

# **Project Report Format:**

## **1. INTRODUCTION**

1.1 Project Overview

1.2 Purpose

## **2. LITERATURE SURVEY**

2.1 Existing problem

2.2 References

2.3 Problem Statement Definition

## **3. IDEATION & PROPOSED SOLUTION**

3.1 Empathy Map Canvas

3.2 Ideation & Brainstorming

## **4. REQUIREMENT ANALYSIS**

4.1 Functional requirement

4.2 Non-Functional requirements

## **5. PROJECT DESIGN**

5.1 Data Flow Diagrams & User Stories

5.2 Solution Architecture

## **6. PROJECT PLANNING & SCHEDULING**

6.1 Technical Architecture

6.2 Sprint Planning & Estimation

6.3 Sprint Delivery Schedule

## **7. CODING & SOLUTIONING**

7.1 Feature 1

7.2 Feature 2

## **8. PERFORMANCE TESTING**

8.1 Performance Metrics

## **9. RESULTS**

9.1 Output Screenshots

## **10. ADVANTAGES & DISADVANTAGES**

10.1 Advantages

10.2 DisAdvantages

## **11. CONCLUSION**

## **12. FUTURE SCOPE**

## **13. APPENDIX**

Source Code

GitHub & Project Demo Link

# **1. INTRODUCTION:**

## **1.1. Project Overview:**

Our project on Online fraud detection using machine learning in a multinational financial services setting aims to develop a robust and proactive system to identify and prevent fraudulent activities across diverse sectors, including investment banking, pension management, asset management, and payment services. This initiative recognizes the critical importance of safeguarding financial transactions and assets in the face of evolving cyber threats and fraudulent schemes.

## **1.2. Purpose:**

The purpose of the project is to proactively address the inherent risks associated with fraudulent activities in a multinational financial services. When businesses are involved, it causes some issues; occasionally, they must issue refunds incase of any fraud happens, in order to keep goodwill near customers, therefore it is a financial loss for the company, So it is crucial that both consumers and businesses are aware of these internet scams.

A model is to be made using machine learning to determine if an online payment is fraudulent or not, some features like the type of payment, the recipient's identity, etc, to enhance security, maintain regulatory compliance, and preserve customer trust while optimizing operational efficiency.

## **2. LITERATURE SURVEY:**

### **2.1. Existing problem:**

When it comes to the simplicity of making a payment while sitting anywhere in the world, online payments have been a source of attractiveness. Over the past decade, there has been an increase in online payments. E-payments enable businesses earn a lot of money in addition to consumers. However, because electronic payments are so simple, there is also a risk of fraud associated with them.

### **2.2. References:**

<https://www.geeksforgeeks.org/online-payment-fraud-detection-using-machine-learning-in-python/>

<https://www.slideshare.net/ANILKUMARBATTINA/online-payment-fraud-detection-using-machine-learning-modelpdf>

**Dataset:-** <https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud>

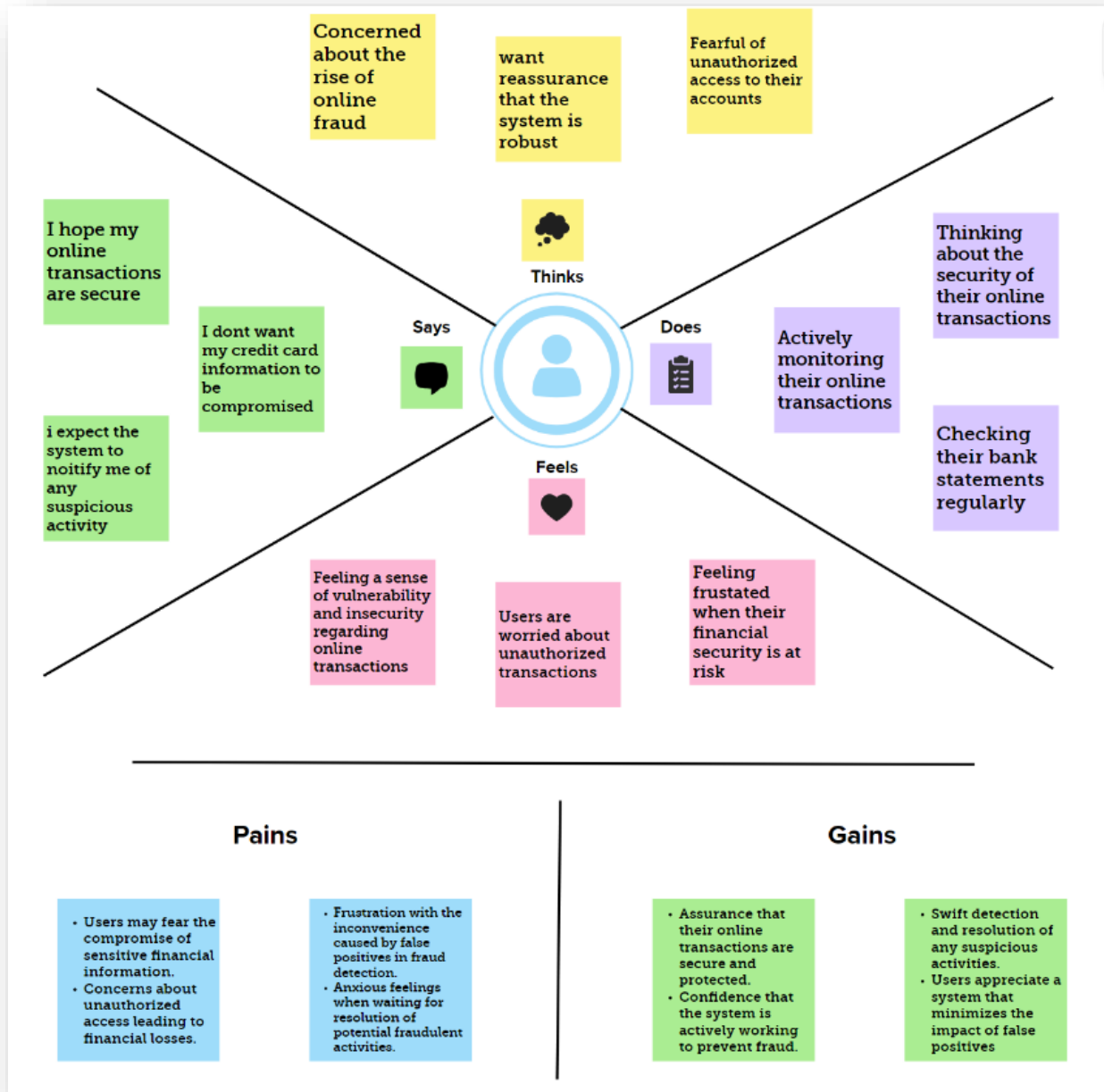
### **2.3. Problem Statement Definition:**

