

Project Report Format

Introduction:

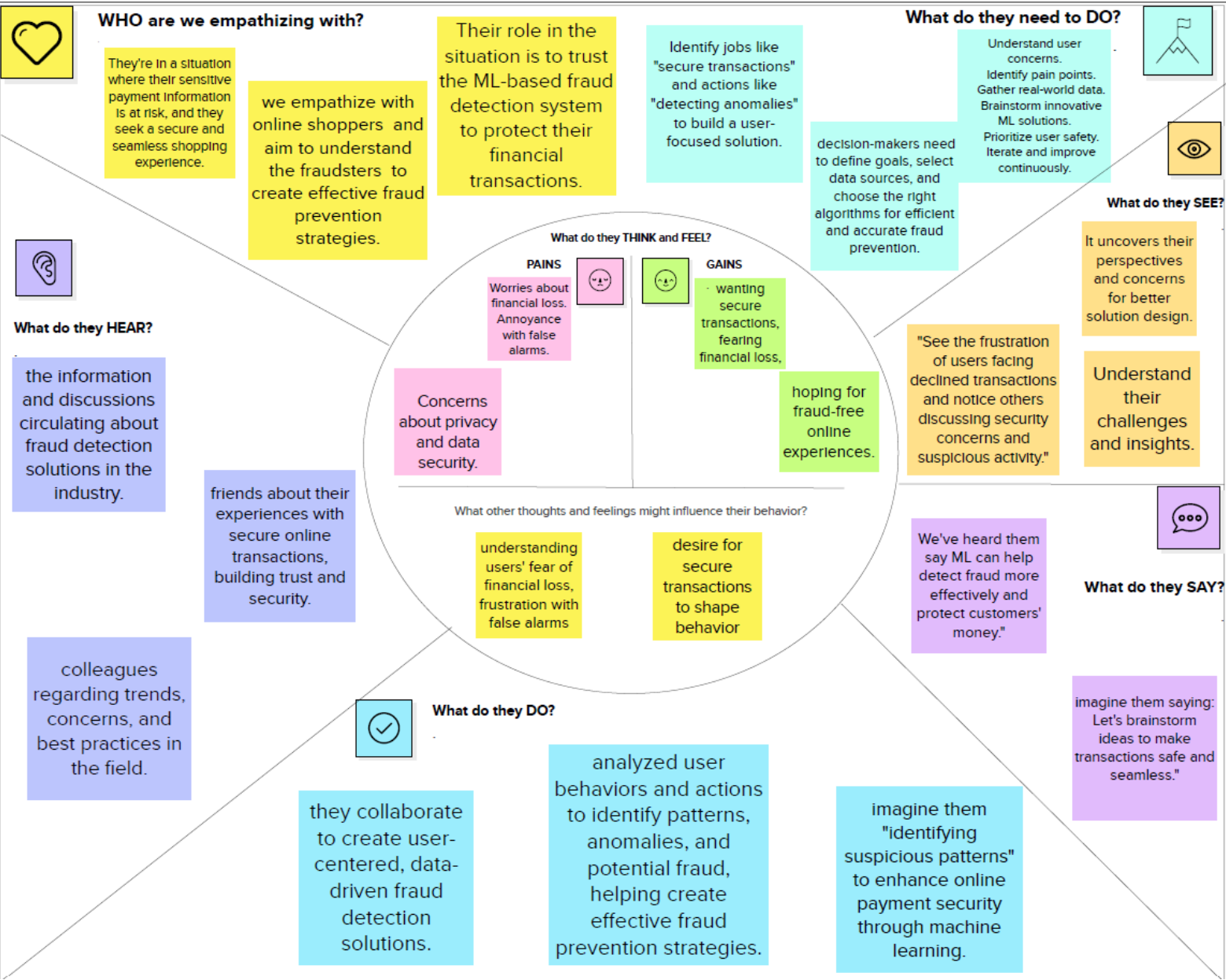
Project Overview: This project focuses on developing an advanced online payments fraud detection system using machine learning techniques. Leveraging a comprehensive dataset of online transactions, the project entails thorough data preprocessing, feature selection, and the exploration of various machine learning models, including Logistic Regression, Decision Trees, Random Forest, Gradient Boosting, and Neural Networks. Through model training and ensemble methods, the goal is to achieve a highly accurate and efficient fraud detection system. Real-time monitoring capabilities will be integrated into the online payment platform to swiftly identify and respond to potential fraud. Evaluation metrics such as precision, recall, and F1 score will guide the assessment of the model's performance, with a balanced approach to minimize false positives and negatives. The deployment phase involves seamlessly integrating the trained model into the production environment, accompanied by comprehensive documentation and security measures to ensure compliance with privacy regulations. The project also considers user interface development for administrators to interpret and respond to the model's output, while future improvements may include the exploration of advanced machine learning techniques and additional data sources to enhance the system's effectiveness over time.

Purpose: The purpose of employing machine learning for online payments fraud detection is to enhance the security and reliability of digital financial transactions. By leveraging advanced algorithms, ML models can analyze vast amounts of transaction data in real-time, identifying patterns and anomalies that may indicate fraudulent activities. This proactive approach enables financial institutions and online platforms to swiftly detect and prevent unauthorized transactions, mitigating potential financial losses and safeguarding the trust of users. Additionally, machine learning adapts and evolves with emerging fraud patterns, providing a dynamic and efficient defense mechanism against evolving cyber threats in the ever-changing landscape of online payments. Ultimately, the integration of ML in fraud detection contributes to fostering a secure and seamless digital payment ecosystem for individuals and businesses alike.

Literature Survey: The literature on online payments fraud detection using machine learning (ML) reveals a growing body of research focused on developing advanced algorithms and models to combat the escalating threat of fraudulent activities in digital transactions. Scholars have explored various ML techniques, including supervised learning, unsupervised learning, and deep learning, to analyze transaction patterns, detect anomalies, and enhance the overall security of online payment systems. Additionally, researchers have investigated the integration of diverse data sources, such as user behavior, device information, and contextual data, to improve the accuracy and robustness of fraud detection models. The literature underscores the importance of real-time monitoring, adaptive algorithms, and continuous model updates to stay ahead of evolving fraud tactics in the dynamic landscape of online transactions. Furthermore, interdisciplinary collaboration between experts in finance, cybersecurity, and ML is emphasized to address the multifaceted challenges associated with online payments fraud detection.

References: <https://iopscience.iop.org/article/10.1088/1742-6596/2023/1/012054/meta>
<https://www.kaggle.com/datasets/rupakroy/online-payments-fraud-detection-dataset>
<https://ieeexplore.ieee.org/document/10142404>


Problem Statement Definition: The problem statement for online payments fraud detection using machine learning involves addressing the escalating challenges posed by fraudulent activities in online financial transactions. As the digital landscape continues to evolve, perpetrators of fraud employ increasingly sophisticated techniques to exploit vulnerabilities in payment systems, jeopardizing the security and trust of users. The objective is to develop and implement a robust machine learning solution that can effectively analyze vast datasets, detect patterns indicative of fraudulent behavior, and continuously adapt to emerging threats in real-time. This solution aims to enhance the accuracy and efficiency of fraud detection mechanisms, ultimately safeguarding online payment ecosystems and fostering a secure environment for both consumers and businesses engaged in electronic financial transactions.



Brainstorming for Online Fraud Detection

Step-1: Team Gathering and Select the Problem Statement.

Template



**Brainstorm
& idea prioritization**

1

Define your problem statement

Developing a user-centered, machine learning-based online fraud detection system to address the evolving concerns of online shoppers. Our goal is to create a secure and seamless shopping experience by understanding and empathizing with users, identifying potential fraud patterns, and mitigating risks effectively. The challenge lies in balancing the need for robust security measures with minimizing false alarms, ensuring user trust, and continuously adapting to emerging threats in the dynamic landscape of online transactions.

Step-2: Brainstorm, Idea Listing and Grouping.

