

Credit Card Fraud Detection

Department of Applied Data Science, San Jose State University

DATA 245: Machine Learning Technology

Prof. Shih Yu Chang

Group 4

Nupur Pathak (016043716)

Revathi Boopathi (015977195)

Sree Divya Cheerla (016018808)

Vani Bhat (015972606)

Acknowledgement

We would like to express our deep gratitude to Professor Shih Yu Chang for guiding us throughout the project, his enthusiastic encouragement, his advice and assistance in keeping our progress on schedule. We would also like to extend our thanks to our Instructional Student Assistant (ISA) Bailey Wang for channeling our doubts in the right direction.

Table of Contents

Acknowledgement.....	2
1. Introduction	5
1.1 Project Background and Problem Definition	5
1.2 Project Objectives	5
1.3 Project Requirements	6
1.4 Project Deliverables	6
1.5 Technology and Solution Survey	7
1.6 Literature survey of existing research	10
2. Data and Project Management Plan	12
2.1 Data Management Plan.....	12
2.2 Project Development Methodology	13
2.2.1 CRISP-DM.....	13
2.3 Project Organization Plan	14
2.4 Project Resource Requirement and Plan	14
2.5 Project Schedule	15
3. Data Engineering	16
3.1 Data Process	16
3.2 Data Collection	17
3.3 Data Pre-Processing	19
3.4 Data Transformation	21
3.5 Data Splitting	26
3.6 Data Statistics	26
3.6.1 Data Cardinality	26
3.6.2 Data Quality Report	26
3.6.3 Analytical Base Table	28
3.7 Data Analytics Results	29
4 Model development	32
4.1 Model Proposals	32
4.1.1 Naive Bayes.....	33
4.1.2 Logistic regression.....	34
4.1.3 Ensemble learning	36
4.1.4 KNN.....	38

4.1.5 SVM.....	39
4.1.6 Convolutional Neural Network (CNN)	41
4.1.8 Artificial Neural Network (ANN).....	42
4.2 Model Comparison	44
4.3 Model Evaluation Methods	44
4.4 Model Validation and Evaluation Results	45
4.5 Model Results Discussion.....	47
5 System Design and Architecture	48
5.1 System Design & Architecture	48
5.2 System Supporting Environment	49
6. System Evaluation and Visualization.....	50
6.1 Analysis of Model Execution and Evaluation Results.....	50
6.2 Achievements and Constraints.....	50
6.3 System Quality Evaluation of Model Functions and Performance	51
6.4 System Visualizations	51
7 Evaluation and Reflection	52
7.1 Benefits and Shortcomings	52
7.2 Experience and Lessons Learned	52
7.3 Recommendations for Future Work	53
7.4 Contributions and Impacts on Society	53
References.....	54

1. Introduction

1.1 Project Background and Problem Definition

There is a huge advancement in Science and Technology in the recent decade. The rapid development in e-commerce, tap and pay systems, and e-payment methods resulted in a tremendous increase in financial frauds. Different types of financial frauds include unauthorized banking, investment frauds, identity thefts, phishing, advance fee fraud, credit card transaction frauds and others. In 2022, 46% of companies reported experiencing fraud, corruption and economic crimes compared to 47 % in 2020 and 49% in 2018. Digital fraud has impacted 38% of Americans in Q1 2022, 29% of people experienced phishing, followed by 26% for stolen credit cards. As fraudsters are increasing every day, it is important to identify such frauds and take necessary precautions to avoid them. Big companies are taking crucial steps and precaution in order to save their customer loyalty. So, there remains a need for formulating fraud detection systems.

1.2 Project Objectives

The objective of the project Credit Card Fraud Detection is to identify fraudulent transactions and prevent them from being processed. This is achieved by using various machine learning and deep learning techniques where several features like the date, time, place and amount used for the transaction are taken as input and is classified as fraud/non-fraud. These can be used by major firms in order to take necessary actions which will save genuine customers from fraudulent activities. We will be addressing following areas in this project:

- Credit card payment processing landscape understanding and identifying patterns in fraud transactions
- Evaluation of machine learning and deep learning models
- Building a payment fraud detection system
- Dashboard to test the query instances for fraudulent transactions and showcase outcomes from the models

1.3 Project Requirements

The objective of the project is to classify the incoming transaction as fraudulent or non-fraudulent based on several features like the date, time, place and amount used for the transaction.

Data requirements include a suitable dataset consisting of a history of fraudulent and non-fraudulent transactions. It should consist of features that can help Machine learning algorithms to train and identify and classify the type of transaction.

Functional requirements include processing each transaction record and classifying it as a fraudulent or non-fraudulent transaction. The credit card transaction dataset has a target label 'is_fraud' which is an indicator of fraud and non_fraud transactions. Building a machine learning model is necessary which learns based on historical data points. Once the model is developed and trained, it should be able to identify the transaction in real-time as fraudulent or non-fraudulent. Machine Learning (ML) requirement includes the proper selection of machine learning algorithms and evaluation metrics to compare and record their performances. It also requires the model to be deployed to a real-time environment to get real-time predictions.

1.4 Project Deliverables

- Project proposal with the objective, background, and literature.
- Data domain concepts and data quality report.
- Data engineering documentation with transaction data pre-processing, transformation, and preparation.
- Deliver mid-term presentation documentation
- Model implementation and assessment.
- Model development and evaluation results documentation.
- Deploy Power BI dashboard
- Deliver final presentation documentation
- Deliver final project report and code implementation of the project

1.5 Technology and Solution Survey

With the development in e-commerce, tap and pay systems, and e-payment methods there has been a tremendous increase in financial frauds. Credit card fraud is a pervasive and costly problem for both customers and financial institutions. In response, financial institutions and technology companies have developed a variety of solutions to detect and prevent credit card fraud. There has been a lot of research going on to identify these frauds by employing Machine Learning models like KNN, SVM, Linear Regression, AdaBoosting, Gradient Boosting etc. Most of the time supervised machine learning models are used as the dataset will be labeled. However, recent development has enabled usage of unsupervised and reinforced machine learning models for fraud detection. This survey aims to explore the current state of technology and solutions for credit card fraud detection. From the Table, various machine learning models used by financial institutions to detect frauds are surveyed.

Figure 1

Technological Survey

Machine Learning Model	Performance Metrics	Advantages	Limitations
Decision Tree	Sensitivity – 79.21% Precision – 85.11% (Khatri et al. 2018) Accuracy – 90.87% Precision – 91% (Dhankhad et al. 2018) Accuracy – 91.12% Precision – 86.95% (D. Tanouz et al. 2021)	<ul style="list-style-type: none">• Simple to understand and interpret• Can handle both numerical and categorical data• Performs well on large datasets	<ul style="list-style-type: none">• Can be easily overfit• Prone to bias• Can be unstable, so small changes might impact the decision tree
KNN	Sensitivity – 81.19% Precision – 91.11% (Khatri et al. 2018) Accuracy – 94.25% Precision – 91%	<ul style="list-style-type: none">• Simple to implement and interpret• No need for extensive data preparation such as feature scaling	<ul style="list-style-type: none">• Computationally intensive• Affected by curse of dimensionality• Changes with the distance metric and the value of k

	<p>(Dhankhad et al. 2018)</p> <p>Accuracy – 99.95% F1 score – 85.71% (Singh et al. 2021)</p>		
Logistic Regression	<p>Sensitivity – 63.34% Precision – 87.67% (Khatri et al. 2018)</p> <p>Accuracy – 93.91% Precision – 94% (Dhankhad et al. 2018)</p> <p>Accuracy – 95.16% Precision – 95.34% (D. Tanouz et al. 2021)</p> <p>Accuracy – 99.91% F1 score – 73.56% (Singh et al. 2021)</p>	<ul style="list-style-type: none"> • Simple and fast to train • Can handle large datasets • Provides probabilities for each class 	<ul style="list-style-type: none"> • Assumes linear relationships between features and the target variable which might not be the case • Can struggle with non-linear and complex datasets • Can be affected by outliers
Random Forest	<p>Sensitivity – 75.25% Precision – 93.83% (Khatri et al. 2018)</p> <p>Accuracy – 94.59% Precision – 95% (Dhankhad et al. 2018)</p> <p>Accuracy – 96.77% Precision – 100% (D. Tanouz et al. 2021)</p> <p>Accuracy – 99.92% F1 score – 75.70% (Singh et al. 2021)</p>	<ul style="list-style-type: none"> • Can handle large and complex datasets with high dimensionality • Handles noise and outliers robustly • provides feature importance scores 	<ul style="list-style-type: none"> • computationally intensive to train and use • difficult to interpret and explain • Affected by unbalances and imbalanced datasets

Naïve Bayes	<p>Sensitivity – 85.15% Precision – 6.56% (Khatri et al. 2018)</p> <p>Accuracy – 90.54% Precision – 91% (Dhankhad et al. 2018)</p> <p>Accuracy – 95.16% Precision – 100% (D. Tanouz et al. 2021)</p>	<ul style="list-style-type: none"> • Requires less training data • Widely used for multiclass prediction problems • It is a fast-learning algorithm so can be used for real time predictions • Famous for spam filtering, sentimental analysis etc. 	<ul style="list-style-type: none"> • It considers that all the variables are independent of each other as it follows bayes theorem which will not be the scenario in real-time. • Not suitable for highly correlated features
SVM	<p>Sensitivity – 99.93% F1 score – 85.71% (Alarfaj et al. 2022)</p> <p>Accuracy – 93.24% Precision – 93% (Dhankhad et al. 2018)</p> <p>Accuracy – 99.92% F1 score– 77.01% (Singh et al. 2021)</p>	<ul style="list-style-type: none"> • Can handle large, complex datasets with high dimensionality • Sensitive to the choice of kernal and hyperparameters 	<ul style="list-style-type: none"> • Not suitable for imbalances and noisy datasets • Difficult to interpret and understand.
CNN	<p>Accuracy – 96.34% (Alarfaj et al. 2022)</p>	<ul style="list-style-type: none"> • Can learn complex, hierarchical patterns in data • Good at recognizing spatial relationships in data 	<ul style="list-style-type: none"> • Require large amount of memory and computational power to train and use • Can be sensitive to the choice of hyperparameters and require careful tuning • Difficult to interpret and understand the learned patterns

ANN	Accuracy – 93.7% (Pradhan et al. 2021)	<ul style="list-style-type: none"> • Can learn complex, non-linear relationships in data • Can be applied to a wide range of tasks • Can learn both unstructured and unlabeled data 	<ul style="list-style-type: none"> • Require large amount of memory and computational power • Can be sensitive to the choice of hyperparameters and require careful tuning • Can be prone to overfitting • Difficult to interpret and understand the learned patterns
-----	---	--	---

1.6 Literature survey of existing research

Credit card fraud is a growing problem and with financial losses from these frauds costing billions of dollars every year. Machine learning models have a potential to provide effective and efficient fraud detection, so it is important to review existing literature to understand the state of the field and to identify potential areas for future work. This literature survey will provide an overview of the use of machine learning for credit card fraud detection, including the types of the models used, their performance and any challenges or limitations.

Dhankhad et al. (2018) in this paper talk about how credit card frauds are increasing at an alarming rate and that advancement in technology has helped detect these frauds. They have employed various supervised machine learning models to detect credit card frauds and have implemented super classifiers using ensemble methodology. They have discussed each of the machine learning models used like Logistic regression, Decision tree, naïve bayes, KNN, SVM, Gradient boost, and XG boost classifier and evaluated their performance by using metrics like accuracy, precision, f1 score, recall etc. They observed that the stacking classifier outperformed all the other models with 95.27% accuracy followed by random forest with 94.59 % accuracy.

D. Tanouz et al. (2021) in their paper discuss that with development of technology, the frauds are also increasing, and it is important to develop an efficient fraud detection algorithm to detect the frauds accurately and in time. They have implemented various machine learning

models like Logistic regression, random forest, and Naïve bayes on data which is highly unbalanced. They have performed under sampling of the dataset to make it balanced and used outlier data mining technique to avoid bias and outliers from the dataset to achieve better accuracy. The authors then compared the performance of the models by obtaining accuracy, precision, recall and f1 scores. They noticed that random forest algorithms performed better with 96.77% accuracy.

Khatri et al. (2018) in their paper discuss that cashless payments which increasing now a days has made it easy for anyone to get sensitive information and thereby increasing in digital frauds. They have deployed some of the machine learning models like Decision tree, KNN, Logistic regression, random forest, and Naïve bayes to help tackle this problem to reduce the fraudulent transactions. These algorithms are trained on data collected previously and analyze it to identify patterns and trends in the data making it extremely useful in identification of frauds. They have compared the performances of these models' using sensitivity and precision metrics and observed that Decision tree model is best suited for fraud detection even though KNN model gave better precision as KNN took more time and time plays an important role in fraud detection.

Alarfaj et al. (2022) in their paper discuss how it is important to detect fraudulent transactions, changes in the nature of the frauds, and identify false alarms. They have initially applied various state-of-the-art machine learning algorithms like Decision tree, Random Forest, Support vector machine, Logistic regression, and XG boost but as they give low accuracy, they have applied various deep learning models like Convolutional neural networks (CNN), Long short-term memory (LSTM), and Residual neural network (RNN). They have noticed that deep learning models have outperformed the traditional machine learning models.

Singh et al. (2021) in this paper talk about how by incorporating previous transaction details they have used Machine learning models to identify fraudulent purchase/order. They have identified the fraud transactions by analyzing inconsistent location calculations for every transaction by employing ML models like SVM, Logistic regression, KNN, and Random Forest. By using credit card transaction details of European card holders available on Kaggle they have performed pre-preprocessing on the data by normalizing it and then split for training and testing. By employing the Machine Learning models mentioned above they have identified the fraudulent transactions in the data and then evaluated the performance of the models by comparing Accuracy score, F1 score and confusion matrix for different models.

Agarwal et al. (2015) in their paper talk about how fraudulent transactions are increasing day by day and how they have employed techniques like Genetic Algorithm, Behavior Based Technique, and Hidden Markov models to limit them. They have collected information like registration details, login details, banking details, and others during the online transactions and fed them to the models. The Hidden Markov model maintains the log of all the transactions done previously by the customer, Behavior Based Technique based on the transaction creates clusters like low, medium, and high profile. Finally, the Genetic algorithm calculates the threshold value. They have deployed these models individually and then by taking the average of the values obtained from these three models, if the value is above the threshold value, they have defined that fraud has occurred.

2. Data and Project Management Plan

2.1 Data Management Plan

The components of AWS present as a part of the data management plan in the below section is considering the ideal application that would be built. Data has been synthetically generated using a Sparkov data generator. A single file is generated for each customer and all the generated files are finally combined. The data that's collected will be stored in the S3 bucket of AWS. From there, the data is further moved for the preprocessing stage. Data will be stored in .csv format. As a part of the future once the process is implemented, a daily directory of files will be created which will further store customer transaction history from which an incremental update happens to the consolidated data file. Also, log files will be maintained by enabling bucket versioning in AWS S3 in order to ensure easy debugging in case of any issues.

The data will be transferred to AWS Glue for further pre-processing and output back to the S3 bucket. From then, the data is moved to SageMaker in order to implement the classification modeling steps. Then once the predictions are generated, they are added as a separate field to the input data and are used by the Power BI visualization. AWS Athena will be used to ensure data quality by performing interim queries. A separate data quality framework will be built in order to ensure data consistency, conformity, uniqueness, etc.