In the global financial services sector, there's a growing challenge with sophisticated fraud affecting investment banking, pension management, asset management, and payment services. Current fraud detection methods struggle to keep up with these evolving tactics. Problems include the need for real-time detection, the complexity of different financial services, meeting regulatory requirements, integrating with existing systems, and staying ahead of new fraud methods. This project aims to solve these issues by using machine learning for better fraud detection. The goal is to improve security, ensure compliance, maintain customer trust, and streamline operations across all financial services provided by the multinational institution.

### 3. IDEATION & PROPOSED SOLUTION:

#### 3.1. Empathy Map Canvas:

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes.



### 3.2. Ideation & Brainstorming:

	Lahari	Yasmitha	Tejeswar
★	We need to explore and implement a variety of machine learning algorithms	Regularly update models based on feedback and evolving fraud patterns.	Develop user-friendly educational materials to raise awareness about common fraud schemes and preventive measures.
	Implement real-time monitoring for instant detection of suspicious activities	Establish communication channels between different sectors to share fraud intelligence.	Stay informed about data protection regulations and ensure compliance.
	Collaborate with legal experts to navigate the regulatory landscape	★ Consider incorporating various transaction features like amount, frequency, location, and time	★ Design the system to scale effectively with the growing user base and transaction volume.
	Introduce gamified elements to encourage users and employees to actively participate in security practices.	Use trend analysis to proactively update fraud detection models and stay ahead of evolving tactics.	Implement features that automatically adjust to changes in compliance requirements.

Top priority →	We need to explore and implement a variety of machine learning algorithms	Consider incorporating various transaction features like amount, frequency, location, and time	Design the system to scale effectively with the growing user base and transaction volume.	Implement real-time monitoring for instant detection of suspicious activities
Medium priority →	Develop user-friendly educational materials to raise awareness about common fraud schemes and preventive measures.	Stay informed about data protection regulations and ensure compliance.	Collaborate with legal experts to navigate the regulatory landscape	Regularly update models based on feedback and evolving fraud patterns.
low priority →	Introduce gamified elements to encourage users and employees to actively participate in security practices.	Use trend analysis to proactively update fraud detection models and stay ahead of evolving tactics.	Establish communication channels between different sectors to share fraud intelligence.	Implement features that automatically adjust to changes in compliance requirements.

## **4. REQUIREMENT ANALYSIS:**

### **4.1. Functional requirement:**

#### **1) Data Pre-Processing:**

handling missing values, removing rows with missing values, imputing missing values with statistical measures (such as : mean, median, mode)

**Outlier Detection:** Performing exploratory data analysis to identify potential outliers in the dataset and handling outliers such as removing them, transforming them, or treating them separately in the analysis.

**Data Cleaning:** Identifying and address any inconsistencies or errors in the data such as standardizing data formats, correct typos, and reconcile discrepancies to ensure data consistency.

#### **2) Normalization and Standardization:**

Normalizing numerical features to bring them to a similar scale and standardizing features by subtracting the mean and dividing by the standard deviation, ensuring that different features are comparable.

#### **3) Feature Engineering:**

Transaction frequency, amount, time duration, type, and customer behavior collectively form key indicators in fraud detection. Unusually high transaction frequency or large deviations in amounts can raise suspicion. Abnormal durations for transactions, specific transaction types, and deviations from established customer behavior patterns are potential signals of fraud. Analyzing these factors collectively provides a comprehensive approach to identifying and addressing fraudulent activities in various transaction scenarios.