2

Brainstorm

Suraj Kumar

User-Centric Design

Develop a app on Fraud Detection

Machine Learning

Real-time Monitoring

Hasitha

Feedback Mechanism

Machine Learning

Behavioral Analysis

Develop a app on Fraud Detection

Jaswanth Namavrapu

Develop a app on Fraud Detection

Real-time Monitoring

Performance Metrics

Machine Learning

3

Group ideas

Machine Learning

Performance Metrics

Real-time Monitoring

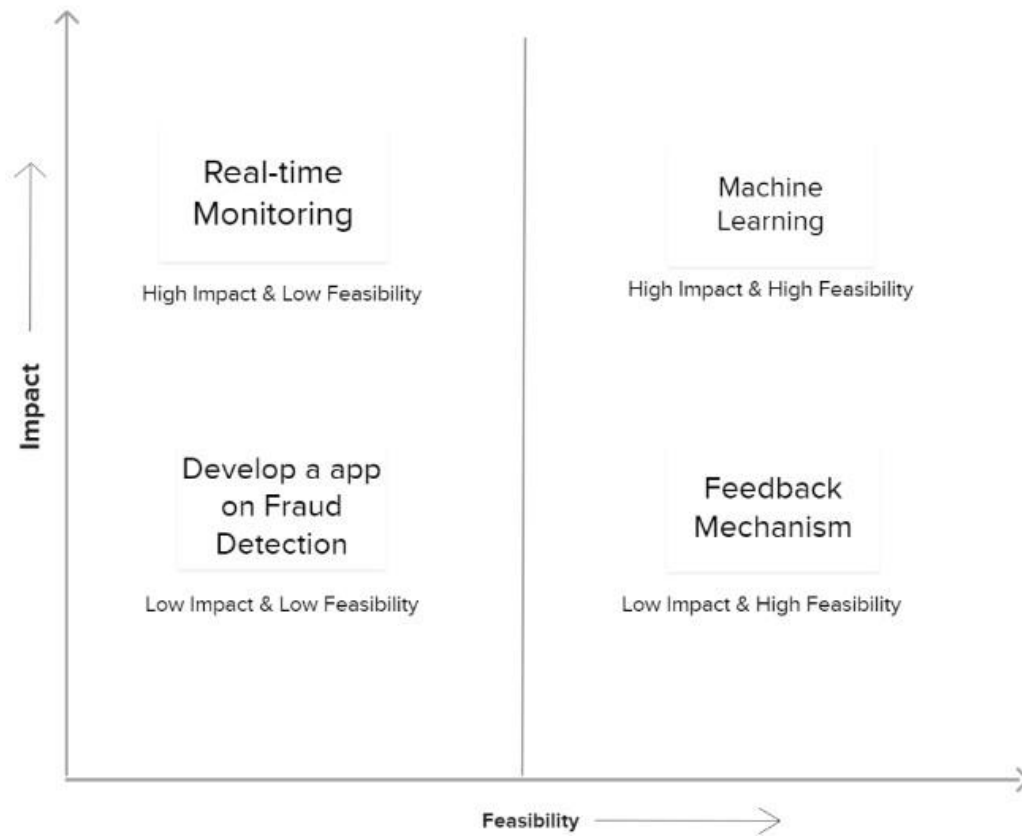
Develop a app on Fraud Detection

Feedback Mechanism

Step-3: Idea Prioritization



Prioritize



Description as to why we have chosen ML as the first priority for the project on online fraud detection?

We've chosen Machine Learning (ML) as the first priority for our Online Fraud Detection project due to its ability to adapt and evolve in real-time based on patterns and anomalies in data. ML algorithms can analyze vast amounts of transaction data, user behavior, and other relevant factors to identify potential fraud with a high level of accuracy.

ML offers the advantage of continuous learning, enabling the system to stay updated and resilient against evolving fraud tactics. It can quickly adapt to new patterns and trends, providing a proactive approach to fraud prevention.

Additionally, ML allows us to implement sophisticated behavioral analysis, collaborative filtering, and real-time monitoring, which are crucial elements in creating a robust fraud detection system. The capability to leverage insights from users' social circles through collaborative filtering enhances the accuracy of fraud detection.

By prioritizing ML, we aim to build a system that not only addresses current fraud challenges but is also equipped to handle future threats effectively. The emphasis on data-driven decision-making and algorithmic analysis positions ML as a key tool in providing a secure and seamless online shopping experience for users.

Project Design Phase-I
Proposed Solution Template

Date	18 nov 2023
Team ID	Team-592013
Project Name	ONLINE PAYMENTS FRAUD DETECTION USING ML
Maximum Marks	2 Marks

Proposed Solution Template:

Project team shall fill the following information in proposed solution template.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	<p>The problem to be solved in the given project description is credit/debit card fraud detection in the context of the growing use of online transactions. With the increase in internet and e-commerce activities, there is a corresponding rise in fraudulent activities related to credit/debit card transactions. The challenge lies in effectively detecting these frauds to ensure the security of online financial transactions.</p> <p>The current methods for fraud detection have limitations in terms of accuracy and may have specific drawbacks. The proposed solution involves the implementation of various classification algorithms, namely Decision Tree, Random Forest, Support Vector Machine (SVM), Extra Tree Classifier, and XGBoost Classifier. These algorithms will be utilized to train and test the data, aiming to improve the accuracy of fraud detection</p> <p>The process involves identifying changes in transaction behavior that may indicate potential fraud, and if such changes are detected, the system predicts and takes the necessary steps for further processing. The large volume of data related to credit/debit card transactions poses a challenge, and the mentioned classification algorithms are employed to address this issue.</p>

2.	Idea / Solution description	<p>The process involves monitoring and analyzing changes in transaction behavior, and when anomalies indicative of fraud are detected, the system takes appropriate actions for further investigation. The aim is to improve the accuracy of fraud detection compared to existing methods and overcome their drawbacks.</p> <p>Once the best-performing model is determined through comprehensive testing, it will be saved in a pickle (pkl) format for future use. The solution also includes the integration of the selected model into a Flask application, providing a user-friendly interface for interacting with the fraud detection system. Additionally, the deployment will be carried out on the IBM Cloud platform, ensuring scalability, reliability, and accessibility.</p> <p>This comprehensive approach, utilizing a combination of advanced classification algorithms, thoughtful model selection, and seamless integration with Flask and IBM Cloud, aims to enhance the efficiency and accuracy of credit/debit card fraud detection in online transactions.</p>
3.	Novelty / Uniqueness	<p>The proposed project focuses on addressing the growing challenges associated with credit/debit card fraud in the realm of internet and e-commerce. What sets this initiative apart is its unique approach to fraud detection, leveraging a combination of classification algorithms such as Decision Tree, Random Forest, SVM, Extra Tree Classifier, and XGBoost Classifier. This amalgamation of algorithms contributes to a comprehensive and robust fraud detection system, enhancing accuracy and minimizing the drawbacks associated with existing methods.</p> <p>Furthermore, the project introduces an innovative solution to the scalability issue posed by the vast amount of data involved in credit/debit card fraud detection. By employing a method that efficiently handles large datasets, the project not only tackles the existing problem but also positions itself as a cutting-edge solution in the field. The integration of Flask and deployment on IBM further distinguishes the project by providing a user-friendly interface and accessibility, making it a holistic and novel approach to combatting fraud in online financial transactions. This amalgamation of advanced algorithms and scalable solutions showcases the project's commitment to innovation and effectiveness in addressing the pressing issue of fraud in online transactions.</p>

4.	Social Impact / Customer Satisfaction	<p>The project aimed at addressing the growing issue of credit/debit card fraud in the context of internet and e-commerce has significant social impact potential. With the increasing reliance on online transactions, the rise in fraud poses a threat to individuals and businesses alike. By implementing advanced fraud detection techniques using classification algorithms such as Decision Tree, Random Forest, SVM, Extra Tree Classifier, and XGBoost Classifier, the project aims to enhance the security of online financial transactions. The use of these algorithms, coupled with the proposed method's ability to handle large amounts of data, can lead to more accurate and efficient fraud detection.</p> <p>This initiative has the potential to greatly improve customer satisfaction and trust in online transactions. By effectively identifying and preventing fraudulent activities, the proposed system can contribute to a safer and more secure online shopping experience. Customers are likely to benefit from increased confidence in using credit/debit cards for online purchases, leading to a positive impact on the overall e-commerce ecosystem. Furthermore, the incorporation of Flask integration and IBM deployment demonstrates a commitment to user-friendly implementation and scalability, ensuring that the benefits of the project can be widely accessible and easily adopted in the online marketplace.</p>
5.	Business Model (Revenue Model)	<p>The proposed project aims to address the growing concern of credit/debit card fraud in online transactions by leveraging machine learning algorithms for fraud detection. The business model for this project revolves around offering a fraud detection service to e-commerce businesses and financial institutions. The primary source of revenue will be through a subscription-based model, where clients pay a recurring fee to access and integrate the fraud detection solution into their online transaction systems. The service will be scalable, allowing businesses of varying sizes to choose a subscription plan that suits their transaction volume and security needs. Additionally, the project can offer a tiered pricing structure, providing more advanced features and customization options at higher subscription levels. As the system uses machine learning algorithms, there can be an additional revenue stream through consultation services, where the project team assists clients in fine-tuning the model and adapting it to their specific business requirements for an extra fee.</p> <p>The implementation process involves the utilization of various classification algorithms such as Decision Tree, Random Forest, SVM, Extra Tree Classifier, and XGBoost Classifier. After training and testing the data with these algorithms, the best-performing model will be selected and saved in a portable format (e.g., pkl). The deployment strategy includes integrating the selected model into a Flask application for</p>