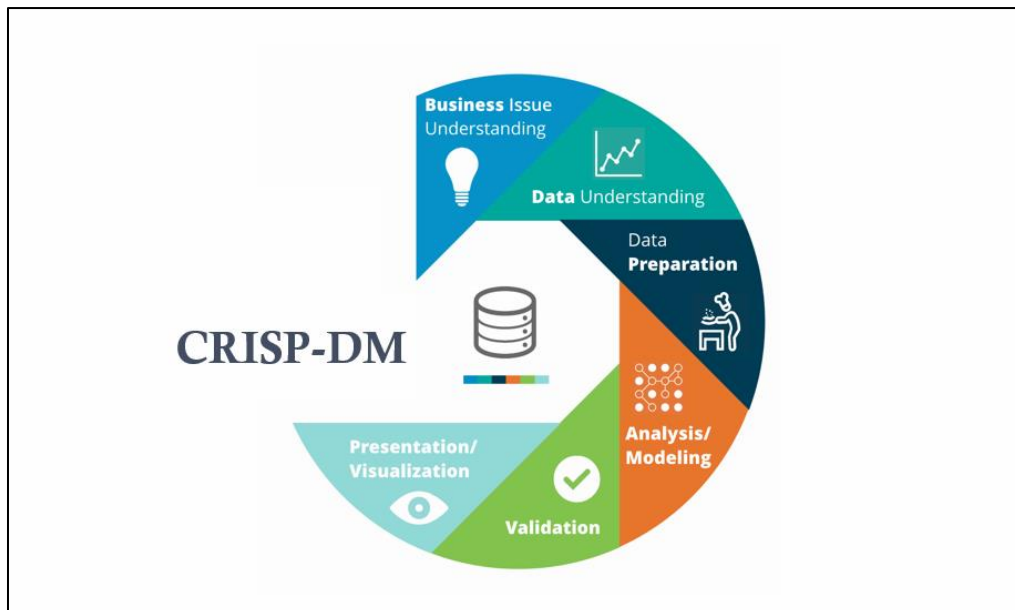
2.2 Project Development Methodology

2.2.1 CRISP-DM

Figure 2 shows the structure that has been used to solve the problem of credit card fraud detection. CRISP-DM framework is leveraged to solve the problem. It starts with the business understanding phase which involves requirements gathering and framing domain concepts where all the features for predicting credit card fraud are identified. The domain concepts include transaction related features like how recently a shopper has purchased, what is the transaction value, etc. Other demographic features include the shopper's gender, tenure with the firm, location of shop, etc. Post formulating the domain concepts, the data understanding phase involves collecting the required data for each of the domain concepts. The data is synthetically generated using a Sparkov data generator. The data preparation phase involves making enough transformations to prepare the data for modeling. The modeling phase involves implementing supervised machine learning like logistic regression, KNN, ensemble learning, etc. Post this, the model validation phase involves evaluation of classification metrics by comparing various model results. Finally, the presentation/ visualization phase involves visualizing the modeling results.

Figure 2

CRISP-DM

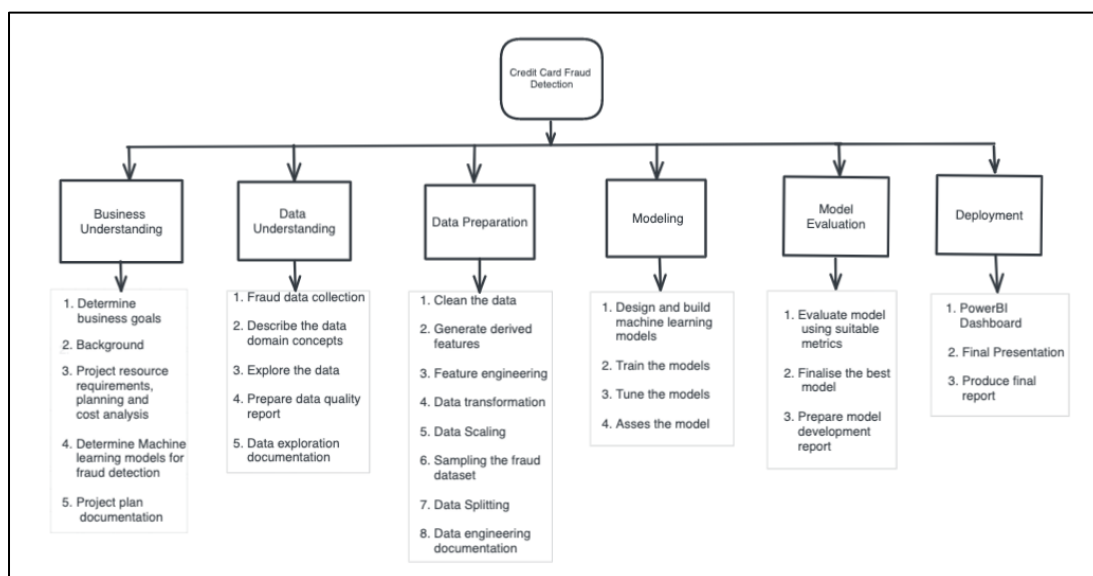


2.3 Project Organization Plan

The Work Breakdown Structure (WBS) in the below figure outlines the high level plan for the tasks involved in the end to end project. The tasks are divided into 6 phases based on the CRISP-DM methodology as shown below in Figure 3.

Figure 3

Work Breakdown Structure



2.4 Project Resource Requirement and Plan

The project requires proper planning with respect to resources that are needed for the successful implementation of the project keeping in mind the overall cost associated with the project. The below Figure 4 outlines the necessary hardware/software/tools needed for this project.

Figure 4

Hardware, Software, and Tools

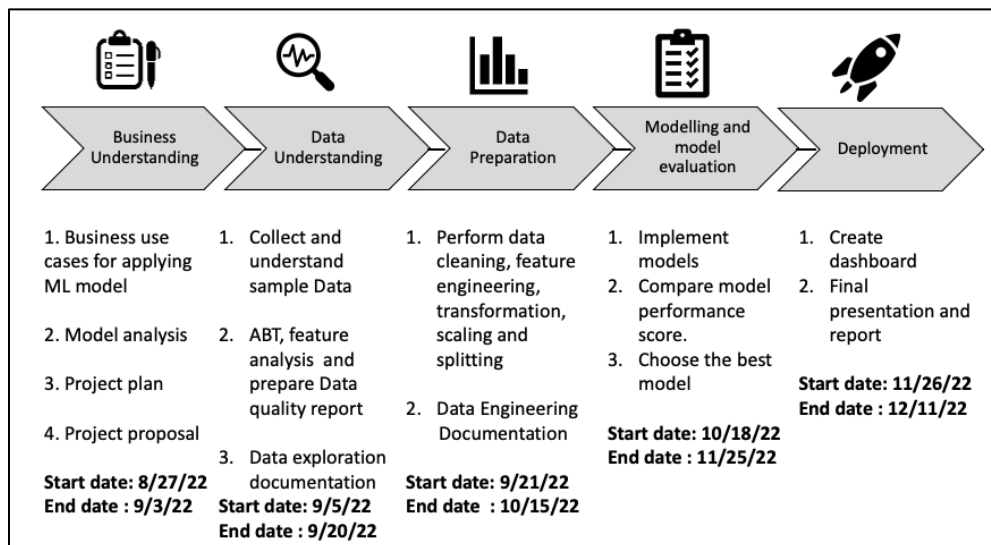
Hardware/Software/Tools	Configuration	Purpose
Storage	Local	Store the credit card transaction dataset
Anaconda distribution package	Python 3.9	Integrated development environment(IDE)
GitHub	Desktop version	Version control and code storage repository
ClickUp	Desktop version	Project management tool for task allocation
Jupyter Notebook	3.9	IDE for model development
PowerBI	Desktop version	For exploratory data analysis and prediction visualizations

2.5 Project Schedule

Project involves various stages of understanding business and data, preparing the data for modeling, model implementation and evaluation and finally deployment. It spans from the end of August to mid of December. Below Figure 5 outlines the important activities in each stage and the respective start and end dates.

Figure 5

Phase-wise Project Implementation



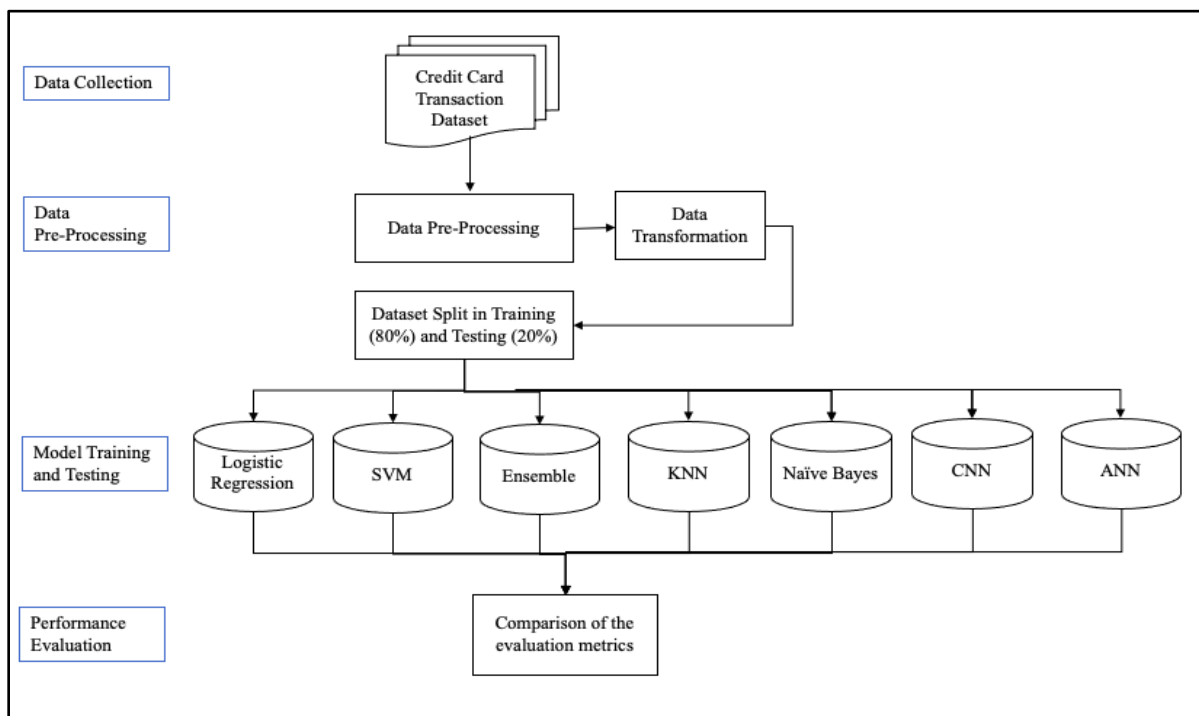
3. Data Engineering

3.1 Data Process

It is hard to get data on credit card fraud transactions majorly because of compliance standards set by various financial firms. So, the only two available approaches were to use existing data from data sources like Kaggle or generate data synthetically using generators. The data for this project has been derived using the latter. Exploratory data analysis is then performed on the data to understand and gain insights about the data. Data is then cleaned, and several new features have been derived from the existing features as well as the target features that are important for model development. The data is transformed and scaled to provide patterns that are easier to understand by the machine learning models. A data imbalance check is performed and is down-sampled to avoid bias in the model and improve the correlation between features. Finally, the data is made model ready by splitting it into training and testing sets to be used by the machine learning model. The end-to-end process flow has been outlined in the below Figure 6.

Figure 6

Process Flow



3.2 Data Collection

The synthetic data for credit card transactions is generated using Sparkov Data Generation Tool. This tool creates transaction files for the customer transactions based on the profile of the customers that comprises both fraud and non-fraud transactions. This data is generated for transactions spanning two years for 1,000 customers as shown in Figure 7. Following parameters are tuned for the datagen script:

- Number of customers to generate: 1,000
- Output folder: data/training set/ medium
- Start Date: 1 Jan 2021
- End Date:

Figure 7

Generation of synthetic data for customer transactions using the Sparkov Data Generator

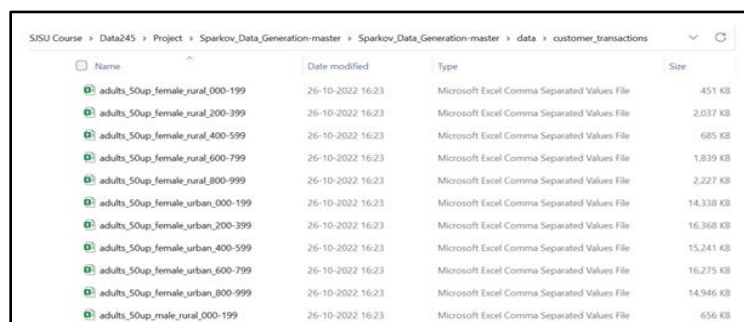
```
Microsoft Windows [Version 10.0.22000.1219]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Nupur>cd C:\Users\Nupur\Documents\SJSU Course\Data245\Project\Sparkov_Data_Generation-master\Sparkov_Data_Generation-master

C:\Users\Nupur\Documents\SJSU Course\Data245\Project\Sparkov_Data_Generation-master\Sparkov_Data_Generation-master>python datagen.py -n 1000 -o "data/customer_transactions"
01-01-2020 12-31-2021
Num CPUs: 8
profile: adults_50up_male_urban.json, chunk size: 200, chunk: 0-199
profile: adults_50up_male_urban.json, chunk size: 200, chunk: 200-399
profile: adults_50up_male_urban.json, chunk size: 200, chunk: 400-599
profile: adults_50up_male_urban.json, chunk size: 200, chunk: 600-799
profile: adults_50up_male_urban.json, chunk size: 200, chunk: 800-999
profile: adults_50up_female_urban.json, chunk size: 200, chunk: 0-199
profile: adults_50up_female_urban.json, chunk size: 200, chunk: 200-399
profile: adults_50up_female_urban.json, chunk size: 200, chunk: 400-599
profile: adults_50up_female_urban.json, chunk size: 200, chunk: 600-799
profile: adults_50up_female_urban.json, chunk size: 200, chunk: 800-999
profile: adults_50up_male_rural.json, chunk size: 200, chunk: 0-199
profile: adults_50up_male_rural.json, chunk size: 200, chunk: 200-399
profile: adults_50up_male_rural.json, chunk size: 200, chunk: 400-599
profile: adults_50up_male_rural.json, chunk size: 200, chunk: 600-799
profile: adults_50up_male_rural.json, chunk size: 200, chunk: 800-999
```

Figure 8

Transaction files generated by the Sparkov Data Generator



Name	Date modified	Type	Size
adults_50up_female_rural_000-199	26-10-2022 16:23	Microsoft Excel Comma Separated Values File	451 KB
adults_50up_female_rural_200-399	26-10-2022 16:23	Microsoft Excel Comma Separated Values File	2,037 KB
adults_50up_female_rural_400-599	26-10-2022 16:23	Microsoft Excel Comma Separated Values File	685 KB
adults_50up_female_rural_600-799	26-10-2022 16:23	Microsoft Excel Comma Separated Values File	1,839 KB
adults_50up_female_rural_800-999	26-10-2022 16:23	Microsoft Excel Comma Separated Values File	2,227 KB
adults_50up_female_urban_000-199	26-10-2022 16:23	Microsoft Excel Comma Separated Values File	14,338 KB
adults_50up_female_urban_200-399	26-10-2022 16:23	Microsoft Excel Comma Separated Values File	16,368 KB
adults_50up_female_urban_400-599	26-10-2022 16:23	Microsoft Excel Comma Separated Values File	15,241 KB
adults_50up_female_urban_600-799	26-10-2022 16:23	Microsoft Excel Comma Separated Values File	16,275 KB
adults_50up_female_urban_800-999	26-10-2022 16:23	Microsoft Excel Comma Separated Values File	14,946 KB
adults_50up_male_rural_000-199	26-10-2022 16:23	Microsoft Excel Comma Separated Values File	656 KB

The transaction files seen in Figure 8 are consolidated to form the dataset for our project. The customer demographics can be seen in Figure 10.