#### **4) Machine Learning Models:**

Four machine learning algorithms were used to train and test the dataset after which they were evaluated using different metrics for best performance and deployment. We selected our models and target variable. The algorithms used were Random Forest and K-Nearest Neighbors. Information's about the transactions carried out by different customers in the bank were recorded in rows and columns. The data was collected, processed, and statistical analysis was done on the dataset. Data verification was done where we checked if we had any missing values and also checked data type.

#### **5) Real time monitoring:**

The online fraud detection system needs to watch transactions in real-time. It sets limits, like how much money in a transaction is too much, or if there are too many transactions happening really quickly. Unusually short or long transaction times and sudden increases in transaction speed are also things to keep an eye on. These limits help the system catch potential fraud and deal with it quickly. Overall, it helps the system do a good job in protecting against tricky activities.

## **4.2. Non-Functional requirements:**

### **1) Performance:**

The system processes transactions in real-time, promptly generates alerts for suspicious activities, and ensures timely responses. It minimizes latency in data retrieval, model inference, and user authentication for quick decision-making on blocking fraudulent transactions. Load balancing evens out workloads, and low network latency ensures effective communication. The system maintains swift threshold evaluations, facilitates rapid dashboard generation, and ensures timely customer notifications.

### **2) Scalability:**

The system is designed for scalable growth, seamlessly handling increased workloads and concurrent users without performance compromise. Scalability extends to data storage, computational resources, and machine learning model scaling for efficient fraud detection. Regular scalability testing ensures readiness for future transaction volume increases, contributing to the system's agility, resilience, and sustained effectiveness in preventing online fraud.

### **3) Reliability:**

We need to Establish requirements for system reliability and minimize single points of failure.

### **4) Availability:**

The online fraud detection system is designed for continuous 24/7 operation with rigorous availability requirements. This includes high uptime percentages, defined downtime allowances for maintenance, redundancy mechanisms, load balancing, and geographic redundancies. Robust monitoring, swift incident response, and scalability for peak loads ensure uninterrupted services, supported by comprehensive backup and recovery procedures. The system's resilience is further enhanced by regular maintenance windows, network availability guaranteeing seamless fraud detection services in the dynamic financial services landscape.

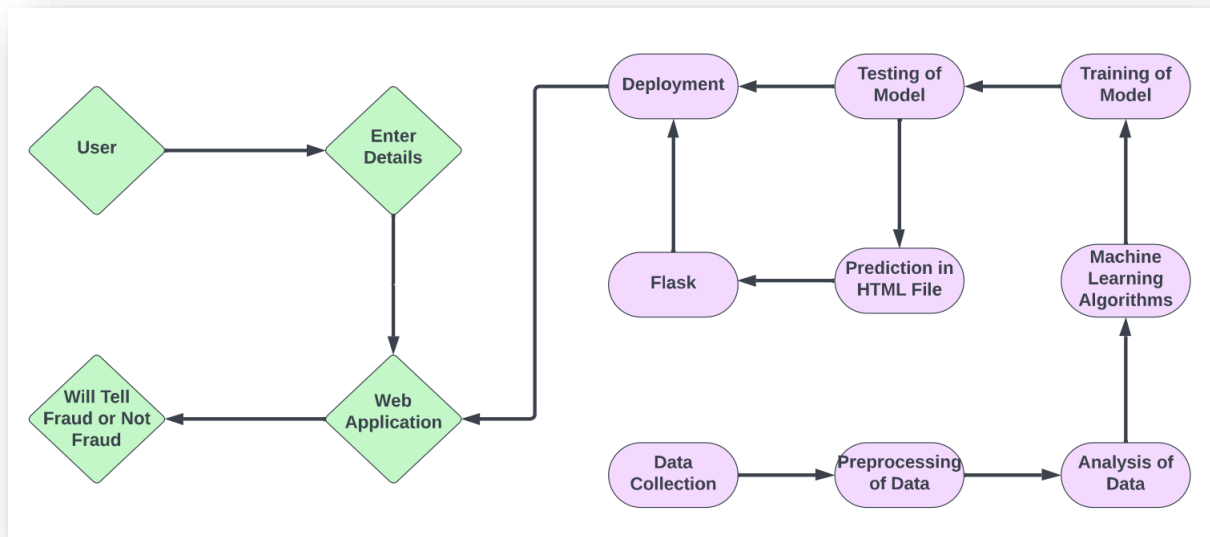
### **5) Security:**

The online fraud detection system emphasizes security through end-to-end encryption, secure APIs, and strict access controls. Robust authentication, secure storage, and ongoing monitoring protect against unauthorized access. Regular audits, incident response planning, and security training further bolster the system's resilience, ensuring continuous protection for sensitive financial data.