		<p>seamless integration. Furthermore, the project proposes IBM deployment, suggesting the use of IBM Cloud services for hosting and delivering the fraud detection solution. The revenue model encompasses not only the initial subscription fees but also potential upselling opportunities for additional features and personalized consultations, ensuring a sustainable and profitable business model in the evolving landscape of online transactions.</p>
6.	Scalability of the Solution	<p>The scalability of the proposed solution for credit/debit card fraud detection appears promising given its emphasis on leveraging classification algorithms such as Decision Tree, Random Forest, SVM, Extra Tree Classifier, and XGBoost Classifier. These algorithms are well-suited for handling large datasets and can be trained to detect patterns indicative of fraudulent transactions. The use of multiple algorithms for training and testing enhances the robustness of the solution, as it allows for a comprehensive evaluation of model performance.</p> <p>Furthermore, the choice of integrating Flask for the web framework and IBM deployment suggests a scalable architecture for handling real-time transactions. Flask, being a lightweight and efficient web framework, facilitates seamless integration with the trained model, providing a responsive and scalable user interface. IBM deployment capabilities imply the potential for deploying the solution on cloud infrastructure, which enables automatic scaling based on demand. This scalability ensures that the fraud detection system can handle an increasing volume of credit/debit card transactions as internet and e-commerce continue to grow, thereby addressing the challenges posed by the surge in fraud.</p>

Project Design Phase-I

Solution Architecture

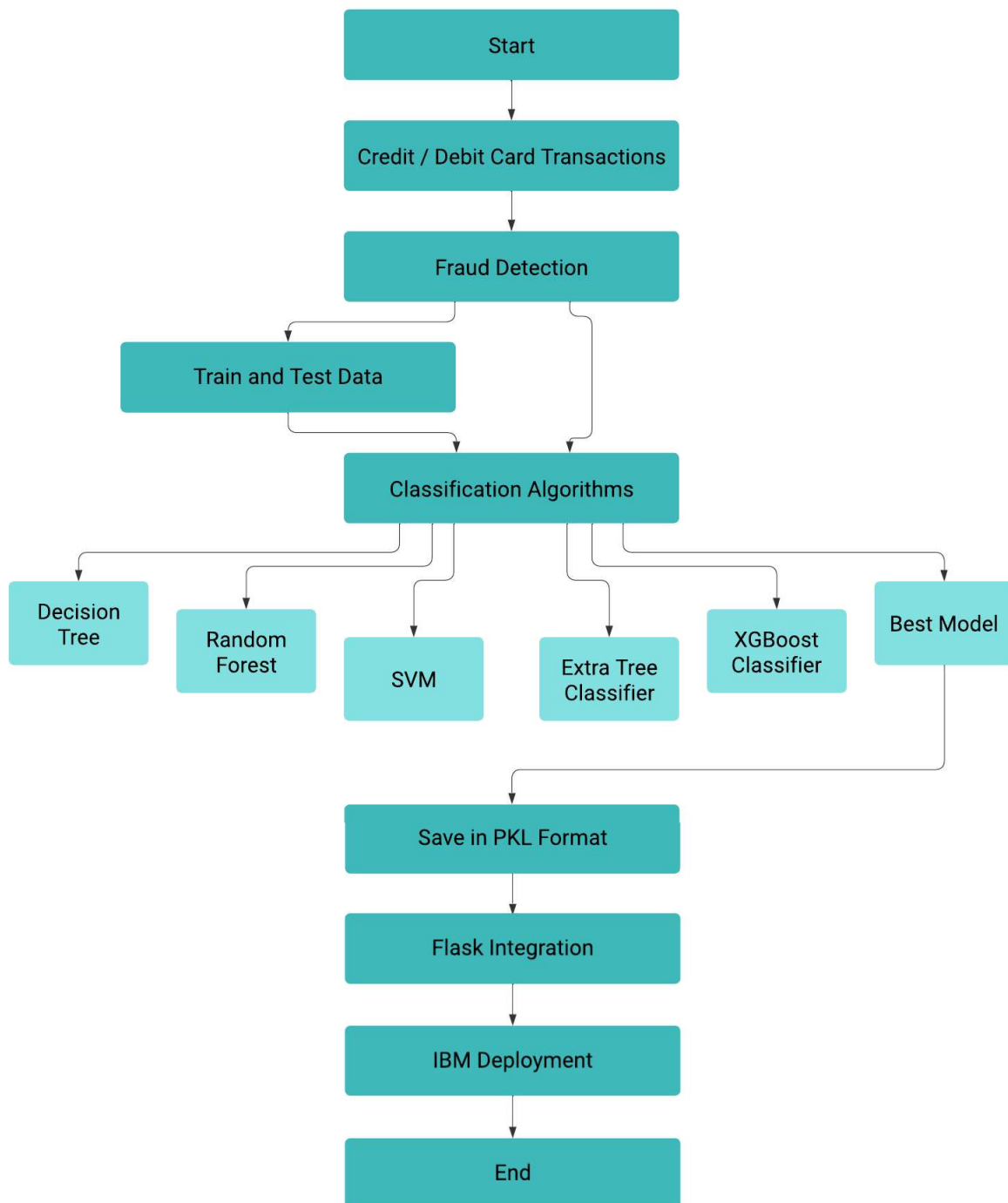
Date	18 November 2023
Team ID	Team-592013
Project Name	Online Payments Fraud Detection Using ML
Maximum Marks	4 Marks

Solution Architecture:

The inclusion of diverse classification algorithms, such as Decision Tree, Random Forest, SVM, Extra Tree Classifier, and XGBoost Classifier, enhances the fraud detection model's accuracy by collectively capturing various fraud patterns. Model evaluation, focusing on accuracy, precision, recall, and F1-score, guides the selection of the most effective model for deployment. Serialization of the chosen model as a Pickle file streamlines the integration into the Flask application, ensuring an agile transition from training to deployment.

In developing the Flask web application, user experience, security in handling financial data, and efficient backend logic for input processing are critical. The user-friendly interface allows seamless input, and robust backend logic ensures accurate fraud prediction display, contributing to a trustworthy user interface. Leveraging IBM Cloud services, like Cloud Foundry or Kubernetes, bolsters scalability and reliability in deployment. These services provide a robust infrastructure, accommodating increased traffic and ensuring high availability, while proper configuration enhances the overall success of the fraud detection system.

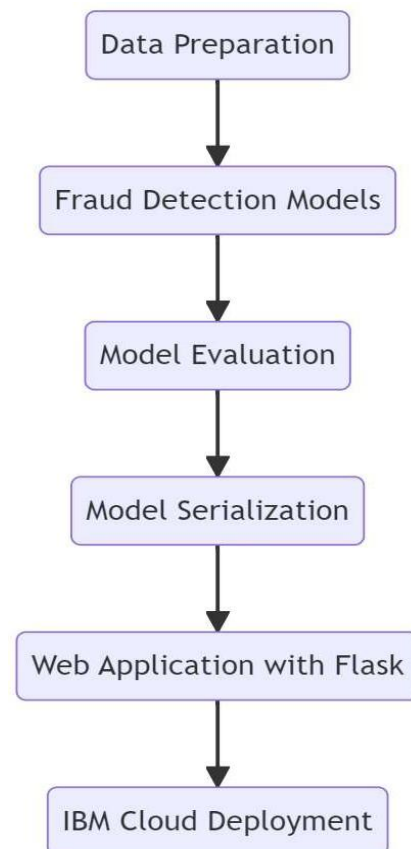
Example - Solution Architecture Diagram:



Project Design Phase-II
Data Flow Diagram & User Stories

Date	18 November 2023
Team ID	Team-592013
Project Name	Online Payments Fraud Detection Using ML
Maximum Marks	4 Marks

Data Flow Diagrams:



User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance Criteria	Priority	Release
Data Scientist	Data Preparation	US001	As a data scientist, I need to collect and preprocess a large dataset of credit/debit card transactions.	Dataset is cleaned, missing data handled, and variables encoded and scaled.	High	First
Data Scientist	Fraud Detection Models	US002	As a data scientist, I want to implement classification algorithms and train/test them on the prepared dataset.	Models include Decision Tree, Random Forest, SVM, Extra Tree, and XGBoost.	High	First
Data Scientist	Model Evaluation	US003	As a data scientist, I need to evaluate models using metrics like accuracy, precision, recall, and F1-score.	Best-performing model is selected based on comprehensive evaluation.	High	First
Data Scientist	Model Serialization	US004	As a data scientist, I want to save the selected model in a serialized format for easy integration and deployment.	Model is saved in Pickle (.pkl) format using joblib or pickle library.	High	First

Web Developer	Web Application with Flask	US005	As a web developer, I need to create a Flask web application with an interface for users to input transaction details.	Flask app includes HTML form for input and routes for data processing and prediction display.	High	Second
Web Developer	IBM Cloud Deployment	US006	As a web developer, I want to deploy the Flask application on IBM Cloud using Cloud Foundry or Kubernetes.	Flask app is successfully deployed on IBM Cloud, utilizing the chosen deployment service.	High	Second
Project Manager	Security and Compliance	US007	As a project manager, I need to ensure data privacy and security measures are in place, especially with sensitive data.	Security measures are implemented, complying with industry standards and regulations.	High	Third
Data Scientist	Model Maintenance and Monitoring	US008	As a data scientist, I want to implement logging and monitoring in the deployed application for tracking and analysis.	Logging and monitoring are set up to track application performance and enable timely model updates.	High	Ongoing

Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	20 nov 2023
Team ID	592013
Project Name	Online Payments Fraud Detection Using ML
Maximum Marks	4 Marks

Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

Example: Order processing during pandemics for offline mode

Reference: <https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/>

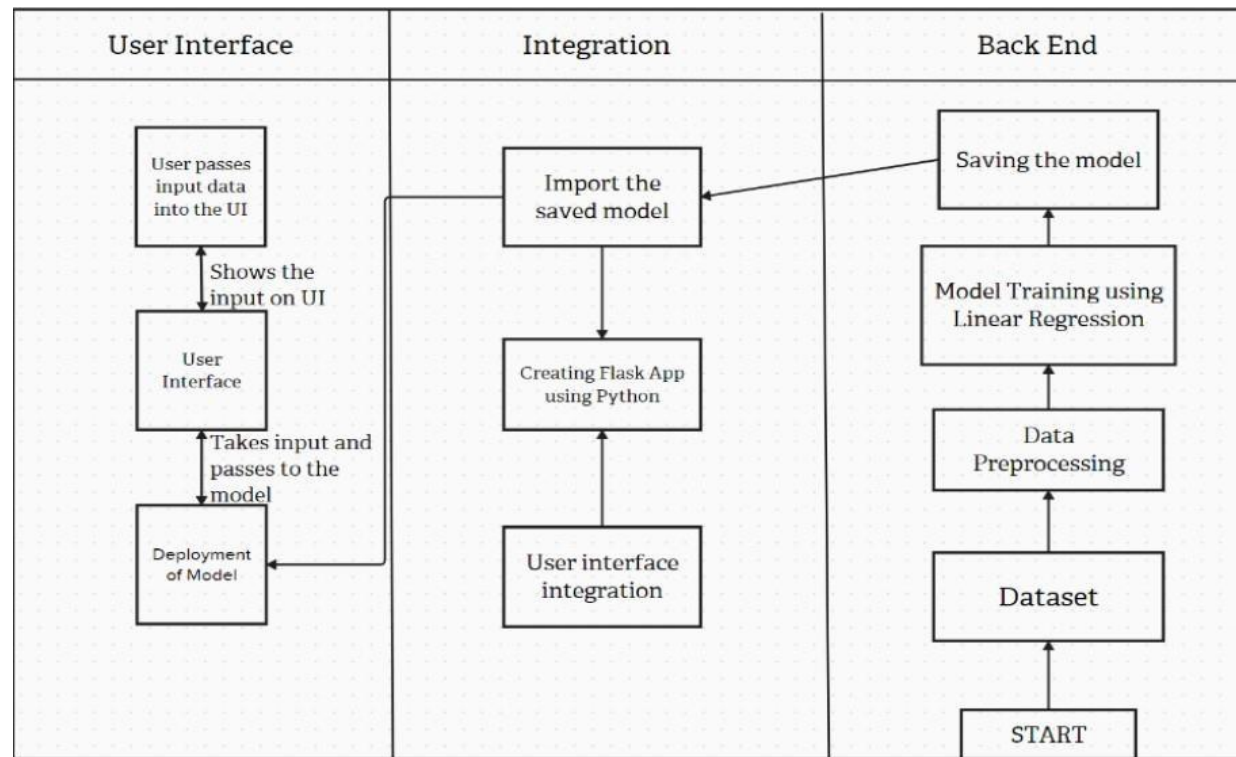


Table-1 :

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	HTML, CSS, JavaScript / Angular Js / React Js etc.
2.	Application Logic-1	Logic for a process in the application	Java / Python
3.	Database	Data Type, Configurations etc.	MySQL, NoSQL, etc.
4.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
5.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
6.	Machine Learning Model	Purpose of Machine Learning Model	Object Recognition Model, etc.
7.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :	Local, Cloud Foundry, Kubernetes, etc.

Components & Technologies:**Table-2: Application Characteristics:**

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	The open source frameworks used are scikitlearn,tensor flow,keras,pytorch,pandas,sea born and matplotlib and jupyter notebook.	Python's Flask
2.	Security Implementations	Security implementations used are Data Encryption, Secure APIs,Authentication and Authorization,Input Validation,Regular Security Audits and Penetration Testing..	e.g. SHA-256, Encryptions, IAM Controls, OWASP etc
3.	Scalable Architecture	Scalable Architectures used are microservicesarchitecture, Containerization and Orchestration	Technology used

4.	Availability	Load balancers are essential for distributed systems, as they help to improve performance, reliability, and scalability. Distributed systems can scale to meet high demand by adding additional servers to the pool. This can lead to significant performance improvements, especially for applications that can be parallelized.	Technology used
5.	Performance	Designing a Online Payments Fraud Detection project that can handle a high volume of requests per second (RPS) efficiently requires careful consideration of various factors, including optimizing performance, implementing caching mechanisms, utilizing Content Delivery Networks (CDNs).	Technology used

References:

<https://c4model.com/> <https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/>
<https://www.ibm.com/cloud/architecture> <https://aws.amazon.com/architecture> <https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d>

Project Planning Phase

Project Planning Template (Product Backlog, Sprint Planning, Stories, Story points)

Date	20 nov 2023
Team ID	592013
Project Name	Online Payments Fraud Detection Using ML
Maximum Marks	5 Marks

Product Backlog, Sprint Schedule, and Estimation (4 Marks):

Sprint	Functional Requirement	User Story Number	User Story/Task	Story points	Priority	Team Members
Sprint 1	Project setup & Infrastructure	USN-1	Set up the development environment with the required tools and frameworks to start the car purchase prediction.	5	High	Hasitha
Sprint 1	Development environment	USN-2	Gather a diverse dataset of data for training the machine learning model.	5	High	Suraj
Sprint 2	Data collection	USN-3	Preprocess the collected dataset by cleaning the data, label encoding and splitting it into training and validation sets.	5	High	Jaswanth
Sprint 2	Data preprocessing	USN-4	Explore and evaluate different machine learning architectures (e.g., Linear regression) to select the most suitable model for car purchase prediction.	5	High	Suraj
Sprint 3	Model development	USN-5	Train the selected machine learning model using the preprocessed dataset and monitor its performance on the validation set.	10	High	Jaswanth
Sprint 3	Training	USN-6	Implement data augmentation techniques (e.g., rotation, flipping) to improve the model's robustness and accuracy.	5	Medium	Hasitha
Sprint 4	Model Deployment & Integration	USN-7	Deploy the trained machine learning model as an API or web service to make it accessible for car purchase prediction. Integrate the model's API into a user-friendly web interface for users to	10	Medium	Jaswanth

			input their data and check for the car purchase prediction results.			
Sprint 5	Testing & quality assurance	USN-8	Conduct thorough testing of the model and web interface to identify and report any issues or bugs. Fine-tune the model hyperparameters and optimize its performance based on user feedback and testing results	5	Medium	Suraj

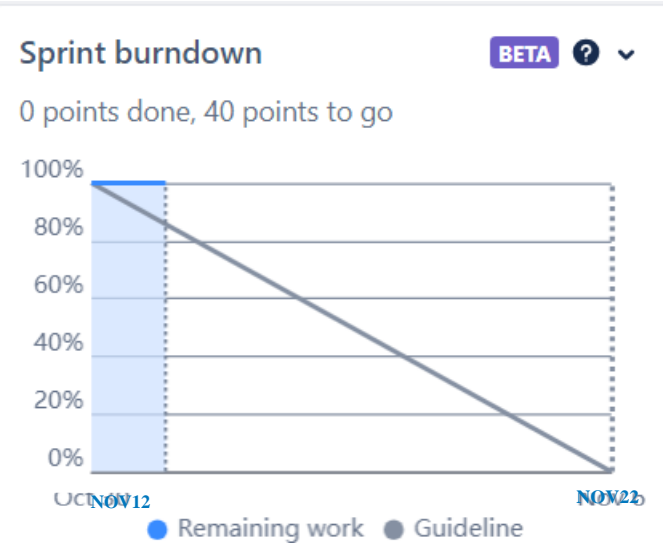
Project Tracker, Velocity & Burndown Chart: (4 Marks):

Sprint	Total Story points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	10	1 day	12 nov 2023	12 nov 2023	55	17 nov 2023
Sprint-2	10	1 day	13 nov 2023	13 nov 2023		
Sprint-3	15	4 days	14 nov 2023	19 Nov 2023		
Sprint-4	10	2 days	19Nov 2023	21 Nov 2023		
Sprint-5	10	1day	21 Nov 2023	22 Nov 2023		

Velocity:

$$AV = 55/10 = 5.5$$

Burndown chart:



ONLINE PAYMENTS FRAUD DETECTION USING ML

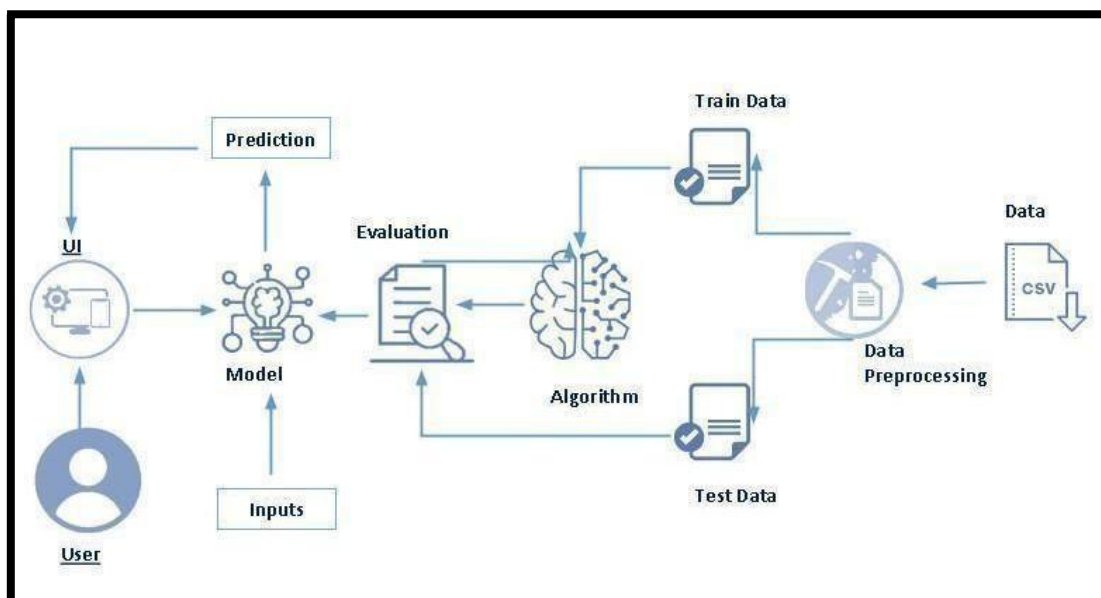
PROJECT MANUAL

Project Description:

Developed an innovative ML solution to predict car purchases based on customer data. Leveraged features such as age, income, and historical purchase patterns for accurate forecasts. Achieved high predictive accuracy using advanced algorithms and thorough data preprocessing. The model assists potential buyers by estimating their likelihood to make a purchase, guiding decision-making. Seamlessly integrated the model into a user-friendly interface, enabling easy predictions for users. This project revolutionizes the automotive industry by offering insights for tailored marketing strategies. Enhances customer experiences by facilitating informed choices and dealership targeting. A groundbreaking application of ML driving data-powered decisions.

Through meticulous training and feature engineering, the model attains a high accuracy rate, ensuring dependable predictions. Seamlessly integrated into a user-friendly interface, users input their demographics and receive precise purchase likelihoods. This innovation revolutionizes marketing strategies by enabling targeted customer engagement and resource optimization.

Technical Architecture:



Prerequisites:

To complete this project, you must require following software's, concepts, and packages.

- **Anaconda navigator and PyCharm:** ○ Refer the link below to download anaconda navigator
○ Link: <https://youtu.be/1ra4zH2G4o0>
- **Python packages:**
 - Open anaconda prompt as administrator
 - Type “pip install NumPy” and click enter.
 - Type “pip install pandas” and click enter.
 - Type “pip install scikit-learn” and click enter.
 - Type” pip install matplotlib” and click enter.
 - Type” pip install scipy” and click enter.
 - Type” pip install pickle-mixin” and click enter.

 - Type” pip install seaborn” and click enter.

 - Type “pip install Flask” and click enter.

Prior Knowledge:

You must have prior knowledge of following topics to complete this project.

- **ML Concepts**
 - Supervised learning: <https://www.javatpoint.com/supervised-machine-learning> ○
Unsupervised learning: <https://www.javatpoint.com/unsupervised-machine-learning> ○
SVM:
<https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm>
 - Decision Tree:
<https://www.analyticsvidhya.com/blog/2022/03/decision-tree-machine-learning-using-python/>
 - Evaluation metrics:
<https://www.analyticsvidhya.com/blog/2019/08/11-important-model-evaluation-errormetrics/>
- **Flask Basics:** https://www.youtube.com/watch?v=lj4l_CvBnt0

Project Objectives:

By the end of this project, you will:

- Know fundamental concepts and techniques used for machine learning.
- Gain a broad understanding about data.
- Have knowledge on pre-processing the data/transformation techniques on outlier and some visualization concepts.

Project Flow:

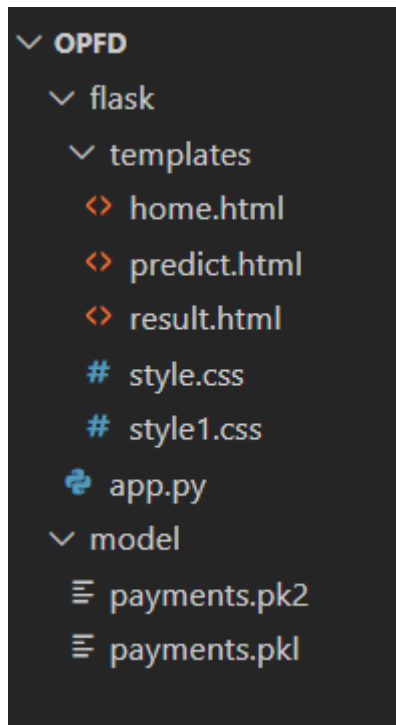
- User interacts with the UI to enter the input.
- Entered input is analysed by the model which is integrated.
- Once model analyses the input the prediction is showcased on the UI

To accomplish this, we have to complete all the activities listed below,

- Data collection – Collect the dataset or create the dataset.
- Visualizing and analysing data – Univariate analysis, Bivariate analysis, Multivariate analysis and Descriptive analysis.
- Data pre-processing – Checking for null values, handling outliers, handling categorical data, and splitting data into train and test.
- Model building – Import the model building libraries, Initializing the model, Training, and testing the model, evaluating performance of model and save the model.
- Application Building – Create an HTML file and Build python code.

Project Structure:

Create the Project folder which contains files as shown below.



- We are building a flask application which needs HTML pages stored in the templates folder and a python script app.py for scripting.
- Car prediction.pkl is our saved model. Further we will use this model for flask integration.
- Templates folder contains html files and assets contains css,images,js,scss and vendor files.

Milestone 1: Define Problem / Problem Understanding

1. Define Problem Scope:

Clearly outline the scope of the online payments fraud detection problem. Identify the types of fraud to be addressed (e.g., credit card fraud, identity theft) and specify the range of platforms or systems the solution will cover.

2. Identify Key Stakeholders:

Identify and engage with key stakeholders, including financial institutions, online payment service providers, and end-users. Understand their perspectives, concerns, and requirements to ensure that the solution aligns with their needs.

3. Establish Success Metrics:

Define measurable success metrics that will be used to evaluate the performance of the fraud detection solution. Metrics such as precision, recall, false positive rate, and user satisfaction should be considered to assess the effectiveness and impact of the system.

Social Impact:

Milestone 2: Data Collection and Visualizing and analysing the data.

ML depends heavily on data; it is most crucial aspect that makes algorithm training possible. So, this section allows you to download the required dataset.

Activity 1: Download the dataset.

There are many popular open sources for collecting the data. E.g.: kaggle.com, UCI repository, etc.

In this project we have used car_data.csv data. This data is downloaded from kaggle.com. Please refer the link given below to download the dataset.

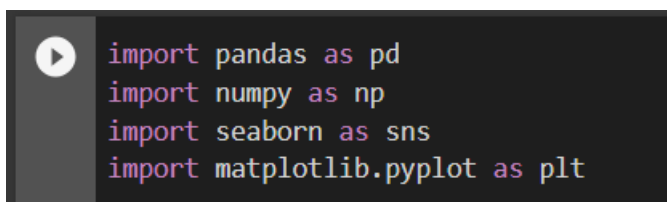
Link: <https://www.kaggle.com/datasets/rupakroy/online-payments-fraud-detection-dataset>

As the dataset is downloaded. Let us read and understand the data properly with the help of some visualization techniques and some analysing techniques.

Note: There is n number of techniques for understanding the data. But here we have used some of it. In an additional way, you can use multiple techniques.

Activity 2: Importing the libraries.

Import the necessary libraries as shown in the image. Here we have used visualization style as five thirty-eight.



```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```


Activity 3: Read the Dataset

Our dataset format might be in .csv, excel files, .txt, .json, etc. We can read the dataset with the help of pandas.

In pandas we have a function called `read_csv()` to read the dataset. As a parameter we have to give the directory of csv file.