Figure 9

Consolidated transactions dataset

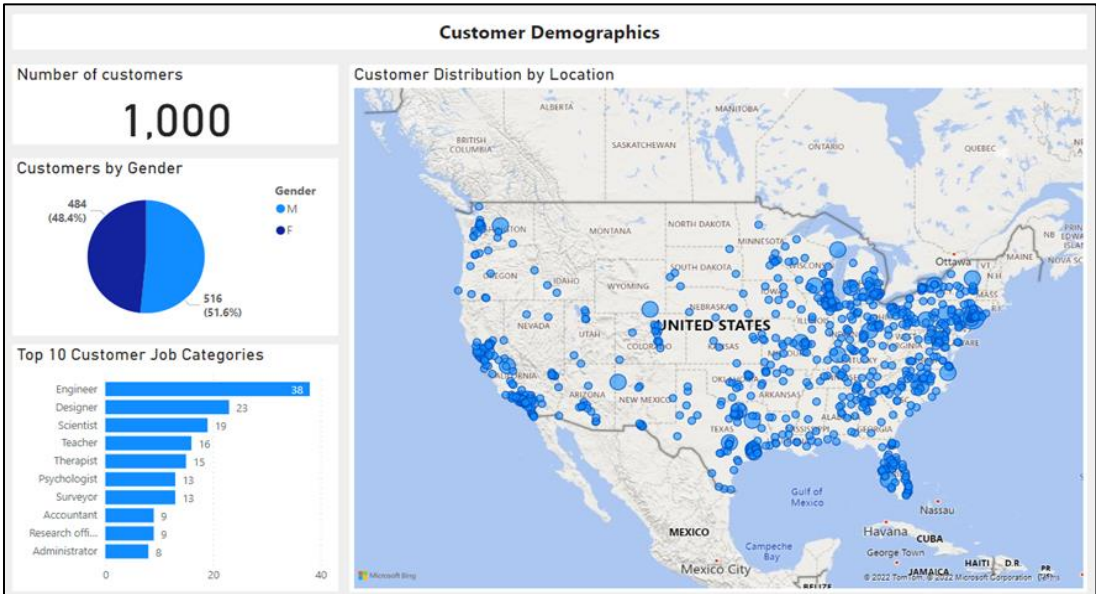
df.head()																						
	ssn	cc_num	first	last	gender	street	city	state	zip	lat	...	trans_num	trans_date	trans_time	unix_time	category	amt	is_fraud	merchant	merch_lat	merch_long	
0	195-33-0728	3508835615951480	Elen	Ortiz	F	482 Robert Light Apt. 994	Columbia Falls	ME	4623 44 6699	...	5c15602a5c509764ed50149a78e279a	2021-04-29	00:07:35	1.619680e+09	gas_transport	6.73	1.0	fraud_Jenkins, Hauck and Friesen	44.084527	-67.954129		
1	195-33-0728	3508835615951480	Elen	Ortiz	F	482 Robert Light Apt. 994	Columbia Falls	ME	4623 44 6699	...	549f7a7671e25c7a1832952ccf999	2021-04-29	07:10:46	1.619705e+09	gas_transport	12.74	1.0	fraud_Zieme, Bode and Dooley	45.584705	-67.692301		
2	195-33-0728	3508835615951480	Elen	Ortiz	F	482 Robert Light Apt. 994	Columbia Falls	ME	4623 44 6699	...	844caa31863ba1973a63d3666c47f0a	2021-04-29	10:09:12	1.619716e+09	gas_transport	9.53	1.0	fraud_King Inc	45.380492	-67.385962		
3	195-33-0728	3508835615951480	Elen	Ortiz	F	482 Robert Light Apt. 994	Columbia Falls	ME	4623 44 6699	...	a77272e5cb108c1c9da38c5fcea95d7	2021-04-29	09:15:39	1.619713e+09	grocery_pos	10.95	1.0	fraud_Miller-Hauck	45.350439	-67.644179		
4	195-33-0728	3508835615951480	Elen	Ortiz	F	482 Robert Light Apt. 994	Columbia Falls	ME	4623 44 6699	...	e7594588105594902276aa61b960c919	2021-04-29	00:22:42	1.619681e+09	grocery_pos	353.18	1.0	fraud_McDemott-Weinann	45.424536	-67.852448		

5 rows x 25 columns

It can be seen that the customer dataset comprises 1,000 customers. The number of customers by gender are equally distributed between male and female genders with 484 females and 516 males. Engineer, designer, scientist, teacher and therapist are the top job profiles of the customers.

Figure 10

Customer demographics details



3.3 Data Pre-Processing

Once the data is collected and explored, as part of data pre-processing new features which are important for model development have been derived. Newly generated features are listed below.

- From the profile column of the customer, a new feature 'area'(urban/rural) has been derived. This can give an idea if fraudulent transactions are more prone in urban/rural areas.
- From the dob (date of birth) column, the 'age' of the customer has been derived. This can help in identifying which age range people are more targeted to fraud.
- From the latitude and longitude column of the customer and merchant, the distance between the customer and merchant where the transaction happened is captured in a new column 'distance'. This can give an idea if the fraudulent transactions are happening at a closer or farer distance.
- From the trans_time(transaction time) column, 'hour' and 'hour_type' column has been derived which indicates the hour of the transaction and if the transaction happened at daylight/midnight which is an important feature for the model
- From the trans_date(transaction date) column, 'month', 'hol_month' and 'is_weekend' column has been derived which has the transaction month. This indicates if the transaction was done in a holiday month like November/December and if the transaction was done on the weekend. This helps the model identify a pattern if fraud is more in the holiday month and/or weekend.
- New column 'recent_shopper' has been derived by using columns 'ssn' and 'trans_date' to get details on the customers who have recently shopped.
- Using merchant latitude, longitude and is_fraud , a new column 'lat_long_type' has been created to flag latitude that is suspicious of fraud. Here, the target variable has been used to identify such places.
- A new column 'categ_type' is created from the 'category' column showing the category where the maximum number of frauds has happened. This can give an indication of which category customers are more prone to fraud.

Figure 11 shows DataFrame after creating new derived features.

Figure 11

Dataframe of the derived features

	gender	city_pop	amt	is_fraud	area	age	distance	hol_month	is_weekend	hour_type	recent_shopper	lat_long_type	categ_type
0	F	39502	317.35	1.0	urban	95	25.136	1.0	0	midnight	past	1	3
1	F	39502	880.85	1.0	urban	95	7.231	1.0	0	daylight	recent	1	3
2	F	39502	962.98	1.0	urban	95	14.856	1.0	0	midnight	recent	1	3
3	F	8399	816.18	1.0	urban	57	35.963	2.0	0	midnight	recent	2	3
4	F	8399	1034.97	1.0	urban	57	45.031	2.0	0	daylight	recent	2	3

Data Cleaning: Once the data is prepared and the data quality report is generated, cleaning steps are performed. As part of cleaning the following operations are performed. Figure 12 shows sample feature status after data cleaning and checking for duplicates.

- Dropped data rows where all data points are null
- Dropped data rows where target variable 'is_fraud' is null
- Dropped data rows where any of the data points are null
- Dropped duplicate data rows

Figure 12

Data Cleaning and Duplicate Check

Post data cleaning	Post duplicate check
<pre>gender: False city_pop: False amt: False is_fraud: False area: False age: False distance: False hol_month: False is_weekend: False hour_type: False recent_shopper: False lat_long_type: False categ_type: False</pre>	<pre># Check for duplicates df.duplicated().value_counts() False 1751721 dtype: int64</pre>

Once all the new features are derived, the chi-square test of independence is performed to find the plausible dependency of newly created features with the target variable. A significance level of 0.05 is selected in order to eliminate less correlated features. The output of the chi-square test shown in Figure 13 shows that the 'area' column doesn't contribute much to the fraudulent transaction classification.

Figure 13

Feature Selection

```
gender 0.011816646789754925 True  
area 0.9952116981974969 False  
hol_month 3.384267476615273e-36 True  
is_weekend 1.711962979716311e-80 True  
hour_type 0.0 True  
recent_shopper 1.9626976385969158e-90 True  
lat_long_type 4.8147231182969436e-111 True  
categ_type 0.0 True
```

3.4 Data Transformation

The dataset consists of many **categorical** features. However, many of the machine learning models usually require numerical values as the input. Hence, the necessary transformation of data from categorical to numerical is a must. In this section, we take multiple measures to convert data to make it suitable for modeling.

1. Label encoding is performed on column type 'categ_type' and 'lat_long_type'
2. One hot encoding is performed on column 'gender', 'hour_type', 'recent_shopper'

Figure shows sample dataframe after data transformation

Figure 14

Transformed Dataframe

	city_pop	amt	is_fraud	age	distance	hol_month	is_weekend	lat_long_type	categ_type	gender_M	hour_type_midnight	recent_shopper_recent
0	39502	317.35	1.0	95	25.136	1.0	0	0	2	0	1	0
1	39502	880.85	1.0	95	7.231	1.0	0	0	2	0	0	1
2	39502	962.98	1.0	95	14.856	1.0	0	0	2	0	1	1
3	8399	816.18	1.0	57	35.963	2.0	0	1	2	0	1	1
4	8399	1034.97	1.0	57	45.031	2.0	0	1	2	0	0	1

Data Scaling: Columns like city_pop, amt., and distance have a wide range of values. It is important to bring the value to the uniform scale before using them for model development. Hence, the features have been range normalized for the range [0,1]

We can see from the plots shown in Figure 15 that most of the features do not follow the Gaussian distribution. Hence, normalization is performed rather than standardization.

Figure 15

Feature Distribution

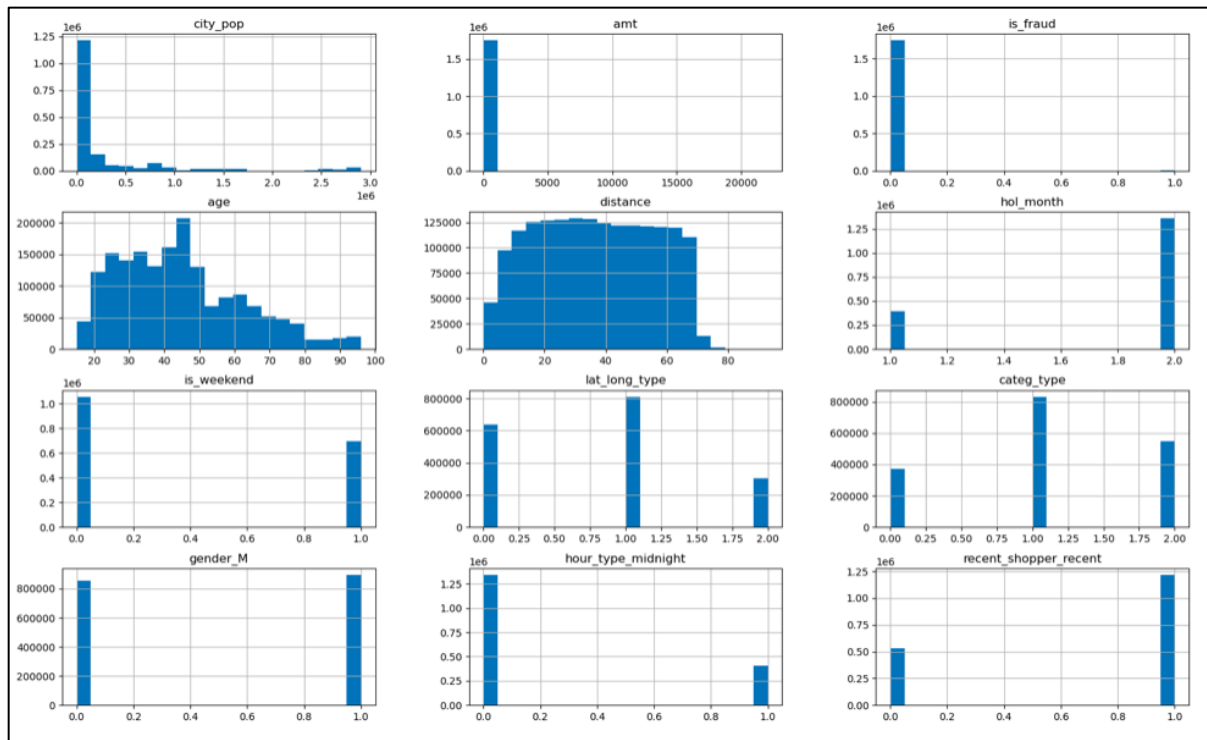


Figure 16

Data frame post Normalization

	city_pop	amt	is_fraud	age	distance	hol_month	is_weekend	lat_long_type	categ_type	gender_M	hour_type_midnight	recent_shopper_recent
0	0.013530	0.014344	1.0	0.987654	0.270223	0.0	0.0	0.0	1.0	0.0	1.0	0.0
1	0.013530	0.039896	1.0	0.987654	0.077621	0.0	0.0	0.0	1.0	0.0	0.0	1.0
2	0.013530	0.043620	1.0	0.987654	0.159642	0.0	0.0	0.0	1.0	0.0	1.0	1.0
3	0.002829	0.036963	1.0	0.518519	0.386687	1.0	0.0	0.5	1.0	0.0	1.0	1.0
4	0.002829	0.046884	1.0	0.518519	0.484230	1.0	0.0	0.5	1.0	0.0	0.0	1.0

Data Sampling: In the credit card transaction dataset, the non-fraud transactions far outweigh the fraudulent transactions as shown in Figure 17. Using this dataset for modeling will induce bias in the results and one can observe the correlation values amongst features are not good enough as shown in Figure 18. Hence, it is important to handle this before model implementation.