## 5. PROJECT DESIGN:

### 5.1. Data Flow Diagrams & User Stories:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

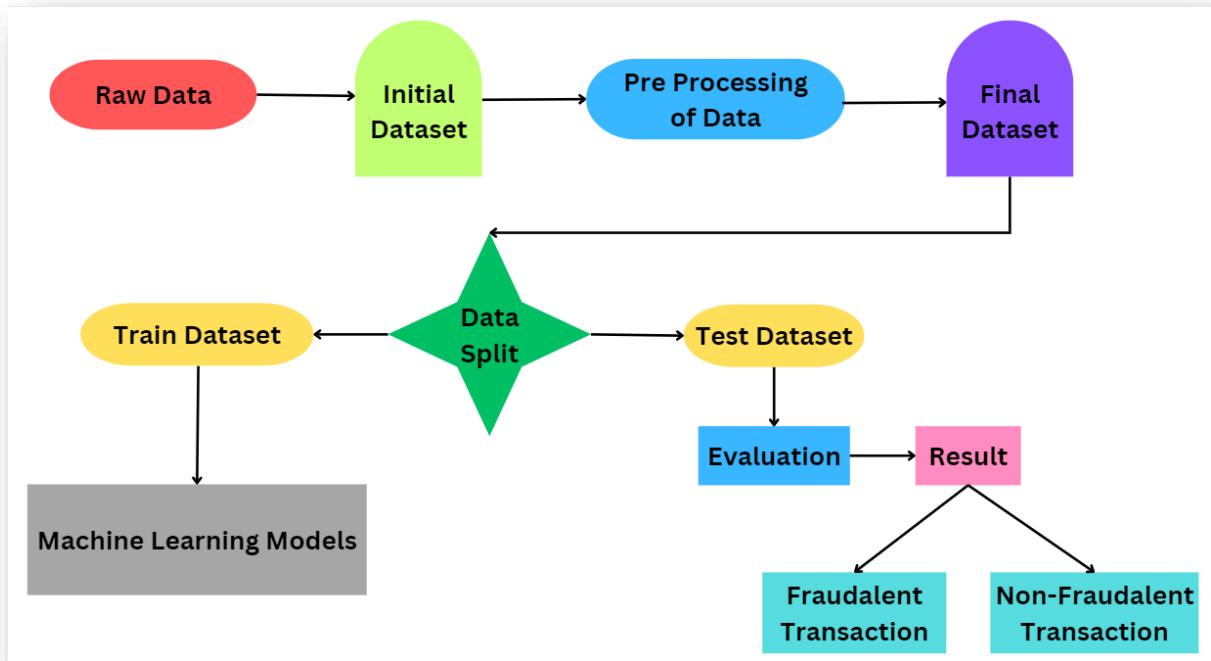




## User Stories:

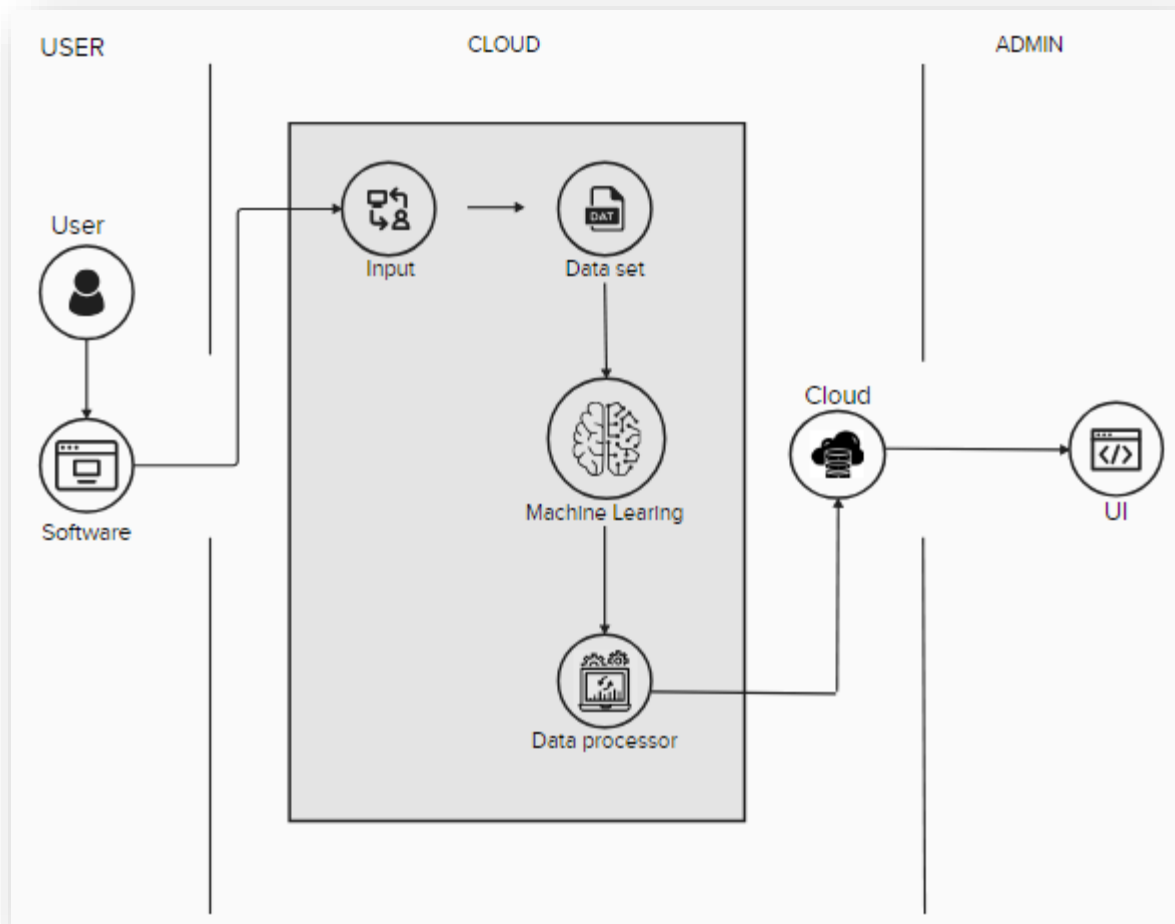
User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Retail banking customer	Real-time alerts	USN-1	I want to receive instant notifications on my mobile device if any suspicious activity is detected on my account.	The system sends real-time alerts for transactions that deviate from my typical spending patterns.	High	Sprint-1
Investment banking client	Real time monitoring	USN-2	I want the investment platform to automatically trigger alerts if there are unusual trading patterns or unexpected changes in my portfolio.	Receive immediate alerts for any high-risk trading activities or portfolio adjustments.	High	Sprint-1
Pension plan participant	Detects irregularities	USN-3	I want the pension management system to alert me if there are irregularities in my contribution or disbursement history.	Receive notifications for any unexpected changes in pension contributions or withdrawals.	Medium	Sprint-2
Payment services user	Detects unauthorized transactions	USN-4	I want the payment system to detect and prevent unauthorized transactions, especially for online and mobile payments.	Experience seamless authentication processes and receive alerts for any suspicious transactions.	Medium	Sprint-2
System administrator for the financial services group	Regular software updates	USN-5	I want to ensure that the fraud detection software is regularly updated to adapt to emerging fraud patterns and maintain optimal performance.	Receive regular updates on software enhancements, and have the ability to schedule and implement updates seamlessly.	Medium	Sprint-2