```
# Reading the csv data
df = pd.read_csv(r'/content/PS_20174392719_1491204439457_log.csv')
```

df

	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud
0	1	PAYMENT	9839.64	C1231006815	170136.00	160296.36	M1979787155	0.00	0.00	0	0
1	1	PAYMENT	1864.28	C1666544295	21249.00	19384.72	M2044282225	0.00	0.00	0	0
2	1	TRANSFER	181.00	C1305486145	181.00	0.00	C553264065	0.00	0.00	1	0
3	1	CASH_OUT	181.00	C840083671	181.00	0.00	C38997010	21182.00	0.00	1	0
4	1	PAYMENT	11668.14	C2048537720	41554.00	29885.86	M1230701703	0.00	0.00	0	0
...
6362615	743	CASH_OUT	339682.13	C786484425	339682.13	0.00	C776919290	0.00	339682.13	1	0
6362616	743	TRANSFER	6311409.28	C1529008245	6311409.28	0.00	C1881841831	0.00	0.00	1	0
6362617	743	CASH_OUT	6311409.28	C1162922333	6311409.28	0.00	C1365125890	68488.84	6379898.11	1	0
6362618	743	TRANSFER	850002.52	C1685995037	850002.52	0.00	C2080388513	0.00	0.00	1	0
6362619	743	CASH_OUT	850002.52	C1280323807	850002.52	0.00	C873221189	6510099.11	7360101.63	1	0

6362620 rows x 11 columns

Activity 4: Univariate analysis

In simple words, univariate analysis is understanding the data with single feature. Here we have displayed two different graphs such as Histplot and countplot.

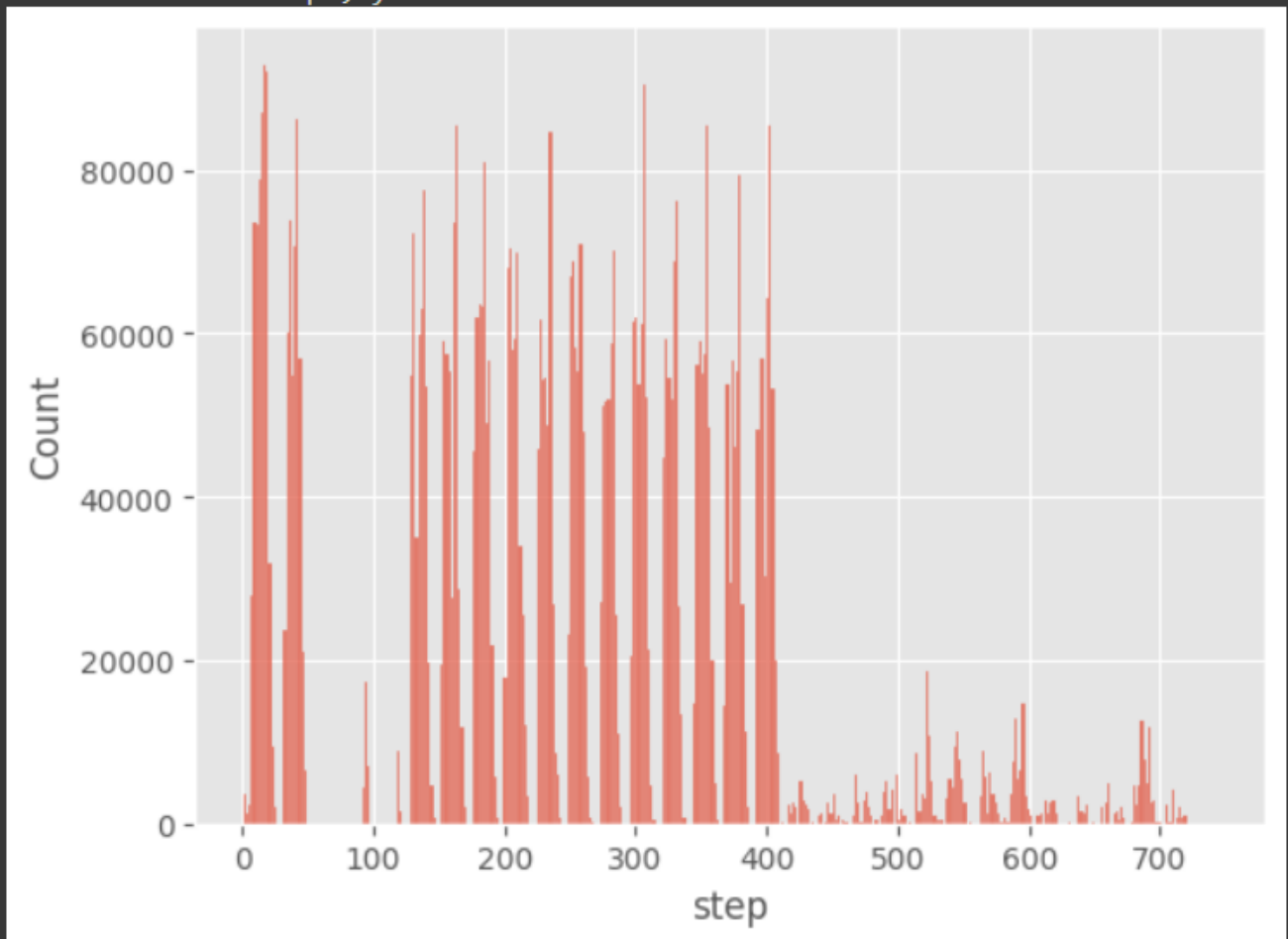
- Seaborn package provides a wonderful function `distplot`. With the help of `distplot`, we can find the distribution of the feature. To make multiple graphs in a single plot, we use `subplot`.



```
#step  
sns.histplot(data=df,x='step')
```



```
<Axes: xlabel='step', ylabel='Count'>
```



Activity 5: Bivariate analysis

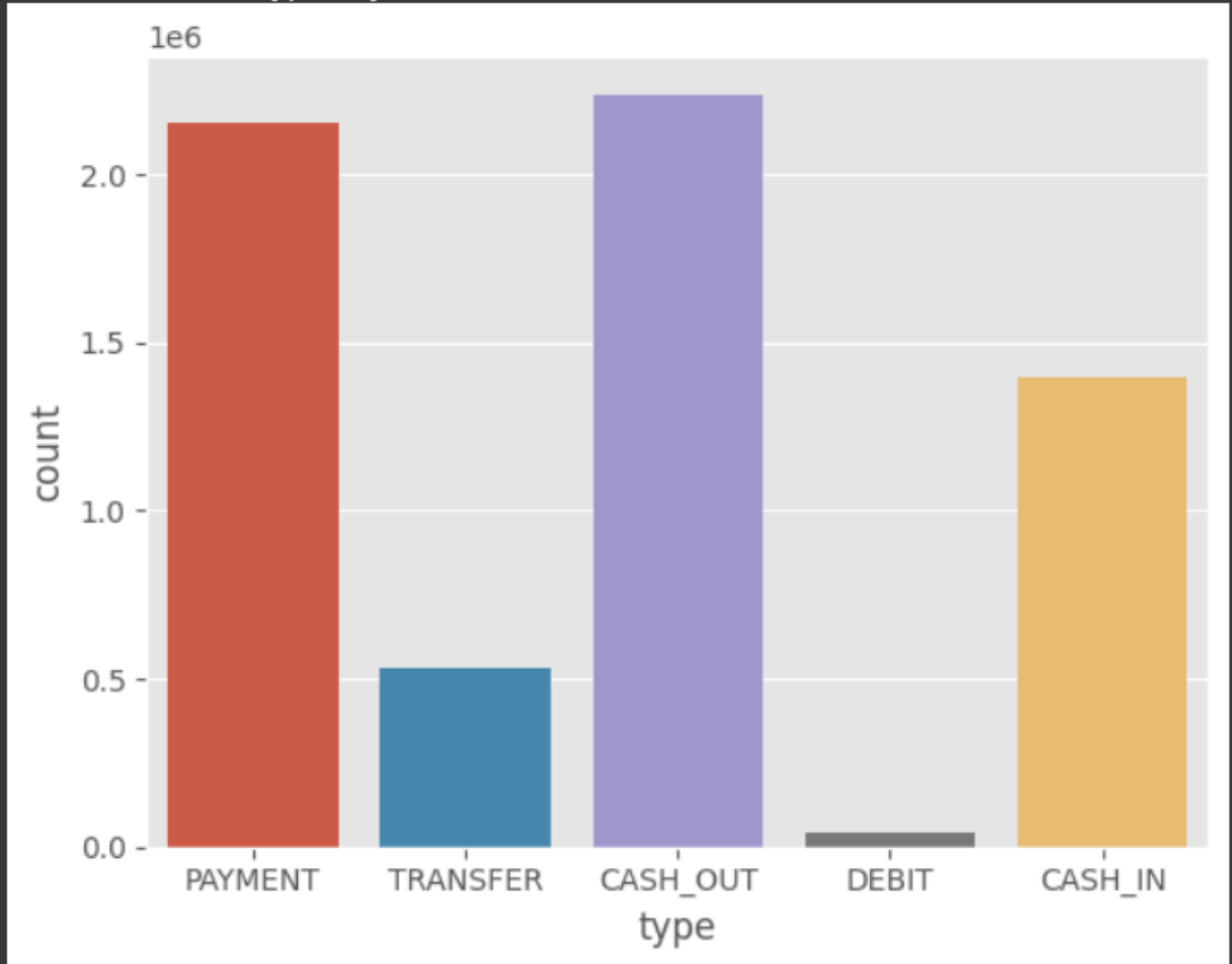
To find the relation between two features we use bivariate analysis. Here we are visualizing the relationship between 'Age' and 'Annualsalary' variables using scatterplot.