Imbalance in data can be fixed using various sampling techniques

Figure 17

Imbalance dataset

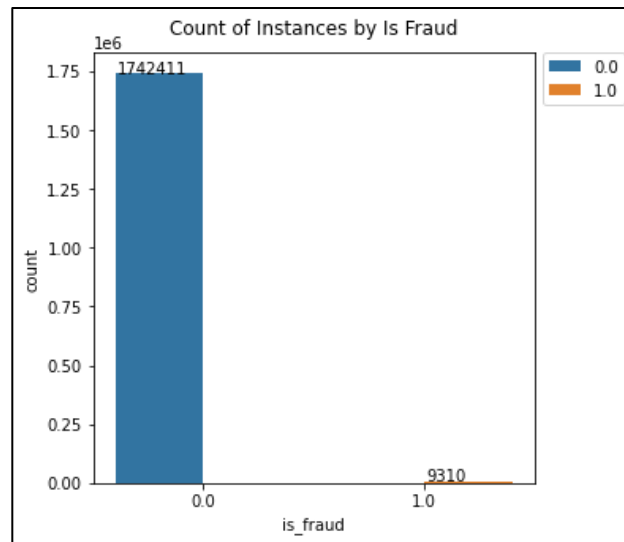
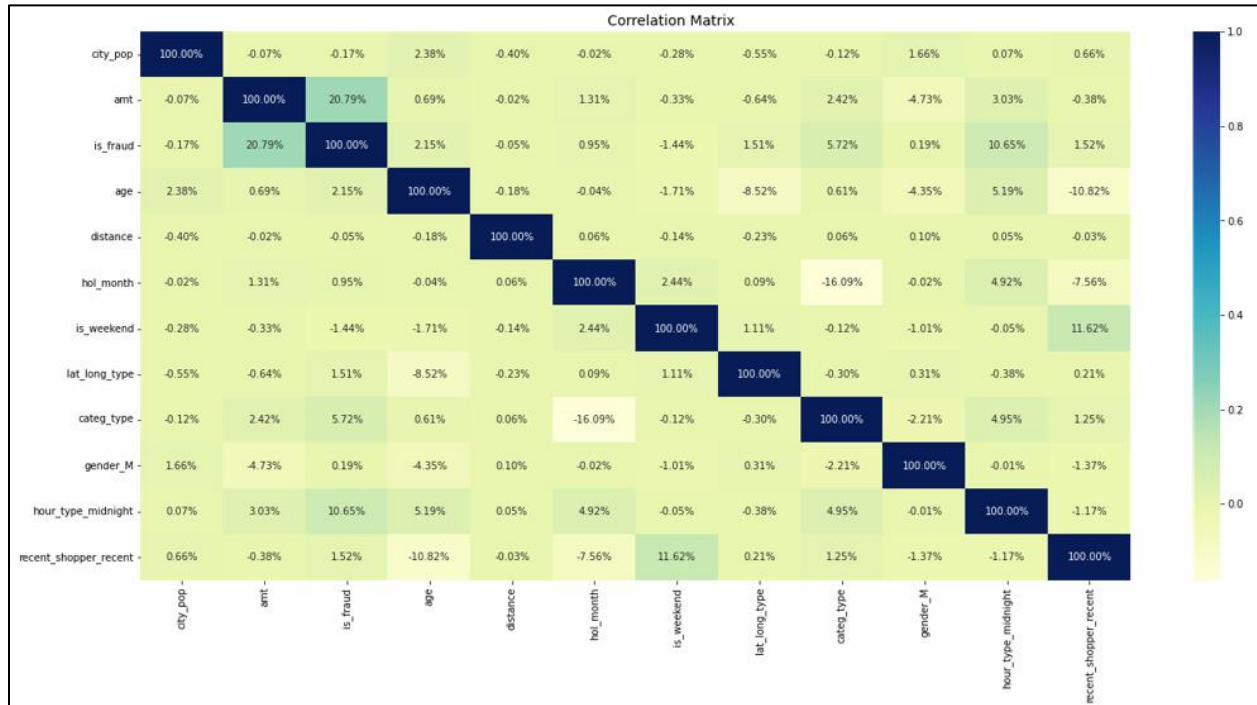


Figure 18

Correlation matrix of imbalanced dataset



There are multiple sampling techniques which can be used to balance the dataset. They are down-sampling and up-sampling. After various experiments on multiple sampling techniques, it was found that down-sampling is ideal for the given dataset and the specified target. Down sampling reduces the samples of the majority class. Hence, the record count reduces. In this case, the majority class of non-fraud was reduced to minority-class fraud to bring balance in the dataset. Hence, the record count was reduced from 1742411 to 9310 as shown in Figure 19.

Figure 19

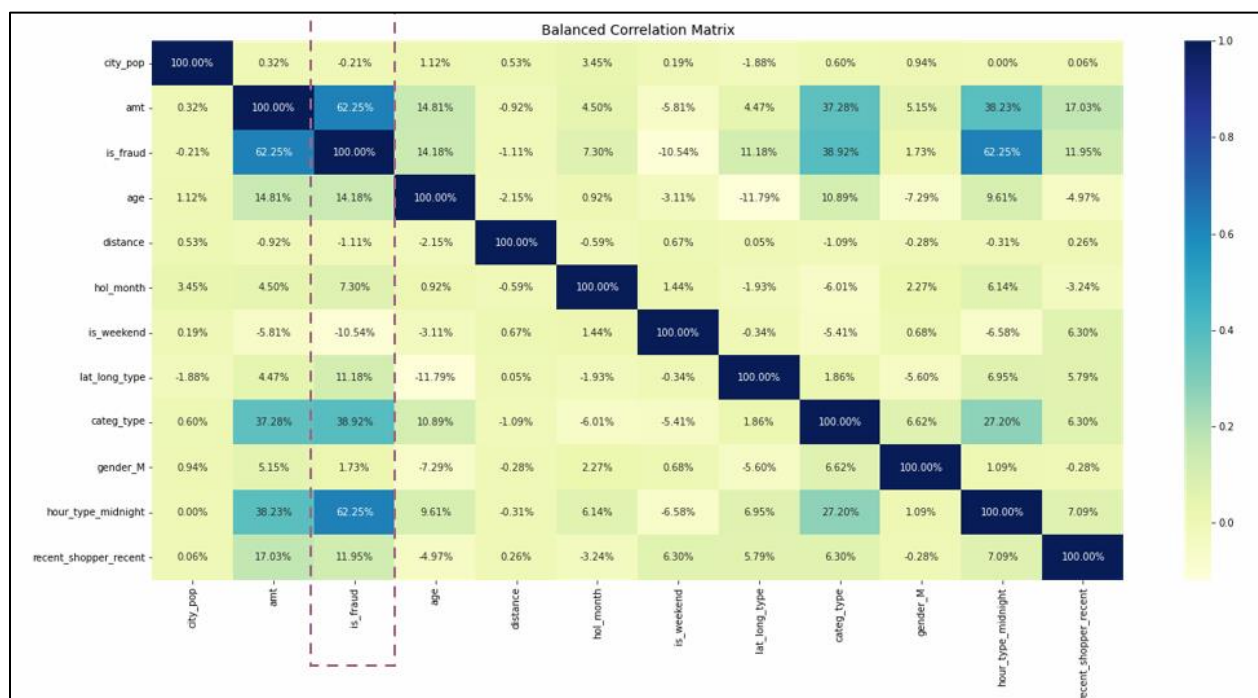
Down sampled data distribution



One can observe from Figure 19 that an equal number of instances of fraud and non-fraud transactions are created. With the balanced dataset, the correlations of the features with the target feature have improved as shown in Figure 20 which in turn will improve the performance of the model.

Figure 20

Correlation matrix of a balanced dataset



3.5 Data Splitting

Once the data is pre-processed and transformed, the data is split into training and testing by shuffling the data. As a preparation for the modeling phase, the data is divided into 80% and 20% for the training set and testing set respectively.

3.6 Data Statistics

3.6.1 Data Cardinality

Based on the count values in Figure 21, we can observe that there are no missing values in the dataset as the count for all the features are similar. There are over 1.7 million transactions in the dataset.

Figure 21

Data Cardinality

	cc_num	zip	lat	long	city_pop	acct_num	unix_time	amt	is_fraud	merch_lat	merch_long
count	1.752721e+06	1.752721e+06	1.752721e+06	1.752721e+06	1.752721e+06	1.752721e+06	1.751721e+06	1.751721e+06	1.751721e+06	1.751721e+06	1.751721e+06
mean	3.843047e+17	5.121929e+04	3.755229e+01	-9.171316e+01	3.154471e+05	4.987205e+11	1.611112e+09	7.051059e+01	5.314773e-03	3.755235e+01	-9.171333e+01
std	1.257718e+18	2.977578e+04	5.158298e+00	1.640430e+01	6.226473e+05	2.859440e+11	1.822508e+07	1.666364e+02	7.270852e-02	5.190740e+00	1.641442e+01
min	6.040616e+10	1.040000e+03	2.132950e+01	-1.593448e+02	1.760000e+02	2.348758e+09	1.577866e+09	1.000000e+00	0.000000e+00	2.032995e+01	-1.603443e+02
25%	1.800159e+14	2.729200e+04	3.387960e+01	-9.862490e+01	1.924000e+04	2.720987e+11	1.595429e+09	9.020000e+00	0.000000e+00	3.383905e+01	-9.865916e+01
50%	3.516698e+15	4.830900e+04	3.861910e+01	-8.662130e+01	6.016300e+04	4.956758e+11	1.609552e+09	4.385000e+01	0.000000e+00	3.848595e+01	-8.652043e+01
75%	4.514627e+15	7.770700e+04	4.150220e+01	-7.953990e+01	2.141120e+05	7.408025e+11	1.627089e+09	8.126000e+01	0.000000e+00	4.150950e+01	-7.944400e+01
max	4.986227e+18	9.950700e+04	6.115350e+01	-6.775340e+01	2.906700e+06	9.993899e+11	1.641024e+09	2.205483e+04	1.000000e+00	6.215252e+01	-6.675340e+01

3.6.2 Data Quality Report

Continuous Fields:

Figure 22 represents the Data Quality Report for the continuous features in the dataset. We observe that the % Miss is 0, as we have handled the missing values in the pre-processing steps. For the features amt, city_pop, and age the difference between third quartile and maximum is very high, indicating that there are outliers. This can be seen in the box plots for the features.

Figure 22*Data Quality Report for Continuous Fields*

Data Quality Report											
Total records: 4											
	Data Type	Count	%Miss	Cardinality	Min	1st Qrt	Mean	Median	3rd Qrt	Max	Std_Dev
city_pop	float64	1751721	0.0	798	176.00	19240.00	315454.76	60163.00	214112.00	2906700.00	622657.87
amt	float64	1751721	0.0	60327	1.00	9.02	70.51	43.85	81.26	22054.83	166.64
age	int64	1751721	0.0	81	15.00	31.00	44.66	43.00	56.00	96.00	17.43
distance	float64	1751721	0.0	75368	0.02	20.53	36.69	36.44	53.03	92.98	19.08

Categorical Fields:

Figure 23 represents the Data Quality Report for the categorical features in the dataset. We observe that the % Miss is 0, as we have handled the missing values in the pre-processing steps. The cardinality for all the features is 2 as we have considered only the required features in the analysis. The mode frequency for is_fraud 0 is very high at 99.46%.

Figure 23*Data Quality Report for Categorical Fields*

Data Quality Report - Categorical Features											
Total records: 9											
	Data Type	Count	%Miss	Cardinality	Mode	Mode Freq	Mode Perc	Second Mode	Second Mode Freq	Second Mode Perc	
gender	object	1751721	0.0	2	M	895503	51.121326	F	856218	48.878674	
is_fraud	object	1751721	0.0	2	0.0	1742411	99.468523	1.0	9310	0.531477	
area	object	1751721	0.0	2	urban	1680714	95.946444	rural	71007	4.053556	
hol_month	object	1751721	0.0	2	2.0	1358082	77.528442	1.0	393639	22.471558	
is_weekend	object	1751721	0.0	2	0	1055304	60.243840	1	696417	39.756160	
hour_type	object	1751721	0.0	2	daylight	1342626	76.646110	midnight	409095	23.353890	
recent_shopper	object	1751721	0.0	2	recent	1218395	69.554170	past	533326	30.445830	
lat_long_type	object	1751721	0.0	3	2	810978	46.296071	1	637355	36.384504	
categ_type	object	1751721	0.0	3	2	833343	47.572816	3	546081	31.173971	

Based on the observations from the data quality reports, the potential handling strategies for the data quality issues are listed in Table 24.

Figure 24

Data Quality Plan

Feature	Data Quality Issue	Potential Handling Strategies
amt	High difference between the third quartile and max value of the feature, indicating outliers	As outliers are important in the anomaly detection, we will retain the instances in the dataset.
city_pop	Outliers with high population values	The values of the city population should be validated. The outliers will not be removed as there is a possibility of high fraudulent transactions in highly populated areas.
is_fraud	Over 99% of the instances are non-fraudulent. This is an imbalanced dataset	Sampling methods to balance the data
age	High difference between the third quartile and max value of the feature, indicating outliers	These transactions will be validated and retained for the model

3.6.3 Analytical Base Table

Figure 25

Analytical Base Table

index	ssn	cc_num	first	last	gender	street	city	state	zip	lat	long	city_pop	job	dob	acct_num	profile	trans_num	trans_date	trans_time	unix_time
0	195-33-0720	3508835615951480	Ellen	Ortiz	F	482 Robert Light Apt. 994	Columbia Falls	ME	4623	44.6699	-67.7534	1054	Optician, dispensing	1997-08-14	917181406434	adults_2550_female_rural.json	5c15602a5c089764ed5d149a78a279a	2021-04-29	00:07:35	1619600055.0
1	195-33-0728	3508835615951480	Ellen	Ortiz	F	482 Robert Light Apt. 994	Columbia Falls	ME	4623	44.6699	-67.7534	1054	Optician, dispensing	1997-08-14	917181406434	adults_2550_female_rural.json	5499f9a7671e25c7a183d2952ccf999	2021-04-29	07:10:46	1619705446.0
2	195-33-0728	3508835615951480	Ellen	Ortiz	F	482 Robert Light Apt. 994	Columbia Falls	ME	4623	44.6699	-67.7534	1054	Optician, dispensing	1997-08-14	917181406434	adults_2550_female_rural.json	844caa31063ba1973a63d3666ca47f30	2021-04-29	10:09:12	1619716152.0
3	195-33-0728	3508835615951480	Ellen	Ortiz	F	482 Robert Light Apt. 994	Columbia Falls	ME	4623	44.6699	-67.7534	1054	Optician, dispensing	1997-08-14	917181406434	adults_2550_female_rural.json	a7f727e5cb10bc1c5da38c5fcca95d7	2021-04-29	09:15:39	1619712939.0
4	195-33-0728	3508835615951480	Ellen	Ortiz	F	482 Robert Light Apt. 994	Columbia Falls	ME	4623	44.6699	-67.7534	1054	Optician, dispensing	1997-08-14	917181406434	adults_2550_female_rural.json	e7594588105594902276aa61b9b0c519	2021-04-29	00:22:42	1619680962.0
5	195-33-0728	3508835615951480	Ellen	Ortiz	F	482 Robert Light Apt. 994	Columbia Falls	ME	4623	44.6699	-67.7534	1054	Optician, dispensing	1997-08-14	917181406434	adults_2550_female_rural.json	05d43309c56e425234ce154d22069d8a	2021-04-29	02:01:46	1619686906.0
6	195-33-0728	3508835615951480	Ellen	Ortiz	F	482 Robert Light Apt. 994	Columbia Falls	ME	4623	44.6699	-67.7534	1054	Optician, dispensing	1997-08-14	917181406434	adults_2550_female_rural.json	c642273f95413149344079795a1a7bc2	2021-04-29	00:09:39	1619680179.0
7	195-33-0728	3508835615951480	Ellen	Ortiz	F	482 Robert Light Apt. 994	Columbia Falls	ME	4623	44.6699	-67.7534	1054	Optician, dispensing	1997-08-14	917181406434	adults_2550_female_rural.json	a91b5047562ebd60be0515d21300c32	2021-04-29	02:28:16	1619688496.0
8	195-33-0728	3508835615951480	Ellen	Ortiz	F	482 Robert Light Apt. 994	Columbia Falls	ME	4623	44.6699	-67.7534	1054	Optician, dispensing	1997-08-14	917181406434	adults_2550_female_rural.json	83aa91d430b5e4965eb5a430c03f13060	2021-04-29	02:59:45	1619690385.0
9	195-33-0728	3508835615951480	Ellen	Ortiz	F	482 Robert Light Apt. 994	Columbia Falls	ME	4623	44.6699	-67.7534	1054	Optician, dispensing	1997-08-14	917181406434	adults_2550_female_rural.json	b1e2b76f6d5ca23d809264a902947249	2021-04-29	01:04:55	1619683495.0
10	195-33-0728	3508835615951480	Ellen	Ortiz	F	482 Robert Light Apt. 994	Columbia Falls	ME	4623	44.6699	-67.7534	1054	Optician, dispensing	1997-08-14	917181406434	adults_2550_female_rural.json	040ecf7c76afed6b0d837df04a3be992	2021-04-29	03:45:32	1619693132.0
11	195-33-0728	3508835615951480	Ellen	Ortiz	F	482 Robert Light Apt. 994	Columbia Falls	ME	4623	44.6699	-67.7534	1054	Optician, dispensing	1997-08-14	917181406434	adults_2550_female_rural.json	13324e0cfac68d418224317e9aacc8b6	2021-04-30	23:41:16	1619851276.0
12	195-33-0728	3508835615951480	Ellen	Ortiz	F	482 Robert Light Apt. 994	Columbia Falls	ME	4623	44.6699	-67.7534	1054	Optician, dispensing	1997-08-14	917181406434	adults_2550_female_rural.json	6423a2e93666cab367899b9ba086c97	2021-04-30	23:01:25	1619848885.0
13	195-33-0728	3508835615951480	Ellen	Ortiz	F	482 Robert Light Apt. 994	Columbia Falls	ME	4623	44.6699	-67.7534	1054	Optician, dispensing	1997-08-14	917181406434	adults_2550_female_rural.json	1a678737f1e0dec526c72d6de4766832d	2021-04-30	20:16:57	1619839017.0
14	195-33-0728	3508835615951480	Ellen	Ortiz	F	482 Robert Light Apt. 994	Columbia Falls	ME	4623	44.6699	-67.7534	1054	Optician, dispensing	1997-08-14	917181406434	adults_2550_female_rural.json	3c38049645bd8625ed57605659301cb6	2021-04-30	12:40:16	1619811616.0
15	195-33-0728	3508835615951480	Ellen	Ortiz	F	482 Robert Light Apt. 994	Columbia Falls	ME	4623	44.6699	-67.7534	1054	Optician, dispensing	1997-08-14	917181406434	adults_2550_female_rural.json	5d4ae7c3a1056652a79f6831a077fb01	2020-07-01	11:02:40	1593626560.0
16	195-33-0728	3508835615951480	Ellen	Ortiz	F	482 Robert Light Apt. 994	Columbia Falls	ME	4623	44.6699	-67.7534	1054	Optician, dispensing	1997-08-14	917181406434	adults_2550_female_rural.json	a255aaf9d3c770a642a055679b6f657d	2020-03-20	07:57:09	1584716229.0