## 5.2. Solution Architecture:



## 6. PROJECT PLANNING & SCHEDULING:

### 6.1. Technical Architecture:



## 6.2. Sprint Planning & Estimation:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Real-time alerts	USN-1	I want to receive instant notifications on my mobile device if any suspicious activity is detected on my account.	2	Medium	Lahari
Sprint-1	Real time monitoring	USN-2	I want the investment platform to automatically trigger alerts if there are unusual trading patterns or unexpected changes in my portfolio.	1	High	Yasmitha
Sprint-2	Detects irregularities	USN-3	I want the pension management system to alert me if there are irregularities in my contribution or disbursement history.	2	Low	Tejeswar
Sprint-3	Detects unauthorized transactions	USN-4	I want the payment system to detect and prevent unauthorized transactions, especially for online and mobile payments.	2	Medium	Yasmitha
Sprint-4	Regular software updates	USN-5	I want to ensure that the fraud detection software is regularly updated to adapt to emerging fraud patterns and maintain optimal performance.	1	High	Tejeswar

### 6.3. Sprint Delivery Schedule:

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	3	6 Days	23 Oct 2023	28 Oct 2023	3	28 Oct 2023
Sprint-2	5	6 Days	30 Oct 2023	04 Nov 2023	5	04 Nov 2023
Sprint-3	7	6 Days	06 Nov 2023	11 Nov 2023	7	11 Nov 2023
Sprint-4	5	6 Days	13 Nov 2023	1 Nov 2023	5	18 Nov 2023

#### Velocity:

We have a 24 day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{\text{sprint duration}}{\text{velocity}}$$

$$AV = 24/20 = 1.2$$

## 7. CODING & SOLUTIONING:

### 7.1. Feature 1: Detecting The Online Fraud

The first feature is detecting the online banking frauds. The Flask web application takes user input for Key paramenters like type\_code, amount, oldbalanceDest,newbalanceDest, step, oldbalanceOrg, newbalanceOrg. The model, loaded from 'Model.pkl' predicts the resource allocation using Algorithms. The result is displayed on the web page.

#### Code Implementation:

```
1  import numpy as np
2  from flask import Flask, request, jsonify, render_template
3  import pickle
4
5  app = Flask(__name__)
6  # prediction function
7  model = pickle.load(open("model.pkl", "rb"))
8  def ValuePredictor(to_predict_list):
9      to_predict = np.array(to_predict_list).reshape(1, 7)
10     loaded_model = pickle.load(open("model.pkl", "rb"))
11     result = loaded_model.predict(to_predict)
12     print(result)
13     if int(result)== 1:
14         prediction = 'Given transaction is fradulent'
15     else:
16         prediction = 'Given transaction is NOT fradulent'
17     print(prediction)
18     return int(result)
19
20 #ValuePredictor([-9,0.44,1.41,-0.25,-0.7,-1.4,-0.2])
21 @app.route('/')
22 def home():
23     return render_template("index.html")
24
25 @app.route('/predict',methods=['POST','GET'])
26 def predict():
27
28     int_features = [float(x) for x in request.form.values()]
29     print(int_features)
30     prediction=ValuePredictor(int_features)
31     if prediction == 0:
32         val = 'Given transaction is fradulent'
33         return render_template('index.html',
34                                prediction_text=val.format(
35                                    val),
36                                )
37     else:
38         val='Given transaction is NOT fradulent'
39         return render_template('index.html',
40                                prediction_text=val.format(
41                                    val),
42                                )
43
44 if __name__ == "__main__":
45     app.run(debug=False)
```