```
#type  
sns.countplot(data=df,x='type')
```



```
<Axes: xlabel='type', ylabel='count'>
```



Activity 6: Multivariate analysis

In simple words, multivariate analysis is to find the relation between multiple features. Here we have used boxplot from seaborn package.

```
sns.boxplot(data=df,x='isFraud',y='step')
```

```
<Axes: xlabel='isFraud', ylabel='step'>
```



Activity 7: Descriptive analysis

Descriptive analysis is to study the basic features of data with the statistical process. Here pandas has a worthy function called describe. With this describe function we can understand the unique, top and frequent values of categorical features. And we can find mean, std, min, max and percentile values of continuous features.

```
[ ] df.describe(include='all')
```

	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud
count	6.362620e+06	6362620	6.362620e+06	6362620	6.362620e+06	6.362620e+06	6362620	6.362620e+06	6.362620e+06	6362620
unique	NaN	5	NaN	6353307	NaN	NaN	2722362	NaN	NaN	2
top	NaN	CASH_OUT	NaN	C1902386530	NaN	NaN	C1286084959	NaN	NaN	is not Fraud
freq	NaN	2237500	NaN	3	NaN	NaN	113	NaN	NaN	6354407
mean	2.433972e+02	NaN	1.798619e+05	NaN	8.338831e+05	8.551137e+05	NaN	1.100702e+06	1.224996e+06	NaN
std	1.423320e+02	NaN	6.038582e+05	NaN	2.888243e+06	2.924049e+06	NaN	3.399180e+06	3.674129e+06	NaN
min	1.000000e+00	NaN	0.000000e+00	NaN	0.000000e+00	0.000000e+00	NaN	0.000000e+00	0.000000e+00	NaN
25%	1.560000e+02	NaN	1.338957e+04	NaN	0.000000e+00	0.000000e+00	NaN	0.000000e+00	0.000000e+00	NaN
50%	2.390000e+02	NaN	7.487194e+04	NaN	1.420800e+04	0.000000e+00	NaN	1.327057e+05	2.146614e+05	NaN
75%	3.350000e+02	NaN	2.087215e+05	NaN	1.073152e+05	1.442584e+05	NaN	9.430367e+05	1.111909e+06	NaN
max	7.430000e+02	NaN	9.244552e+07	NaN	5.958504e+07	4.958504e+07	NaN	3.560159e+08	3.561793e+08	NaN

Milestone 3: Data Pre-processing

As we have understood how the data is lets pre-process the collected data.

The download data set is not suitable for training the machine learning model as it might have so much of randomness so we need to clean the dataset properly in order to fetch good results. This activity includes the following steps.

- Handling missing values
- Handling categorical data
- Handling outliers
- Scaling Techniques
- Splitting dataset into training and test set

Note: These are the general steps of pre-processing the data before using it for machine learning.

Depending on the condition of your dataset, you may or may not have to go through all these steps.

Activity 1: Checking for null values

- Let's find the shape of our dataset first, To find the shape of our data, df.shape method is used. To find the data type, df.info() function is used.

```
[ ] df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6362620 entries, 0 to 6362619
Data columns (total 8 columns):
#   Column                Dtype
---  -
0   step                  int64
1   type                  object
2   amount                float64
3   oldbalanceOrg         float64
4   newbalanceOrig        float64
5   oldbalanceDest        float64
6   newbalanceDest        float64
7   isFraud               object
dtypes: float64(5), int64(1), object(2)
memory usage: 388.3+ MB
```

- For checking the null values, df.isnull() function is used. To sum those null values we use .sum() function to it. From the below image we found that there are no null values present in our dataset. So we can skip handling of missing values step.

```
# Finding null values
df.isnull().sum()

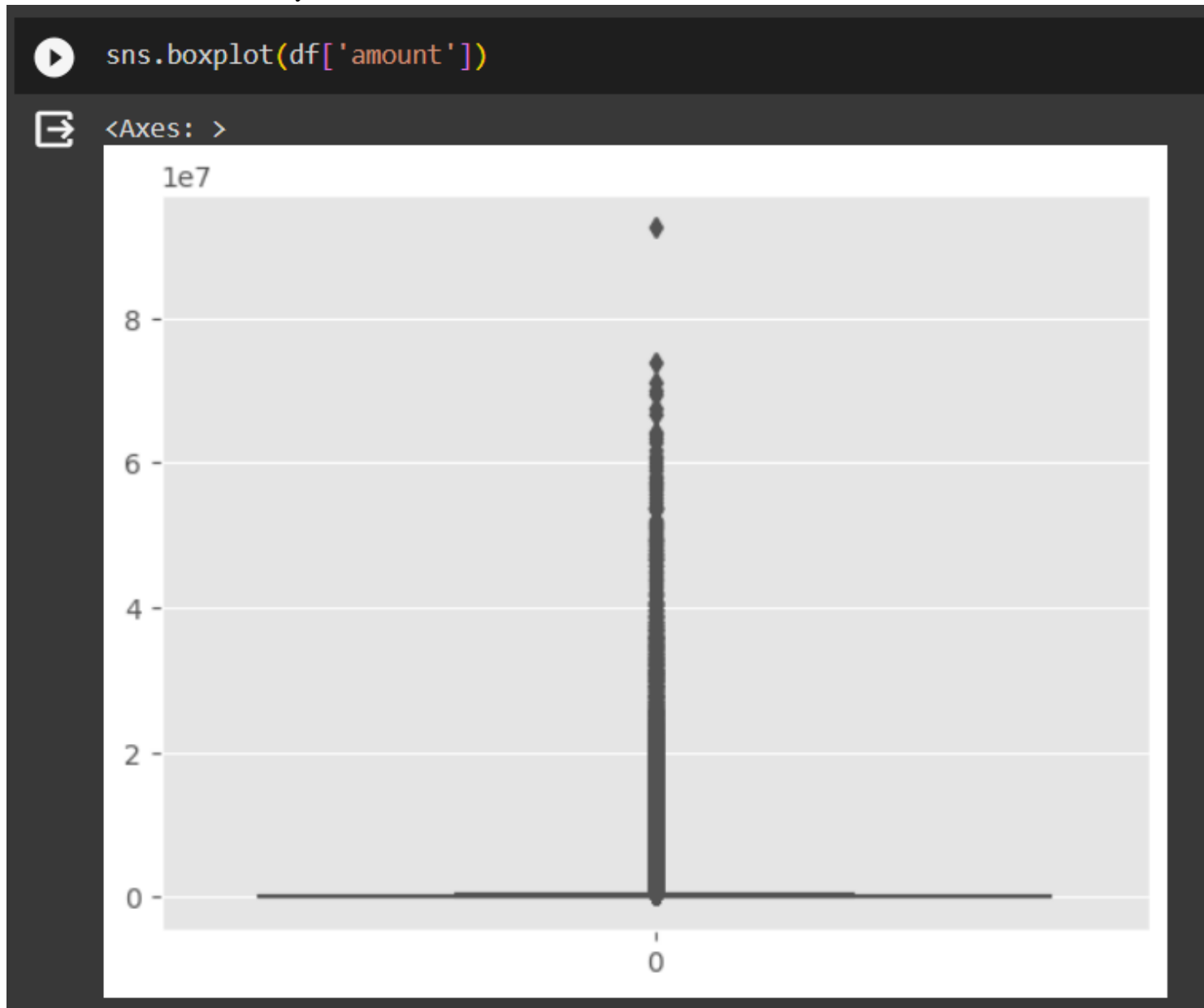
step      0
type      0
amount    0
oldbalanceOrg  0
newbalanceOrig  0
oldbalanceDest  0
newbalanceDest  0
isFraud   0
dtype: int64
```

Let's look for any outliers in the dataset.

Activity 2: Handling outliers

With the help of boxplot, outliers are visualized. And here we are going to find upper bound and lower bound of all features with some mathematical formula.

- From the below diagram, we could visualize that Monetary feature has outliers. Boxplot from seaborn library is used here.



Activity 3: Splitting data into train and test

Now let's split the Dataset into train and test sets. First split the dataset into x and y and then split the data set

Here x and y variables are created. On x variable, df is passed with dropping the target variable. And on y target variable is passed. For splitting training and testing data we are using train_test_split() function from sklearn. As parameters, we are passing x, y, test_size, random_state.