3.7 Data Analytics Results

In the data analytics section, we will analyse the dataset for the descriptive features and target feature. The descriptive features are analyzed separately for categorical and continuous features. In the target feature plot as seen in Figure 26, the proportion of instances with fraudulent transactions is very less as compared to the non-fraudulent transactions.

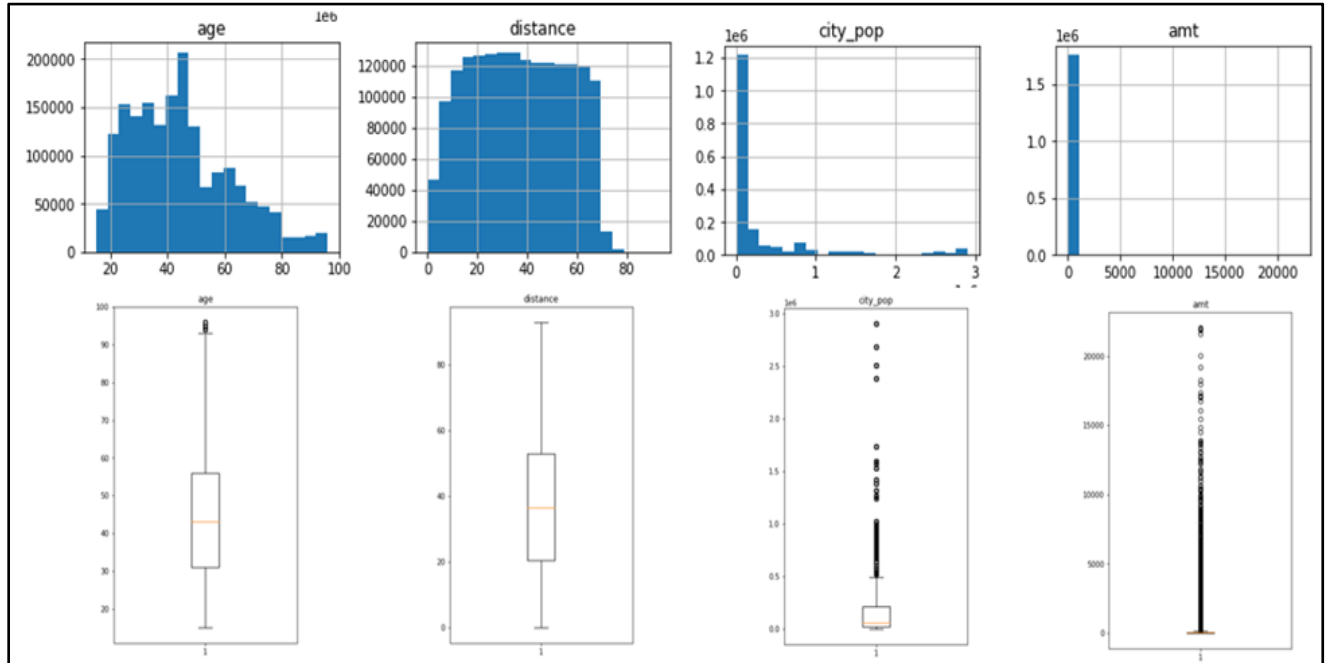
Figure 26

Count of instances by fraud and non-fraud



Figure 27

Histogram and Box Plot for Continuous Features



Small multiples visuals are used to visualize the relationship between two categorical fields. For each feature, the bar plot is plotted for all levels of the target feature is_target, followed by is_fraud = 0, and is_fraud = 1. If there is a strong relationship between the feature compared and target feature, the bar plot for each level of the target feature will be significantly different. It can be observed from Figure 28, features hour_type, recent_shopper, and is_weekend have significantly different plots for the different levels of the target feature, thereby indicating a strong correlation with the target feature. Whereas, gender, area, and hol_month have weak or no correlation with the target feature.

Figure 28

Small Multiples Visualization for Categorical Features

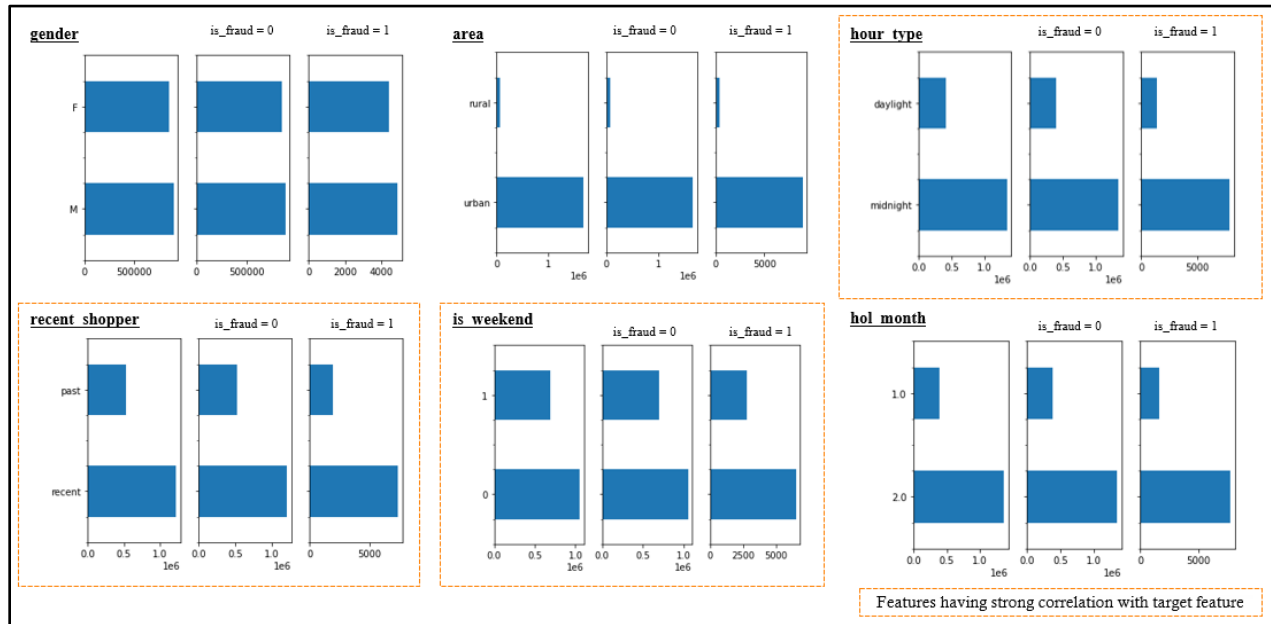
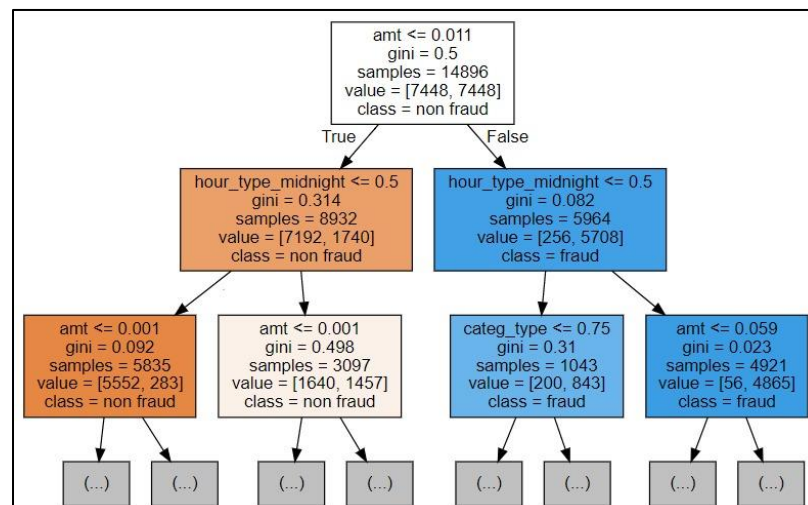


Figure 29 shows the feature importance split which portrays the various features that are effectively used to classify the data into fraud and non-fraud. Gini index is used as the feature to select feature importance. It is observed that the numerical feature amount seems to be the most important with the highest Gini index. The top 3 levels of split have been shown in the figure.

Figure 29

Feature Information



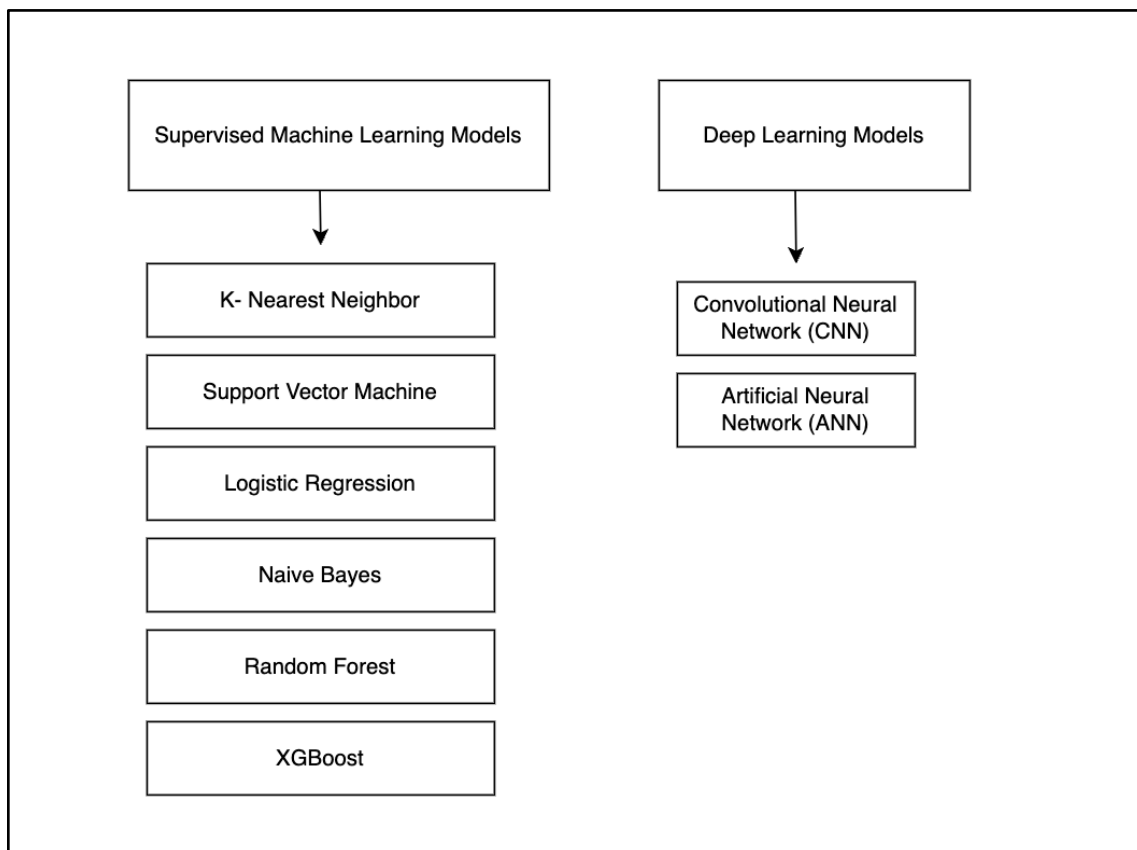
4 Model development

4.1 Model Proposals

Classification algorithms are widely used for detecting fraudulent transactions. For this project, we have employed four state-of-the-art supervised machine learning algorithms such as Naive Bayes , Logistic regression , Random Forest, XGBoost classifier, KNN classifier , Support vector machine (SVM) algorithms for identifying frauds and compared their performances to determine the one with best results. Supervised machine learning models takes in the dataset that has been labeled with the correct output, and use that information to make predictions on new data Along with machine learning models , some of the deep learning models like Convolutional Neural Network (CNN) and Artificial Neural Network (ANN) were also experimented and evaluated based on their performance. The models used can be seen from Figure 30.

Figure 30

Models used



4.1.1 Naive Bayes

Naïve Bayes classifier is a probabilistic machine learning model widely used for classification and predictive modeling. Probabilistic classifiers are used to predict multiple classes and the decision is based on conditional probability. It is a collection of classification algorithms based on Bayes theorem.