## 7.2. Feature 2: User Friendly Interface

The second feature mainly focuses on creating a user friendly interface by using HTML Styling. The provided HTML code includes styling rules to enhance the visual appeal of web application. And mainly focused on Very Simple User Interface. This future contributes a positive user experience when interacting with the application.

### Code Implementation:

```
1  input,
2  button {
3      position: fixed;
4      top: 50%;
5      left: 50%;
6      -webkit-transform: translate(-50%, -300%);
7      transform: translate(-50%, -300%);
8      display: block;
9      width: 70vw;
10     opacity: 0;
11     pointer-events: none;
12     -webkit-transition: all 0.5s cubic-bezier(0.4, 0.25, 0.8, 0.3);
13     transition: all 0.5s cubic-bezier(0.4, 0.25, 0.8, 0.3);
14 }
15
16 input {
17     padding: .25rem 0;
18     border: 0;
19     border-bottom: 1px solid #bb1515;
20     outline: 0;
21     background: transparent;
22     color: #fff;
23     font-size: 3rem;
24     line-height: 4rem;
25     letter-spacing: .125rem;
26     -webkit-transition: all 0.5s cubic-bezier(0.4, 0.25, 0.8, 0.3);
27     transition: all 0.5s cubic-bezier(0.4, 0.25, 0.8, 0.3);
28
29
203 label[for="input-5"] .nav-dot {
204     margin-top: -25px;
205 }
206 label[for="input-6"] .nav-dot {
207     margin-top: 0px;
208 }
209 label[for="input-7"] .nav-dot {
210     margin-top: 25px;
211 }
212
213
214 * {
215     margin: 0;
216     padding: 0;
217     box-sizing: border-box;
218 }
219
220 html, body {
221     width: 100%;
222     height: 100%;
223     background-image: -webkit-gradient(linear, left top, right bottom, from(#111E25), to(#111));
224     background-image: linear-gradient(to bottom right, #111E25 0%, #111 100%);
225     font-family: 'Lato', sans-serif;
226 }
227
228 form {
229     width: 100%;
230     height: 100%;
231     overflow: hidden;
232 }
```

