they can be utilized in future scenarios will be discussed in this section.

7.1 Benefits and Shortcomings

This section will cover the project's advantages and drawbacks that were observed after evaluating the performance of the machine-learning model. The project's model can be used in

the real world; the dataset used to build it was simulated to correspond to real-life circumstances; as a result, this system can be used to spot fraudulent transactions. This will help reduce the risk associated with such transactions and counter cyberattacks from fraudsters. The compatibility of the models in accordance with real-world scenarios was closely paid attention to. When training the model with real-world scenarios, it has been designed to accommodate the intricacies associated with it and perform well with a large number of data while still being able to uncover patterns that would exist and detect illegal transactions immediately. Another benefit of using a hybrid model is that unlabeled or semi-structured data can be passed as input in real time, and the model can catch mules and track account takeovers easily.

The drawback of this model is that it uses personal information to understand customer patterns which give the model a dimension to predict illegitimate transactions. When applying this to the real world, the cost can be high due to masking or hashing personal information, as it can be misused. Another shortcoming is that the amount of data coming in can be significant; hence setting up a system to handle such a high influx of data could also be a high expense. The system needs to be set to handle outliers that are not a part of the constraints so that the model can continue to learn from itself and perform well.

7.2 Experience and Lessons Learned

This section covers the different ways in which the project was explored in order to achieve the goal. The area and techniques were implemented to help the machine-learning model identify the patterns and trends to perform well.

- **Imbalanced Data:** SMOTE was used to create more data points for the minority class. This method chooses records randomly in a given feature space, which will help balance the dataset.

- **Conversion of Data:** The dataset consists of transaction time and date. The data was converted to a single format and used to generate additional features such as hours, months, and days which could help uncover trends.
- **Model Evaluation Score:** The evaluation techniques include the F1 score, accuracy, precision, and ROC curve. The value of precision and F1 score were monitored to ensure the number of true negatives is not too high, as the purpose of the prediction model will be lost.
- **Choosing Cross-Validation Folds:** The value of the fold will help determine the resampling technique that can be used to evaluate the model. The value of k will divide the data into k number of groups. The large number of k can misrepresent the data; hence choosing the correct value of k is essential.
- **PCA:** This was used to reduce the data to a small dimension which will help interpret the data keeping the patterns and trends the same. Reducing the dataset to the correct dimension can improve the model's performance at a meager cost and reduce the noise in data. Thus, choosing the correct dimension is essential to retain the patterns.
- **Silhouette Score:** This will help uncover any significant gaps between the clusters. If the score is high, we can say the label assigned to the data point is correct. This will help us asses how many clusters need to be assigned to group the data points properly.

7.3 Recommendations for Future Work

The project carried out here to detect credit card fraud covered the majority of the crucial points in terms of design, processing, and machine-learning solutions. Undeniably, the model has long-term benefits in detecting fraudulent transactions and mitigating their risks in the future. Nonetheless, no single work can cover all aspects of a specific topic due to time or resource

constraints. As a result, it draws attention to the existing work, encourages further investigation of the subject, and suggests improvements that could be extremely useful in the future. A few recommended approaches regarding the project's future scope can be explored for this project which is explained in detail below.

The first is concerned with the accuracy of the models used. Except for the SVM classifier, LinearSVC, most models generated a better prediction accuracy. Several hyperparameter settings, such as class_weight, and max_iter, could be included in the minimally performing models while building and can be validated if there is any increase in the model's accuracy score. Although the inclusion of hyperparameters cannot guarantee a staggering improvement in accuracy, it is recommended to implement this approach to validate if it leads to any positive outcome.

Over the past several years, deep learning has become the most popular technique in solving AI problems. The primary reason for this is that deep learning has shown superior performance on a wide variety of tasks. Thus, this project can also reap some benefits by implementing deep learning. In addition to improved performance, deep learning can scale more effectively with data than classical machine learning algorithms when the data becomes vast. In addition, the feature engineering step can be skipped, if not entirely, but to some extent, as deep networks can help achieve good performance without performing the feature engineering process. Deep learning models such as the Convolutional neural network (CNN) can learn from the short-term sequence in the data and can be explored further and utilized in this project

7.4 Contributions and Impacts on Society

Credit card fraud is a growing concern for financial institutions. As previously stated, statistics show it is one of the most prevalent kinds of fraud on its way to toppling the existing

ranking of frauds. Fraudsters are constantly coming up with new ways to commit fraud. An effective and robust classifier that provides accurate prediction is required to deal with fraudulent threats. The primary goal of a fraud detection system is to make accurate predictions while reducing the number of false positives. This project proposed robust machine-learning solutions while keeping the goals mentioned above in mind. The performance of a machine learning solution depends on a business type, and here, input data plays a vital role in driving the proposed machine learning model. In credit card fraud detection, several dominant factors, such as the number of features, transactions, and the correlation between the features, are significant in determining a model's performance. Before arriving at an efficient machine-learning solution, this project met the prerequisites mentioned above. The machine learning models developed here for credit card fraud detection achieved better performance and accuracy by incorporating SVM, Logistic regression, Random forest, Decision tree, Naive Bayes, and KNN.