Figure 31

Bayes Theorem

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Where,

$P(A)$ – Probability of hypothesis A being true regardless of data based on its prior probability

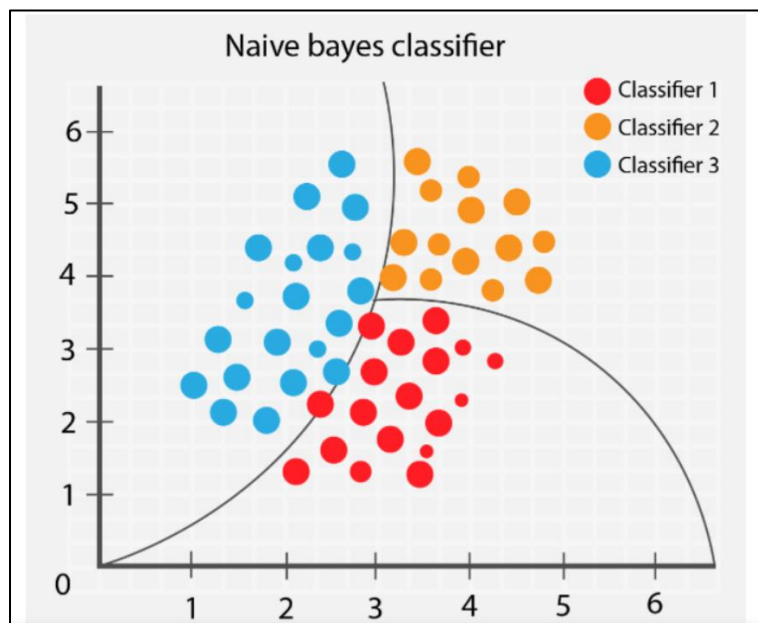
$P(B)$ – Probability of the data B

$P(A|B)$ – Probability of hypothesis A given data B – posterior probability

$P(B|A)$ – Probability of data B given hypothesis A being true

Figure 32

Naive Bayes Algorithm



Source:

<https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.analyticsvidhya.com%2Fblog%2F2022>

%2F03%2Fbuilding-naive-bayes-classifier-from-scratch-to-perform-sentiment-analysis%2F&psig=AOvVaw2FbOySLrwrCiYKcJWfGPur&ust=1670813250350000&source=images&cd=vfe&ved=0CA8QjRxqFwoTCLCssMTG8PsCFQAAAAAdAAAAABAE

Bayes theorem works on conditional probability, which gives the probability of some event happening based on some event already happened. It gives the probability of an event occurring based on its past knowledge. It is a family of algorithms where all of them share a common principle. The assumption here is that the features are independent of each other, and their presences will not affect the other, hence called naïve. That means that all the predictors have equal effect on the outcome.

There are different types of naïve bayes algorithms like Gaussian Naïve bayes where if the features have continuous values, then assumption is that the values in each class follow gaussian distribution, Multinomial Naïve bayes which is mainly used on data which follows multinomial distribution. It is widely used in Natural language processing (NLP) for text classification where each event constitutes to the presence of the word and Bernoulli naïve bayes is used where data is distributed according to multivariate Bernoulli distribution which means there can be multiple features existing, but it is assumed that each one contains a binary value.

Naive Bayes algorithm is widely used for multiclass classification applications for both categorical and numerical datasets. It is a highly extensible algorithm which learns very fast and can be used to make predictions in real time. It is popularly used for sentimental analysis, spam email filtering and for building recommendation engines. It works well with smaller as well as larger datasets. However, the major disadvantage of this classifier is that it considers that all the variables are independent in nature which will not be the situation in real time as most of the time the variables will be dependent on each other which hinders the performance of the classifier. Additionally, the model can struggle with data sets that have a small number of sample or a large number of features.

4.1.2 Logistic regression

Logistic regression is a supervised machine learning algorithm which can be used to solve classification problems in the real world. This algorithm can be used to classify data into categories or classes by predicting the probability that a data point falls into a particular class

based on the features. It is often used for binary classification, though it can be extended to more than 2 classes. Models are trained on historical labeled datasets and aim to predict which category new observations will belong to.

In the project, the goal of the algorithm is to predict the incoming transaction as fraudulent or non-fraudulent based on the history of transactions. It acts as a binary logistic classifier.

Logistic regression makes use of Logistic function / sigmoid function to classify the target variable y as one of the two categories (0 or 1). The function takes any value and converts it to a number between 0 and 1. Here, Sigmoid function is a machine learning activation function that is used to introduce non-linearity in the mode. For the given inputs $X = \{ x_1, x_2, x_3, \dots, x_n \}$, y belongs to a particular category Fraud, Non-Fraud is a sigmoid function which is given by

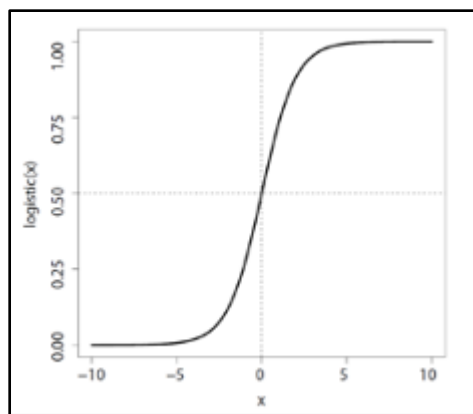
$$P(y = 1|X) = \text{sigmoid}(z) = \frac{1}{1 + e^{-z}} \quad \text{Where } z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n$$

Where Z is the linear predictor which is transformed by the sigmoid function so that the values fall between 0 and 1 and therefore can be understood as probabilities. The resulting probability can be used to predict the target class for ‘ y ’ based on the input features ‘ X ’.

When we plot the above equation, we get the S shaped curve as shown in Figure 33.

Figure 33

Sigmoid function for logistic regression



The most important takeaway from the preceding graph is that the output along the vertical axis will always fall between 0 and 1 regardless of the value of x we use in the logistic or sigmoid function.

Logistic function can be used to predict the probabilities of each outcome which in turn can predict the class fraud/non-fraud. We use a classification threshold, or decision boundary, to decide the

predicted class based on the probability of each class given the feature values. A typical threshold is 0.5. When the result of the sigmoid function is greater than 0.5, we classify the label as class 1 or positive class or fraudulent transaction; if it's less than 0.5, we can classify it as a negative class or 0 or non-fraudulent transaction. This threshold can be adjusted.

4.1.3 Ensemble learning

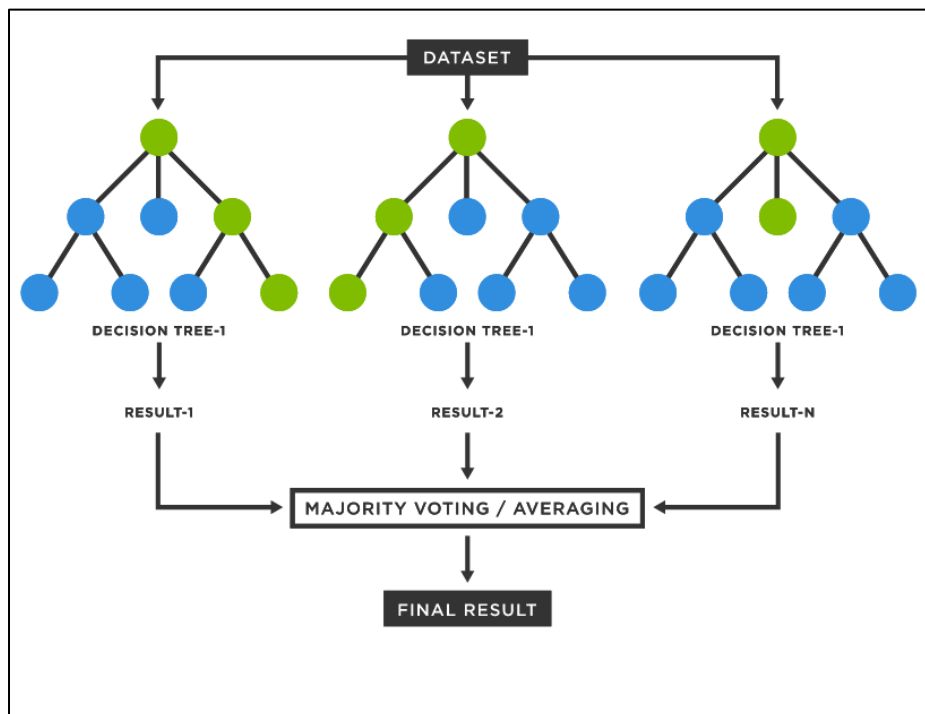
Random forest and XGBoost models of Ensemble learning are leveraged for the credit card classification.

Random Forest

Random Forest is a bagging classifier which aggregates the input from various decision trees. Various parameters like number of trees required, number of samples to be considered for each tree, number of features required for each tree, which mechanism needs to be used for tree splitting, are the various hyperparameters required for tuning and arriving at an optimal modeling parameters. Random Forests are known to decrease overfitting as they combine the output of various individual decision trees.

Figure 34

Random Forest Classifier



Source: <https://www.tibco.com/de/reference-center/what-is-a-random-forest>

Finally, the output from each of the decision trees is combined by taking a voting from the majority of the trees.

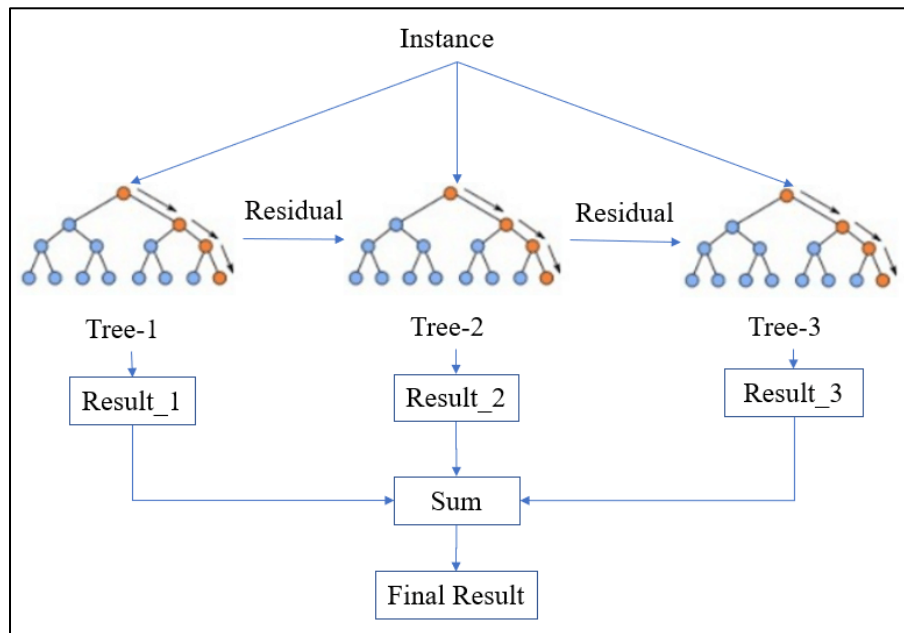
XGBoost

XGBoost is a boosting classifier which uses gradient boosting technique. It is a sequential modeling technique which takes the output of one model and feeds it to another model. The results that aren't classified correctly are given more weightage when being fed to every consequent model. This removes the faulty classification from the individual models. For the project purpose, decision trees are considered as base models whose outputs are fed sequentially to the upcoming models.

Figure 35 shows a pictorial representation of the XGBoost model with decision trees being the base classifiers. Various parameters like number of trees required, number of samples to be considered for each tree, number of features required for each tree, which mechanism needs to be used for tree splitting, are the various hyperparameters required for tuning and arriving at an optimal modeling parameters.

Figure 35

XGBoost Classifier



Source: https://www.researchgate.net/figure/Simplified-structure-of-XGBoost_fig2_348025909

4.1.4 KNN

K Nearest Neighbor (k-NN) is a similarity-based algorithm that is extensively used in detection systems. The prediction in this algorithm is based on the majority value of the target feature of the nearest neighbors. The advantage of the k-NN model is the simplicity of the training phase to build the model and requires all the training instances to be stored in the memory. To make the predictions for a query instance, the model computes the distance in the feature space between each instance in the memory and the query instance. Based on the value of k, the majority of the target levels of k nearest neighbours to the query instance is returned by the algorithm. There are three distance metrics used to compute the distance.

$$Euclidean(\mathbf{a}, \mathbf{b}) = \sqrt{\sum_{i=1}^m (\mathbf{a}[i] - \mathbf{b}[i])^2}$$

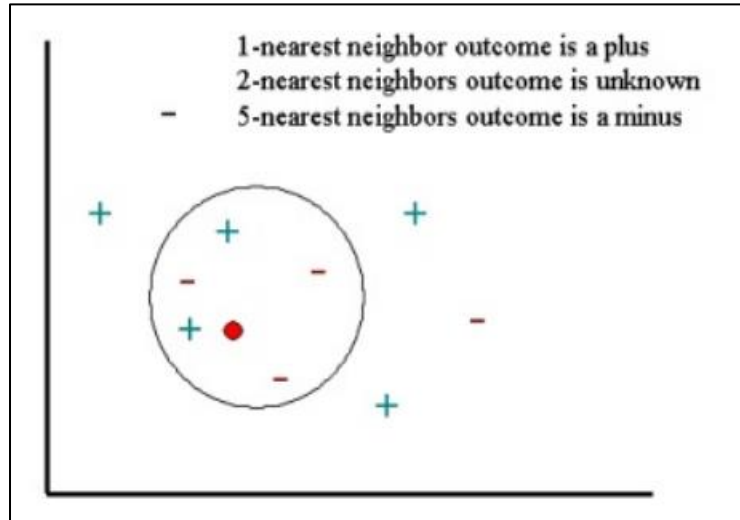
$$Manhattan(\mathbf{a}, \mathbf{b}) = \sum_{i=1}^m abs(\mathbf{a}[i] - \mathbf{b}[i])$$

$$Euclidean(\mathbf{a}, \mathbf{b}) = \left(\sum_{i=1}^m abs(\mathbf{a}[i] - \mathbf{b}[i])^p \right)^{1/p}$$

Euclidean distance is the default distance metric used in the k nearest neighbor model. The distance of the query instance is computed for all the instances in the dataset. For a k value as 5 as shown in Figure 36, the outcome predicted is minus based on the majority labels in the five nearby neighbors. We have experimented with different values of k and assessed the performance of the model for them.

Figure 36

KNN Model



Note. KNN model adapted from Analysis on Credit Card Fraud Identification Techniques based on KNN and Outlier Detection by N.Malini and Dr.M.Pushpa (2017)

For the k nearest neighbor based on the distance calculation, the majority target level is used for the prediction of the target level.

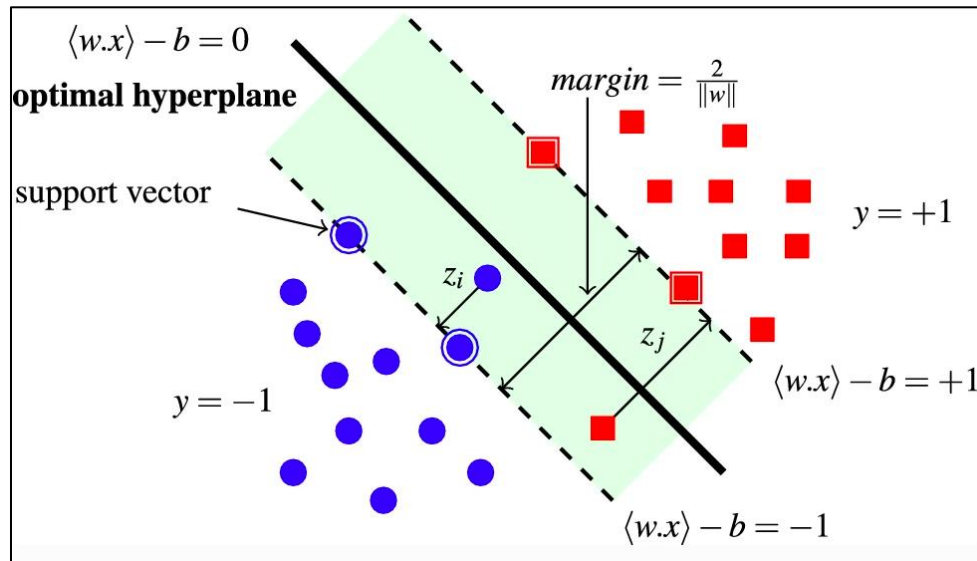
$$\mathbb{M}_k(\mathbf{q}) = \arg \max_{l \in \text{levels}(t)} \sum_{i=1}^k \delta(t_i, l)$$

4.1.5 SVM

Support Vector Machines is a supervised machine learning algorithm which is used for both classification and regression tasks. However, it is popularly used for classification tasks. Higher speed and greater performance with fewer samples are their two key benefits (in the thousands).