## 8. PERFORMANCE TESTING:

### 8.1. Performace Metrics:

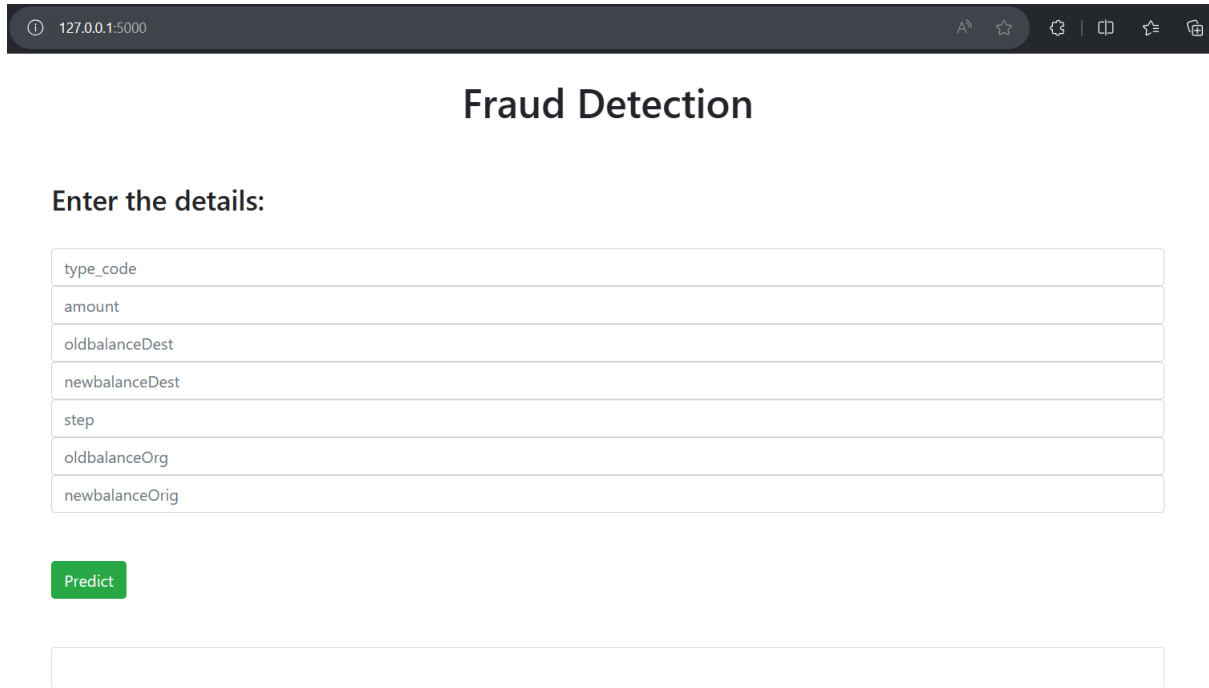
S.No.	Parameter	Values	Screenshot																														
1.	Metrics	<b>Regression Model:</b> MAE, MSE, RMSE, R2 score  <b>Classification Model:</b> Confusion Matrix, Accuracy Score & Classification Report	<div>Mean Absolute Error: 0.1248462659562 Mean Squared Error: 0.1248462659562 Root Mean Squared Error: 0.387745996988</div> <div>Scores(R^2): [0.1478543 0.14750456 0.1761771 0.1561489 0.1110178]</div> <div>Accuracy:97.06%</div> <div><table><tr><td></td><td>precision</td><td>recall</td><td>f1-score</td><td>support</td></tr><tr><td>0.0</td><td>0.97</td><td>0.98</td><td>0.98</td><td>623</td></tr><tr><td>1.0</td><td>0.98</td><td>0.97</td><td>0.97</td><td>577</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.97</td><td>1200</td></tr><tr><td>macro avg</td><td>0.97</td><td>0.97</td><td>0.97</td><td>1200</td></tr><tr><td>weighted avg</td><td>0.97</td><td>0.97</td><td>0.97</td><td>1200</td></tr></table></div> <div><pre>print("confusion")</pre><div><pre>[630 15] [540 410]</pre></div></div>		precision	recall	f1-score	support	0.0	0.97	0.98	0.98	623	1.0	0.98	0.97	0.97	577	accuracy			0.97	1200	macro avg	0.97	0.97	0.97	1200	weighted avg	0.97	0.97	0.97	1200
	precision	recall	f1-score	support																													
0.0	0.97	0.98	0.98	623																													
1.0	0.98	0.97	0.97	577																													
accuracy			0.97	1200																													
macro avg	0.97	0.97	0.97	1200																													
weighted avg	0.97	0.97	0.97	1200																													
2.	Tune the Model	Hyperparameter Tuning - Validation Method	<div>Cross-Vadidation Scores (R^2): [[0.1478543 0.14750456 0.1761771 0.1561489 0.1110178] Mean R^2 Score:0.64988842</div>																														



## 9. RESULTS:

### 9.1. Output Screenshots:

We need to enter the Details



127.0.0.1:5000

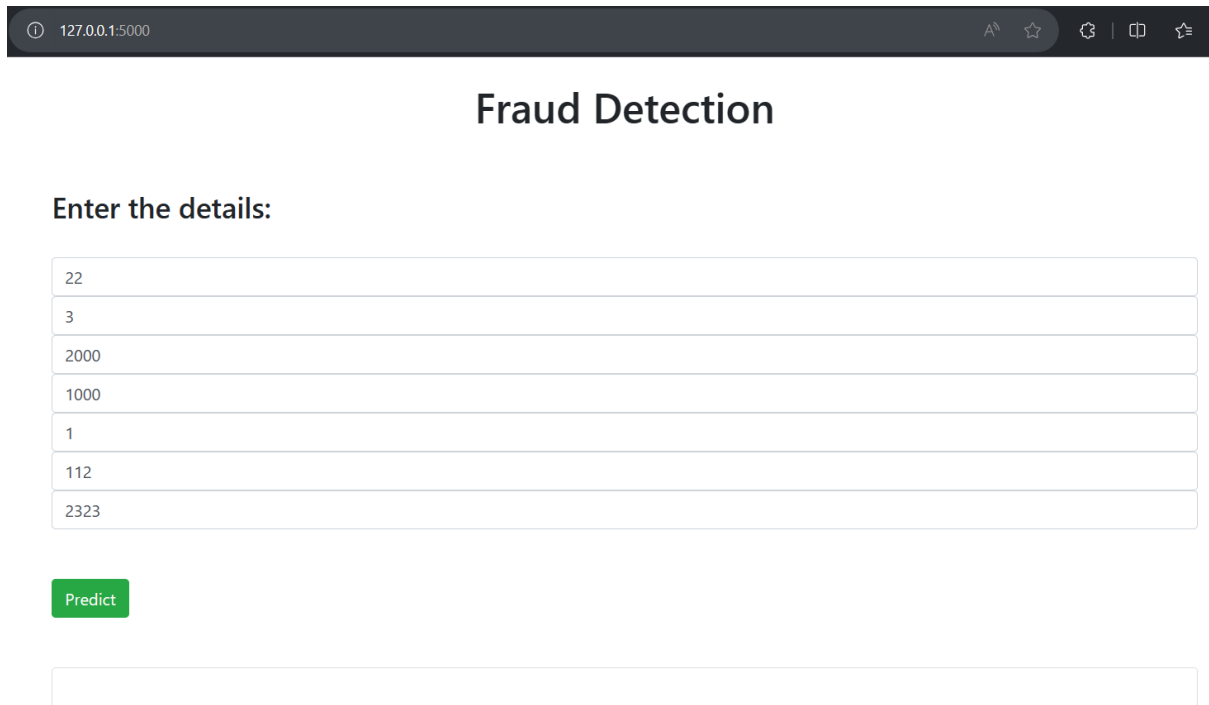
### Fraud Detection

Enter the details:

type_code
amount
oldbalanceDest
newbalanceDest
step
oldbalanceOrg
newbalanceOrig

Predict

After Entering we need to Click on Predict Button



127.0.0.1:5000

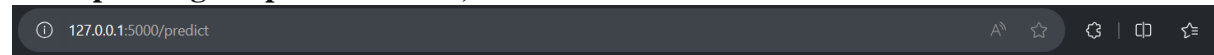
### Fraud Detection

Enter the details:

22
3
2000
1000
1
112
2323

Predict

**After pressing the predict button, the result will show as Fradulent or Not Fradulent**



## Fraud Detection

Enter the details:

type_code
amount
oldbalanceDest
newbalanceDest
step
oldbalanceOrg
newbalanceOrig

Predict

Given transaction is NOT fradulent

## 10. ADVANTAGES & DISADVANTAGES:

### 10.1. Advantages:

- **Increased Accuracy:** Machine learning algorithms can analyze vast amounts of data quickly and accurately, enhancing the accuracy of fraud detection compared to traditional methods.
- **Real-time Detection:** Machine learning enables real-time monitoring and detection of fraudulent activities, allowing for swift intervention and prevention
- **Adaptability to Evolving Threats:** Machine learning models can adapt to new and evolving fraud patterns, providing a proactive defense against emerging threats.
- **Reduced False Positives:** Advanced machine learning algorithms can reduce false positives by better distinguishing between normal and fraudulent behavior, minimizing the impact on legitimate transactions.
- **Used in multinational financial services:** Online fraud detection in multifinancial services offers key advantages, including timely prevention of fraud, heightened security, improved customer trust, regulatory compliance, operational efficiency, reduced false positives, adaptability to evolving threats, and efficient risk management. This ensures a secure and resilient financial environment for institutions and clients alike.

### 10.2. DisAdvantages:

- **Data Privacy Concerns:** Handling sensitive financial data raises concerns about data privacy and security, requiring stringent measures to comply with regulations and protect customer information.
- **High Initial Costs:** Implementing machine learning systems involves significant upfront costs for technology, infrastructure, and training, which may be a barrier for some organizations.
- **Complex Implementation:** Deploying machine learning solutions requires specialized knowledge and expertise, making implementation complex and potentially challenging for organizations without a strong technical background.
- **Over-reliance on Technology:** Depending solely on machine learning for fraud detection may lead to over-reliance on technology, potentially overlooking human insights and intuition.
- **Potential Bias:** Machine learning models may inadvertently inherit biases present in the training data, leading to discriminatory outcomes. Addressing bias requires ongoing monitoring and adjustment.

## **11. CONCLUSION:**

Implementing a project on online fraud detection using machine learning in multinational financial services, encompassing investment banking, pension management, asset management, and payment services, is imperative for fostering a secure, efficient, and trustworthy financial environment. The utilization of advanced machine learning algorithms offers real-time detection, adaptability to evolving threats, and a reduction in false positives, enhancing the overall accuracy and effectiveness of fraud prevention measures. This project not only mitigates financial risks and ensures regulatory compliance but also strengthens customer trust through the protection of assets and timely intervention against fraudulent activities. By embracing the advantages of machine learning in fraud detection, financial institutions can proactively safeguard their operations and maintain the integrity of financial services in the face of evolving and complex challenges.

## **12. FUTURE SCOPE:**

Future work for the project on online fraud detection in multinational financial services includes continuous improvement of machine learning models, exploration of advanced technologies like AI and deep learning, incorporation of behavioral analytics, extending fraud detection across multiple channels, implementation of explainable AI for model interpretability, fostering collaborative intelligence among financial institutions, dynamic risk scoring based on real-time assessments, integration of blockchain technology, geospatial analysis for transaction context, enhancement of user authentication methods, continual adaptation to regulatory requirements, and a focus on continuous training and skill development. These efforts aim to fortify the system against evolving fraud patterns, ensuring its resilience and effectiveness in safeguarding financial transactions across diverse services.

### 13. APPENDIX:

**Source Code:-** Uploaded in Github (Phase 4 – Project Development)

<https://github.com/smartinternz02/SI-GuidedProject-608897-1697944099/tree/main/Phase%20-%20Project%20Development/Fraud-Detection-ML-WebApp-master>

**GitHub Link:-** <https://github.com/smartinternz02/SI-GuidedProject-608897-1697944099>

**Project Demo Link:-** <https://drive.google.com/drive/folders/1p-6er0I71KjRSPaEpn19qOVLQwrqUBT?usp=sharing>