```
[ ] from sklearn.model_selection import train_test_split

x_train,x_test,y_train,y_test=train_test_split(x,y,random_state=0,test_size=0.2)
print(x_train.shape)
print(x_test.shape)
print(y_test.shape)
print(y_train.shape)

(5090096, 7)
(1272524, 7)
(1272524,)
(5090096,)
```

Milestone 4: Model Building

Now our data is cleaned and it's time to build the model. We can train our data on different algorithms. For this project we are applying four classification algorithms. The best model is saved based on its performance.

Activity 1: Decision Tree Classifier

Decision Tree Classifier algorithm is initialized and training data is passed to the model with .fit() function. Test data is predicted with .predict() function and saved in new variable. For evaluating the model, confusion matrix and classification report is do

```
[ ] from sklearn.tree import DecisionTreeClassifier
    dtc=DecisionTreeClassifier()
    dtc.fit(x_train, y_train)

    y_test_predict2=dtc.predict(x_test)
    test_accuracy=accuracy_score(y_test,y_test_predict2)
    test_accuracy

0.999704524236871
```



```
[ ] y_train_predict2=dtc.predict(x_train)
    train_accuracy=accuracy_score(y_train,y_train_predict2)
    train_accuracy
```

1.0

```
[ ] pd.crosstab(y_test,y_test_predict2)
```

col_0	is Fraud	is not Fraud
isFraud		
is Fraud	1445	196
is not Fraud	180	1270703

```
▶ print(classification_report(y_test,y_test_predict2))
```

	precision	recall	f1-score	support
is Fraud	0.89	0.88	0.88	1641
is not Fraud	1.00	1.00	1.00	1270883
accuracy			1.00	1272524
macro avg	0.94	0.94	0.94	1272524
weighted avg	1.00	1.00	1.00	1272524

Activity 2: Extra Trees Classifier model

```
[ ] from sklearn.ensemble import ExtraTreesClassifier
    etc=ExtraTreesClassifier()
    etc.fit(x_train,y_train)

    y_test_predict3=etc.predict(x_test)
    test_accuracy=accuracy_score(y_test,y_test_predict3)
    test_accuracy
```

0.9996990233583021

```
[ ] y_train_predict3=etc.predict(x_train)
    train_accuracy=accuracy_score(y_train,y_train_predict3)
    train_accuracy
```

1.0

```
[ ] pd.crosstab(y_test,y_test_predict3)
```

col_0 is Fraud is not Fraud		
isFraud		
is Fraud	1274	367
is not Fraud	16	1270867

```
▶ print(classification_report(y_test,y_test_predict3))
```

	precision	recall	f1-score	support
is Fraud	0.99	0.78	0.87	1641
is not Fraud	1.00	1.00	1.00	1270883
accuracy			1.00	1272524
macro avg	0.99	0.89	0.93	1272524
weighted avg	1.00	1.00	1.00	1272524

Activity 3: Evaluating performance of the model

```
[ ] from sklearn.preprocessing import LabelEncoder
```

```
la = LabelEncoder()  
y_train1 = la.fit_transform(y_train)
```

```
[ ] y_test1=la.transform(y_test)  
y_test1
```


```
array([1, 1, 1, ..., 1, 1, 1])
```

```
[ ] y_train1
```

```
array([1, 1, 1, ..., 1, 1, 1])
```

```
[ ] def compareModel():  
    print("train accuracy for dtc",accuracy_score(y_train_predict2,y_train))  
    print("test accuracy for dtc",accuracy_score(y_test_predict2,y_test))  
    print("train accuracy for etc",accuracy_score(y_train_predict3,y_train))  
    print("test accuracy for etc",accuracy_score(y_test_predict3,y_test))
```

 compareModel()

 train accuracy for dtc 1.0
test accuracy for dtc 0.999704524236871
train accuracy for etc 1.0
test accuracy for etc 0.9996990233583021

Activity 4: Predict the chance of donation according to given factors.

```
[ ] y_train_predict3=etc.predict(x_train)  
train_accuracy=accuracy_score(y_train,y_train_predict3)  
train_accuracy
```

```
1.0
```

Our model is performing well. So, we are saving the model by pickle.dump().

```
import pickle

pickle.dump(DT_model,open('Car prediction.pkl','wb'))
```

Milestone 5: Application Building

In this section, we will be building a web application that is integrated to the model we built. A UI is provided for the uses where he has to enter the values for predictions. The enter values are given to the saved model and prediction is showcased on the UI.

This section has the following tasks

- Building HTML Pages
- Building server side script

Activity1: Building Html Pages:

For this project create HTML file namely

- home.html save it in Templates folder.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Online Payments Fraud Detection</title>
  <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>

  <header>
    <h1>Online Payments Fraud Detection</h1>
  </header>

  <button id="home-button">Home</button>
  <a
href="predict.html">
<button id="predict-button">Predict</button>
</a>

  <main>
    <p>The objective of this article is to predict online payments fraud given the various parameters. This will be a classification problem
  </main>

</body>
</html>
```

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Online Payments Fraud Detection - Prediction Input</title>
  <link rel="stylesheet" type="text/css" href="style1.css">
</head>
<body>

  <header>
    <h1>Online Payments Fraud Detection </h1>
  </header>

  <main>
    <form>
      <label for="step">Step:</label>
      <input type="number" id="step" name="step" required><br>

      <label for="type">Type:</label>
      <input type="number" id="type" name="type" required><br>

      <label for="amount">Amount:</label>
      <input type="number" id="amount" name="amount" required><br>

      <label for="oldbalanceOrig">OldbalanceOrig:</label>
      <input type="number" id="oldbalanceOrig" name="oldbalanceOrig" required><br>

      <label for="newbalanceOrig">NewbalanceOrig:</label>
      <input type="number" id="newbalanceOrig" name="newbalanceOrig" required><br>

      <label for="oldbalanceDest">Oldbalance Dest:</label>
      <input type="number" id="oldbalanceDest" name="oldbalanceDest" required><br>

      <label for="newbalanceDest">NewbalanceDest:</label>
      <input type="number" id="newbalanceDest" name="newbalanceDest" required><br>

      <button type="submit">Submit</button>
    </form><br>

    <form action="result.html" method="GET">
      <button type="submit">Submit</button>
    </form>
  </main>
</body>
</html>
```

Activity 2: Build Python code:

Import the libraries

```
import numpy as np
from flask import Flask, request, render_template
import pickle
```

Load the saved model. Importing flask module in the project is mandatory. An object of Flask class is our WSGI application. Flask constructor takes the name of the current module (`__name__`) as argument.

```
app = Flask(__name__)

model = pickle.load(open('Car prediction.pkl','rb'))
```

Render HTML page:

```
@app.route('/')
def start():
    return render_template('index1.html')

@app.route('/login',methods =["POST","GET"])
def login():

    if request.method == "POST":
        age = request.form["age"]
        annual_income = request.form["annualincome"]
        gender = request.form["gender"]
```

Here we will be using declared constructor to route to the HTML page which we have created earlier.

In the above example, '/' URL is bound with home.html function. Hence, when the home page of the web server is opened in browser, the html page will be rendered. Whenever you enter the values from the html page the values can be retrieved using POST Method.

Retrieves the value from UI:

Here we are routing our app to predict() function. This function retrieves all the values from the

HTML page using Post request. That is stored in an array. This array is passed to the model.predict() function. This function returns the prediction. And this prediction value will be rendered to the text that we have mentioned in the carprediction.html page earlier.

```
t = np.array([[age,annual_income,gender]])

t_scaled= scale.fit_transform(t)
output =model.predict(t_scaled)
print(output)

if (output == 1):
    return render_template('index1.html', output="Purchasble")
else:
    return render_template('index1.html', output="Not purchasble")
    return render_template("index1.html")

return render_template("index1.html")
```

Main Function:

```
if __name__ == '__main__':
    app.run(debug=True)
```

Activity 3: Run the application

Open Visual studio code and Import all the project folders.

When you run the app.py file and click on the server url in terminal, you will be redirected to home page. The home page will look like:

Online Payments Fraud Detection

[Home](#)[Predict](#)

The objective of this article is to predict online payments fraud given the various parameters. This will be a classification problem since the target or dependent variable is the fraud (categorical values). The purpose of fraud of online payments is to separate the available supply of portable online payments into classes differing in superiority. We will be using classification algorithms such as Decision tree, Random forest, SVM, and Extra tree classifier. We will train and test the data with these algorithms.

Prediction page:

Online Payments Fraud Detection

• Step:

Type:

Amount:

OldbalanceOrg:

NewbalanceOrig:

Oldbalance Dest:

NewbalanceDest:

Submit

• Submit

Online Payments Fraud Detection

The predicted fraud for the online payment is:

Conclusion:

In the realm of online payments, the integration of machine learning for fraud detection represents a crucial stride towards securing digital transactions. By delineating the problem scope, engaging stakeholders, and establishing clear success metrics, we lay the groundwork for a solution adept at identifying and preventing various forms of fraud, from credit card scams to identity theft. The dynamic and evolving nature of online fraud necessitates a solution that can adapt swiftly, striking a delicate balance between accuracy and efficiency to assure not only the integrity of financial transactions but also a seamless user experience.

Collaboration with key stakeholders, including financial institutions and payment service providers, ensures that the developed machine learning solution aligns with industry standards and user expectations. As we navigate the implementation of sophisticated algorithms and leverage diverse data sources, the commitment to transparency, ethical considerations, and legal compliance remains paramount. Ultimately, the pursuit of excellence in online payments fraud detection using machine learning is not merely a technological advancement; it's a holistic approach that aims to fortify the foundations of digital transactions, instilling confidence among users and contributing to the resilience and security of the broader digital economy.