In SVM we perform the classification task by finding the correct hyperplane which can classify/differentiate two classes appropriately as shown in Figure 37.

Figure 37
SVM model



Source: Do, T.-N. (2019, June 25). Automatic learning algorithms for local support vector machines - SN computer science. SpringerLink. Retrieved December 10, 2022, from <https://link.springer.com/article/10.1007/s42979-019-0006-z/figures/1>

The hyperplane is a line that divides the input space into two regions, one for each class. The position of the hyperplane is determined by the support vectors, which are the data points closest to the line. These support vectors "support" the hyperplane by defining the maximum margins between the two classes. The optimal hyperplane is the one that has the largest margin between the two classes. In the case of an SVM, the equation for the hyperplane is given by $wx + b = 0$, where w is a vector that defines the orientation of the hyperplane and b is the bias term. The bias term determines the position of the hyperplane on the y -axis.

In a more general form, the equation for an SVM hyperplane in a d -dimensional space is given by $w_1x_1 + w_2x_2 + \dots + w_dx_n + b = 0$, where x_1, x_2, \dots, x_d are the descriptive features of the data and w_1, w_2, \dots, w_d are the coefficients that determine the orientation of the hyperplane.

In this context, an SVM would be trained on a dataset of historical credit card transactions, where each transaction is represented by a set of features such as the transaction amount, the time of the transaction, the location of the transaction, and so on. The goal of the SVM is to learn the patterns in the data that distinguish fraudulent transactions from normal transactions.

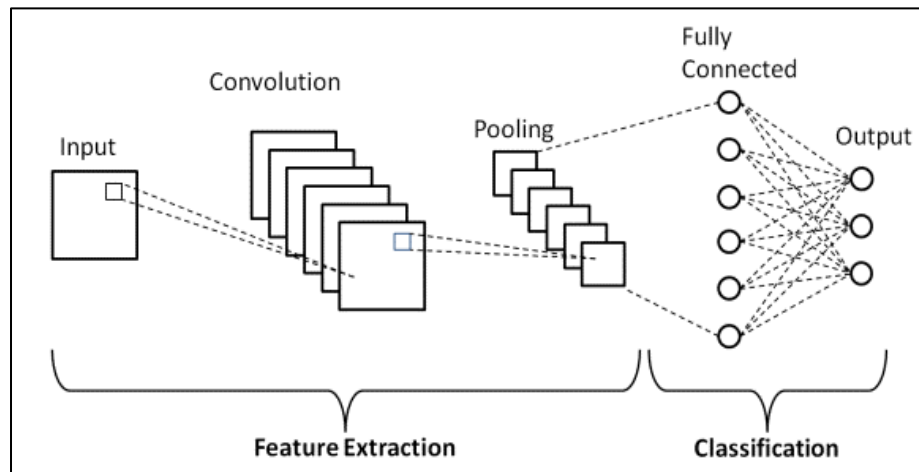
Once the SVM is trained, it can be used to classify new transactions as either normal or fraudulent. For instance, if a new transaction has similar features to transactions that were previously labeled as fraudulent, the SVM would classify it as fraudulent. On the other hand, if a new transaction has similar features to transactions that were labeled as normal, the SVM would classify it as normal.

In general, SVMs are considered to be effective for credit card fraud detection because they can handle high-dimensional data and can find complex, non-linear patterns in the data. Additionally, SVMs can provide good performance even with small datasets, which is often the case in fraud detection.

4.1.6 Convolutional Neural Network (CNN)

Convolutional neural network is a type of deep learning model that is commonly used in image recognition and classification tasks. The model consists of multiple layers, including convolutional layers, pooling layers, and fully connected layers which can be seen from Figure 38. The convolutional layers apply a set of learnable filters to the input data to extract features and are used to learn hierarchical representations of the input data. The output of the convolutional layers is a set of feature maps, which represent the input data at different scales and with different levels of abstraction. The pooling layers then down sample the feature maps to reduce the dimensionality of the data and make the model more computationally efficient. This is done by applying a pooling operation such as max pooling, which selects the maximum value from each region of the feature map. The fully connected layers then combine the extracted learned by the convolutional and pooling layers and make predictions based on those features. This is done by applying a series of linear transformations to the input data, followed by a non-linear activation function such as a sigmoid or ReLu.

Figure 38
CNN Architecture



Source: <https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.upgrad.com%2Fblog%2Fbasic-cnn-architecture%2F&psig=AOvVaw28q28UM3AW4cAJPXXfTQ3K&ust=1670813300803000&source=images&cd=vfe&ved=0CA8QjRxqFwoTCIDWidzG8PsCFQAAAAAdAAAAABAE>

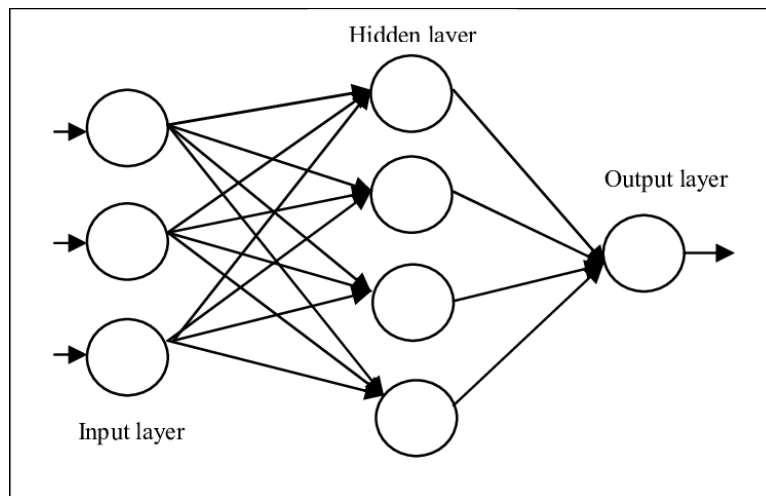
The advantage of using a CNN is that it can automatically learn to extract useful features from the input data. They are efficient to train and use, making them suitable for real-time applications. However, they require a large amount of training data to perform well and can be sensitive to the initial values of the weights in the network. Despite being a powerful tool widely used in many machine learning tasks, the model can sometimes be difficult to interpret, making it hard to understand why the model is making a particular prediction.

4.1.8 Artificial Neural Network (ANN)

Artificial neural network is a type of machine learning model designed to mimic the structure and function of the human brain. It is a computational model consisting of many interconnected processing units called neurons which work together to solve complex problems. The basic ANN architecture is the single-layer perceptron, which consists of a single layer of neurons, each of which receives input from multiple sources and produces a single output and can be seen from Figure 39. Multi-layer perceptron consists of multiple layers of neurons with each layer receiving input from the previous layer and producing output that is then passed on to the next layer. These networks are trained to recognize complex patterns in the data and most used for tasks like image and speech recognition.

Figure 39

ANN Architecture



Source:

https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.researchgate.net%2Ffigure%2FANN-Architecture-Several-ANN-architectures-have-been-developed-such-as-feedforward-NN_fig3_261206036&psig=AOvVaw1y-E8tIqliJjR5RU8N8Gze&ust=1670813622234000&source=images&cd=vfe&ved=0CA8QjRxqFwoTCNjdmPXH8PsCFQAAAAAdAAAAABAL

One of the strengths of ANNs is their ability to learn and adapt based on the data they are given. During training, an ANN can adjust the strength of the connections between its neurons in order to better recognize patterns and make more accurate predictions. This ability is particularly well suited for image and speech recognition tasks where the data can be very complex and varied. However, one of the biggest challenges with using ANN is the need for a large amount of labeled data for the model to train effectively. Additionally, the model can be computationally intensive making it difficult to use for real-time applications which have limited power and resources. ANNs are opaque in their decision making, so it is sometimes difficult to understand how and why the model arrived at a particular prediction.

4.2 Model Comparison

Figure 40

Model Comparison

Characteristic	Logistic Regression	SVM	Ensemble	KNN	Naive Bayes	CNN	ANN
Type	Error based learning	Error based learning	Ensemble learning	Similarity-based learning	Probability based learning	Recognises different patterns and features using multiple layers of interconnected nodes.	Recognises different patterns and features using multiple layers of interconnected nodes.
Preferred dataset	Small	Small with more features	Large	Small	Small with less features	Large	Large
Decision boundary	Linear	Non-linear	Non-linear	Linear	Linear	Non-linear	Non-linear
Memory efficiency	Less memory efficient but can be improved	SVM is memory efficient (use only a subset of training data in the decision phase)	Computational efficiency might be an issue	Memory intensive	Memory efficient (requires only small amount of training data to make predictions)	Not memory efficient (requires large amount of data)	Not memory efficient (requires large amount of data)
Advantages	Perfect for linear separable datasets	Ability to detect fraud at the time of transaction	Automatic variable selection. Suitable for wide data and large datasets	Predictive model before classification is not required	Requires less training data	Can learn complex, hierarchical patterns in data	Ability to detect fraud at the time of transaction
Disadvantages	Does not work well for highly correlated data. (Multicollinearity)	SVM is sensitive to the type of kernel it uses and does not perform well when target classes are overlapping	Biased for features with more levels Slower process if more estimators exists	Cannot detect fraud at the time of fraud	Not suitable for highly correlated features	Can be sensitive to choice of hyperparameters and require careful training.	Prone to overfitting

4.3 Model Evaluation Methods

For evaluation of the performance of the model, the data is split for training and testing. In general, for machine learning models, more data is needed for training the model so keeping that in mind, the data is split in a way that 80% is used for training and 20% is used for testing. Though the dataset we have considered is balanced, steps have been taken to avoid unnecessary bias when splitting the data. To evaluate the performance of the models in detecting fraudulent transactions, evaluation metrics like accuracy, precision, recall and f1 score are employed.

Accuracy refers to the percentage of predictions made by the model that are correct. For fraud detection, high accuracy means that the model is correctly identifying most fraudulent transactions.

Precision on the other hand, measures the proportion of positive predictions that are actually fraudulent. High precision model means that the model is not flagging many non-fraudulent transactions as fraudulent which is very important in the case of fraud detection.

Recall also known as sensitivity measures the proportion of actual fraud transactions that are correctly identified by the model where high recall means the model is effective on identifying most fraudulent transactions.

The F1 score is a combination of precision and recall calculated as the harmonic mean of the two. It is very useful for comparing the performance of different models as it considers both precision and recall.

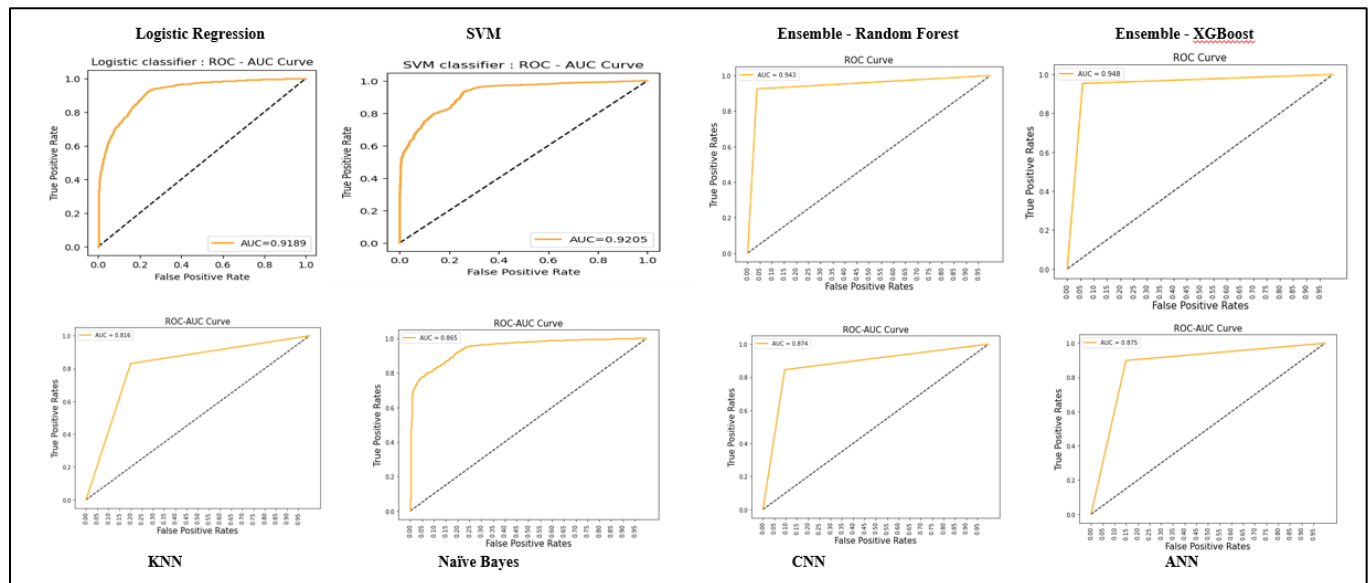
In addition to calculating above mentioned metrics, confusion matrix is another popular tool used to evaluate the performance of the machine learning models. It helps to visualize the model's performance by comparing the predicted values with the actual values. In case of credit card fraud detection, the matrix can be used to evaluate the ability of the machine learning model to accurately identify fraudulent transactions.

4.4 Model Validation and Evaluation Results

Model validation and evaluation is an important step in the development of machine learning models used for fraud detection. In order to evaluate the performance of the models, data is split into training and testing sets and the models are trained on the training set. The performance of the models is then evaluated on the testing set using the aforementioned metrics. The results of the evaluation help to identify the most effective model for credit card fraud detection and can be seen from Figure 41.

Figure 41*Model Evaluation Results*

Model Name	Accuracy	Precision	Recall	F1 score	AUC
KNN	0.815	0.8061	0.8308	0.8183	0.816
Logistic	0.86	0.8701	0.8528	0.8614	0.9189
Random Forest	0.94	0.96	0.92	0.94	0.943
XGBoost	0.95	0.94	0.95	0.95	0.948
SVM	0.82	0.7976	0.855	0.8253	0.9205
Naïve Bayes	0.865	0.886	0.835	0.8613	0.865
CNN	0.88	0.898	0.858	0.878	0.88
ANN	0.874	0.8575	0.877	0.899	0.8748

Figure 42*ROC-AUC Curves for Classifiers*

In summary, XGBoost and random forest have been found to be most effective machine learning models for fraud detection while KNN has been found to perform the least. On the other

hand, when it comes to deep learning models employed for the task of detecting fraudulent transactions, CNN performed better than ANN.

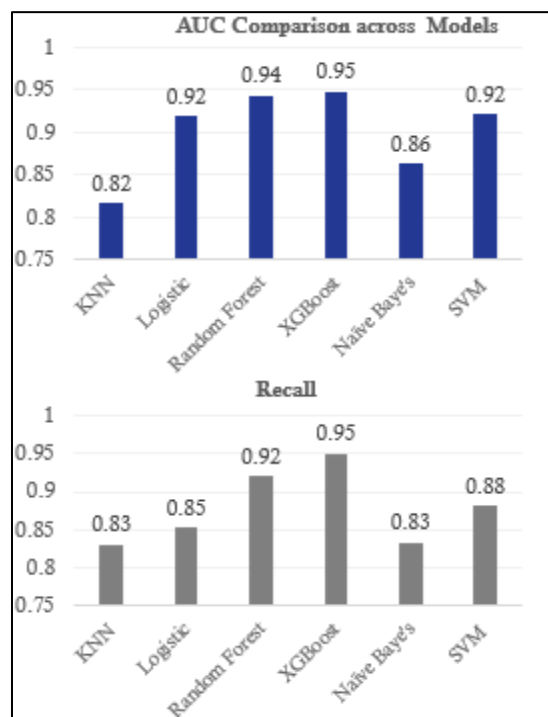
4.5 Model Results Discussion

On observing the results of all the models' performance, it is evident that the ensemble models like Random Forest and XGBoost have outperformed the others in terms of all the classification metrics with F1 score being, precision being, recall being, accuracy being. This is majorly because the ensemble model combines the output of many weak learners and aggregate the results.