Detecting fraudulent activities benefits financial institutions and can assist a wide range of industries that deal with online transactions on a regular basis. Any fraud experienced by a consumer during the transaction process results in a breach of trust. This breach of trust results in the loss of valuable customers as well as financial losses for a business. One solution is to ensure a safe and secure transaction process. This project's credit and fraud detection solution will be immensely useful in detecting and preventing fraudulent activities. The accuracy delivered by the models proves that it provides a stress-free transaction process, and it thus helps curb the loss to a business and mitigate the risks a financial institution may face.

References

Sumanth, C., Kalyan, P. P., Ravi, B., & Balasubramani., S. (2022). Analysis of Credit Card Fraud Detection using Machine Learning Techniques. *2022 7th International Conference on Communication and Electronics Systems (ICCES)*. https://doi.org/10.1109/icces54183.2022.9835751

Saddam Hussain, S. K., Sai Charan Reddy, E., Akshay, K. G., & Akanksha, T. (2021). Fraud Detection in Credit Card Transactions Using SVM and Random Forest Algorithms. *2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*. https://doi.org/10.1109/i-smac52330.2021.9640631

Lucas, Y., Portier, P. E., Laporte, L., Calabretto, S., He-Guelton, L., Oble, F., & Granitzer, M. (2019). Dataset Shift Quantification for Credit Card Fraud Detection. *2019 IEEE Second International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*. https://doi.org/10.1109/aike.2019.00024

Khatri, S., Arora, A., & Agrawal, A. P. (2020). Supervised Machine Learning Algorithms for Credit Card Fraud Detection: A Comparison. *2020 10th International Conference on Cloud Computing, Data Science &Amp; Engineering (Confluence)*. https://doi.org/10.1109/confluence47617.2020.9057851

Wen, H., & Huang, F. (2020). Personal Loan Fraud Detection Based on Hybrid Supervised and Unsupervised Learning. *2020 5th IEEE International Conference on Big Data Analytics (ICBDA)*. https://doi.org/10.1109/icbda49040.2020.9101277

Naveen, P., & Diwan, B. (2020). Relative Analysis of ML Algorithm QDA, LR and SVM for Credit Card Fraud Detection Dataset. *2020 Fourth International Conference on I-SMAC*

(IoT in Social, Mobile, Analytics and Cloud) (I-SMAC). https://doi.org/10.1109/i-smac49090.2020.9243602

Rathore, A. S., Kumar, A., Tomar, D., Goyal, V., Sarda, K., & Vij, D. (2021b). Credit Card Fraud Detection using Machine Learning. *2021 10th International Conference on System Modeling &Amp; Advancement in Research Trends (SMART)*. https://doi.org/10.1109/smart52563.2021.9676262

Rai, A. K., & Dwivedi, R. K. (2020). Fraud Detection in Credit Card Data using Unsupervised Machine Learning Based Scheme. *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*. https://doi.org/10.1109/icesc48915.2020.9155615

Kirar, J. S., Kumar, D., Chatterjee, D., Patel, P. S., & Nath Yadav, S. (2021). Exploratory Data Analysis for Credit Card Fraud Detection. *2021 International Conference on Computational Performance Evaluation (ComPE)*. https://doi.org/10.1109/compe53109.2021.9751922

Tanouz, D., Subramanian, R. R., Eswar, D., Reddy, G. V. P., Kumar, A. R., & Praneeth, C. V. N. M. (2021). Credit Card Fraud Detection Using Machine Learning. *2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)*. https://doi.org/10.1109/iciccs51141.2021.9432308

Jain, V., Agrawal, M., & Kumar, A. (2020). Performance Analysis of Machine Learning Algorithms in Credit Cards Fraud Detection. *2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*. https://doi.org/10.1109/icrito48877.2020.9197762

Consumer Sentinel Network. (2020). In https://www.ftc.gov/system/files/documents/reports/consumer-sentinel-network-data-book-2020/csn_annual_data_book_2020.pdf. Federal Trade Commission.

Wikipedia contributors. (2022, October 19). *Support vector machine*. Wikipedia. https://en.wikipedia.org/wiki/Support_vector_machine

Kanade, V. (2022, April 18). *What Is Logistic Regression? Equation, Assumptions, Types, and Best Practices*. https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-logistic-regression/

Kalyanakrishnan, S. (2014). *Figure 1 from On Building Decision Trees from Large-scale Data in Applications of On-line Advertising | Semantic Scholar*. https://www.semanticscholar.org/paper/On-Building-Decision-Trees-from-Large-scale-Data-in-Kalyanakrishnan-Singh/53391c2d071f14b1f4710c144951e5d6b1bfe188/figure/0

VanderPlas, J. (n.d.). *In Depth: Naive Bayes Classification | Python Data Science Handbook*. https://jakevdp.github.io/PythonDataScienceHandbook/05.05-naive-bayes.html

K-Nearest Neighbors (KNN) - PRIMO.ai. (n.d.). http://primo.ai/index.php?title=K-Nearest_Neighbors_(KNN)

Chauhan, A. (2021, December 31). *Random Forest Classifier and its Hyperparameters - Analytics Vidhya*. Medium. https://medium.com/analytics-vidhya/random-forest-classifier-and-its-hyperparameters-8467bec755f6

Natural Language Processing (NLP) Project Example for Beginners. (2022, June 20). Medium. https://medium.com/analytics-vidhya/natural-language-processing-nlp-project-example-for-beginners-616549300c54