Recall is considered as the metric under consideration as it mentions the percentage of positives that are correctly classified. This shows that ensemble learning models can be used for this purpose based on the observations. Figure 43 shows the comparison of AUC for various selected models.

Figure 43

AUC and Recall for Models



5 System Design and Architecture

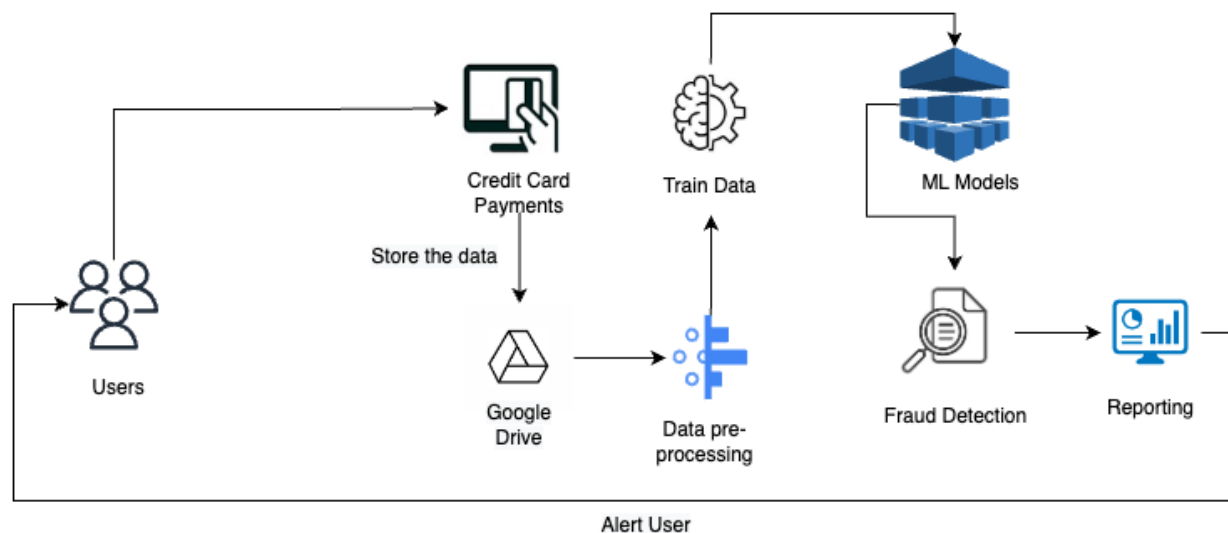
5.1 System Design & Architecture

Advancement in science and technology has increased a lot in recent years. With the development in e-commerce, tap and pay systems, and e-payment methods there has been a tremendous increase in financial frauds. Credit card usage has increased tremendously over the decade which revolutionized the cashless payment methods, but it comes with its own set of risks. During online transactions, the credit card details are collected by cyber criminals and hackers and are used for fraud transactions. To tackle this issue, Machine Learning models Can be employed by financial institutions or banking systems to identify fraudulent transactions and take necessary actions.

In this project, an end-to-end process flow has been developed for detecting credit card fraudulent transactions which can be seen from Figure 44.

Figure 44

Process Flow Architecture



In an ideal scenario, when a user performs any transaction online, the details of the transaction like the date and time during which the transaction took place, what is the location where the transaction happened, the credit card details using which the transaction is made, the card holder name etc., are collected. With the increase in e-commerce systems, huge volumes of transactions

take place every day so companies or financial organizations might leverage cloud services to store the data. As the transaction details discussed above are sensitive information, it might not be possible to collect real time data. In this project, for developing a machine learning model to help identify fraudulent transactions, synthetic data is generated keeping in mind all the necessary parameters so that the data generated closely replicates the real time transactions and can be used for training and testing the models. The data generated is of smaller size and is stored in google drive.

The data is then preprocessed to make it suitable for modeling. Pre-processing steps like removing of null records, duplicates, adding and removing of columns, changing the column names and adding new features required for the model. The preprocessed data is then split for training and testing which are then fed to the models. Using training data, the machine learning models identify the patterns and trends in which the frauds are occurring which helps them identify the frauds when new data/ test data is fed to the model in real time. In this project, various state-of-the-art techniques are benchmarked, and their performance is evaluated using some of the metrics like accuracy, precision, recall and f1 score by comparing the results, the model with best performance is identified and deployed. Once the fraud is detected by using different reporting methods, the financial organizations or users are alerted.

5.2 System Supporting Environment

Below are the system level environments needed for the implementation of this project.

- Operating System: MAC OS Monterey version 12.6 / Windows 11 PC
- Programming languages: Python 3.9
- Machine learning Libraries: sci-kit-learn
- Python Libraries:
 - Pandas: data frame manipulation
 - Numpy: mathematical manipulation
- Storage: Local
- Visualization Tool: Power BI

6. System Evaluation and Visualization

6.1 Analysis of Model Execution and Evaluation Results

The machine learning models that were implemented were KNN, Logistic Regression, Naive Bayes, Random Forest, XGBoost. Deep learning models like CNN and ANN were implemented. The data was heavily imbalanced with less than percent of fraudulent transactions. This raised the need for balancing the data and also crucially selecting the classification metrics that are needed. The classification metrics required is recall. Further, metrics like average class accuracy which considers the average of recall of both classes.

Performance results of Grid Search Cross Validation has proved that the models have performed well and there is no lucky split that has occurred.

6.2 Achievements and Constraints

Achievements

One achievement is the improved accuracy in detecting fraud transactions. By using this prediction model, credit card companies and financial institutions will be able to more accurately identify fraudulent transactions and prevent them from happening. This will help reduce the financial losses associated with credit card fraud and improve overall security of the credit card system.

Constraints

One of the constraints is the quality of the data. In order for a machine learning model to be effective, it must be trained on a large and diverse dataset that accurately represents the underlying distribution of the data. In this project, the dataset was quite imbalanced where the non-fraudulent transactions outnumbered the fraudulent transactions. The dataset was downsampled which in turn reduced the size of the overall dataset. As the model does not accurately represent the types of transactions that the model will encounter in the real world, the model's performance may be poor.

6.3 System Quality Evaluation of Model Functions and Performance

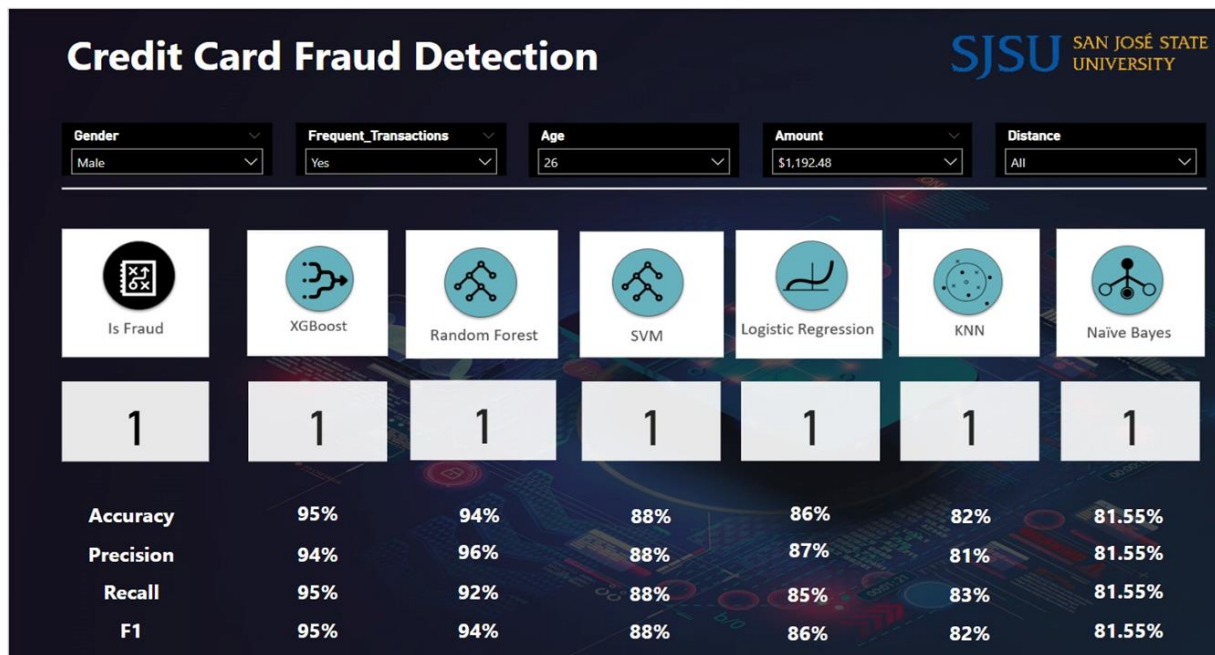
To evaluate the performance of the model after the deployment, an on-going model validation scheme will be utilized to monitor the model performance and identify when there is a significant deviation in the model performance. This is done by monitoring the changes in the model performance. We have used various model performance metrics. Of those, accuracy can be considered as a model performance metric to check the deviation. If the signal indicates a concept drift, corrective actions can be planned and implemented. The correct target feature of `is_fraud` will be identified and updated.

6.4 System Visualizations

Following dashboard is useful to predict the fraud occurrence by providing the details in the query. Various parameters can be set up in the query instance such as gender, age, amount of transaction, distance between the customer location and merchant location, indicator if the customer has made frequent transactions, and so on. Based on the query instance, the output from the models deployed in this project can be seen. As the best performing model in this case is XGBoost, the output predicted by this model can be considered.

Figure 45

Dashboard for Detecting Fraud using Query Instance



7 Evaluation and Reflection

7.1 Benefits and Shortcomings

The models used in this project can be used by the credit card companies to provide proactive alerts for the fraudulent transactions. Such transactions can be alerted to the customers instantly and corrective actions can be taken. Although these models take into consideration multiple factors, there are other useful parameters that can be incorporated in the model. Some of these can be valid authentication at the time of payment, identity validation based on the backend verification process, social security number, and credit score details. Due to the confidentiality issues of the financial transactions data, a data generator is used to create the synthetic data. The performance of the model can be improved and made more relevant if we have real transaction data to build the model.

7.2 Experience and Lessons Learned

In the implementation of this project, we have explored various machine learning techniques that can be used for the credit card fraud detection application. Following are the key takeaways from the project:

- **Data collection technique:** A unique technique to generate the synthetic data for the credit card transactions was explored. In Spite of generating synthetic data, we could achieve good performance for the models implemented in this project.
- **Handling data quality issues:** The data quality report generated for both categorical and continuous features is a useful tool to understand the descriptive feature's characteristics and handle data quality issues before the model development. The problematic features in the dataset could be identified and excluded.
- **Data encoding:** Categorical features have been label encoded in the case of suspicious location fields and one hot encoded in the case of other generic features like hour type, weekend, etc.
- **Sampling method:** Various sampling methods were tested. Selection of the sampling method depends on the type of dataset in the model. As the original dataset was heavily imbalanced, a sampling technique to create balanced target levels was selected. Down

sampling method on the dataset was selected as we observed better correlations of the descriptive features with the fraud indicator.

- **Model evaluation scores:** The evaluation metrics used here are accuracy, precision, recall and f1. However, recall is the ideal metric for evaluation. In the case of credit card fraud detection, False Negatives (FN) can be dangerous as they can result in consequences. Our classification thresholds should be set to optimize recall over other metrics.

7.3 Recommendations for Future Work

As a part of future work, the number of features available could be increased in order to make the output of model predictions more intuitive and accurate and so as to ensure that the model is precisely able to predict fraudulent transactions. Examples of few additional features include adding a field to denote a first-time shopper, more factors on segmentation of customers, etc. The model can be integrated in the AWS environment which will scale the process and further ensure that the system could be served as a package for any company that plans to use it as a product for fraud transaction identification. Also, in future a customer segmentation strategy could be put in place for clusters of customers. Those genuine customers who lie in the same cluster as the fraudsters need to be on watch. Unsupervised machine learning techniques could be put in place in order to formulate customer clusters.

7.4 Contributions and Impacts on Society

The use of machine models for credit card fraud detection has several positive impacts on society. One of the main benefits is that it helps to reduce the amount of credit card fraud that occurs. Machine learning algorithms analyze patterns in the transactions data so that financial institutions can quickly identify suspicious activity and take steps to prevent fraud before it happens. This helps in saving significant amounts of money for individuals and businesses along with protecting their personal information. As the demand for these algorithms increases, there is need for skilled professionals to develop, implement, and maintain these systems. By automating the process of detecting fraud using machine learning models, financial institutions can reduce the number of false positives and minimize the losses.

References

- A. Agrawal, S. Kumar and A. K. Mishra, "Credit Card Fraud Detection: A case study," 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom), 2015, pp. 5-7.
- D. Tanouz, R. R. Subramanian, D. Eswar, G. V. P. Reddy, A. R. Kumar and C. V. N. M. Praneeth, "Credit Card Fraud Detection Using Machine Learning," 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS), 2021, pp. 967-972, doi: 10.1109/ICICCS51141.2021.9432308.
- Malini, N., and M. Pushpa. "Analysis on Credit Card Fraud Identification Techniques Based on KNN and Outlier Detection." *2017 Third International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB)*, 2017, <https://doi.org/10.1109/aeeicb.2017.7972424>.
- F. K. Alarfaj, I. Malik, H. U. Khan, N. Almusallam, M. Ramzan and M. Ahmed, "Credit Card Fraud Detection Using State-of-the-Art Machine Learning and Deep Learning Algorithms," in *IEEE Access*, vol. 10, pp. 39700-39715, 2022, doi: 10.1109/ACCESS.2022.3166891.
- P. Singh, V. Chauhan, S. Singh, P. Agarwal and S. Agrawal, "Model for Credit Card Fraud Detection using Machine Learning Algorithm," 2021 International Conference on Technological Advancements and Innovations (ICTAI), 2021, pp. 15-19, doi: 10.1109/ICTAI53825.2021.9673381.
- S. Dhankhad, E. Mohammed and B. Far, "Supervised Machine Learning Algorithms for Credit Card Fraudulent Transaction Detection: A Comparative Study," 2018 IEEE International Conference on Information Reuse and Integration (IRI), 2018, pp. 122-125, doi: 10.1109/IRI.2018.00025.
- S. Khatri, A. Arora and A. P. Agrawal, "Supervised Machine Learning Algorithms for Credit Card Fraud Detection: A Comparison," 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 2020, pp. 680-683, doi: 10.1109/Confluence47617.2020.9057851.
- S. K. Pradhan, N. V. Krishna Rao, N. M. Deepika, P. Harish, M. P. Kumar and P. S. Kumar, "Credit Card Fraud Detection Using Artificial Neural Networks and Random Forest Algorithms," 2021 5th International Conference on Electronics, Communication and

Aerospace Technology (ICECA), 2021, pp. 1471-1476, doi:
10.1109/ICECA52323.2021.9676142.

Do, T.-N. (2019, June 25). *Automatic learning algorithms for local support vector machines - SN computer science*. SpringerLink. Retrieved December 10, 2022, from <https://link.springer.com/article/10.1007/s42979-019-0006-z/figures/1>