Project Report

1. INTRODUCTION

1.1 Project Overview

The digital landscape has revolutionized the convenience of online payments, but it has also led to a surge of fraudulent activities targeting these systems. To counteract these threats, a project is being developed to fortify the security infrastructure of online payment systems by leveraging Machine Learning (ML) techniques. The goal is to create a sophisticated fraud detection system capable of identifying and proactively preventing fraudulent transactions in real-time. The project involves meticulous data collection, pre-processing, feature engineering, and integrating supervised learning techniques like Logistic Regression, Decision Trees, Random Forests, and Support Vector Machines. Real-time monitoring and alerts form the frontline defence of the system, ensuring timely intervention and prevention of potential fraudulent transactions. The project adopts an iterative approach to model evaluation and refinement, ensuring the system remains responsive to emerging fraud tactics. Regular updates and fine-tuning based on new data and evolving patterns are integral to the project's success in maintaining a high level of accuracy and adaptability. In conclusion, the project aims to be a beacon of security in the digital commerce landscape, providing a shield against the ever-evolving threat of online payment fraud.

1.2 Purpose

ML-based Online Payments Fraud Detection aims to enhance security, prevent fraudulent activities, and maintain user confidence in digital transactions. By continuously monitoring transactions and identifying suspicious patterns, ML-driven fraud detection acts as a vigilant gatekeeper, minimizing the impact on users and financial institutions.

Financial institutions play a significant role in ensuring the integrity of online transactions, and ML-based fraud detection helps mitigate potential financial liabilities. The system is flexible and adaptive, learning from new data and emerging patterns to remain effective in the face of evolving threats.

ML-based fraud detection systems can analyze vast amounts of data in real-time, automating the detection process and reducing reliance on manual intervention. This enhances operational efficiency for financial institutions, allowing them to focus on strategic tasks while the system handles routine fraud detection.

ML algorithms can minimize false positives, ensuring that legitimate transactions are not mistakenly flagged as fraudulent. By demonstrating commitment to security and regulatory adherence, ML-based fraud detection systems help

institutions meet stringent compliance and regulatory requirements, contributing to the overall integrity and trustworthiness of online transactions.

2. LITERATURE SURVEY

2.1 Existing problem

Machine learning models are widely used in Online Payments Fraud Detection to analyse patterns, anomalies, and detect fraudulent activities. Logistic Regression is a foundational model for binary classification problems, while Decision Trees are tree-like models that capture complex decision boundaries and identify patterns in data. Support Vector Machines (SVM) are powerful for binary classification and are effective in high-dimensional spaces. Deep learning models, particularly neural networks, have shown promise in fraud detection, as they can automatically learn intricate patterns and representations from large and complex datasets. K-Nearest Neighbours (KNN) is a simple yet effective algorithm for local fraud patterns. Gradient Boosting Models, Isolation Forests, Autoencoders, and One-Class SVM are powerful ensemble techniques for fraud detection. The selection of a specific model or combination of models depends on the dataset's characteristics, fraud patterns, and computational resources available.

2.2 References

1. Manek H (2019) Title : review on various methods for fraud transaction to secure your paperas per UGC guidelines we are providing a electronic bar code, Nov 2018
2. Chaudhary K, Yadav J, Mallick B (2012) A review of fraud detection techniques: credit card.Int J Comput Appl 45(1):975–8887
3. Abdallah A, Maarof MA, Zainal A (2016) Fraud detection system: a survey. J Netw ComputAppl 68:90–113
4. Van Vlasselaer V et al (2015) APATE: a novel approach for automated credit card transactionfraud detection using network-based extensions. Decis Support Syst 75:38–48
5. Aihua S, Rencheng T, Yaochen D (2007) Application of classification models on credit cardfraud detection. In: Proceedings-ICSSSM'07 2007 International Conference Service System Service Management, no. 1997, 2007, pp 2–5
6. Whitrow C, Hand DJ, Juszczak P, Weston D, Adams NM (2009) Transaction aggregation as astrategy for credit card fraud detection. Data Min. Knowl. Discov. 18(1):30–55
7. Ogwueleka FN (2011) Vol_6(3)_311-322_Ogwueleka.pdf. 6(3):311–322
8. Sahin Y, Duman E (2011) Detecting credit card fraud by decision trees and support vectormachines. Int Multiconference Eng Comput Sci I:6
9. Mahmoudi N, Duman E (2015) Detecting credit card fraud by modified fisher discriminantanalysis. Exp Syst Appl 42(5):2510–2516

10. Awoyemi JO, Adetunmbi AO, Oluwadare SA (2017) Credit card fraud detection usingmachine learning techniques: a comparative analysis. In: Proceedings of the IEEE International Conference Computing Networking Informatics, ICCNI 2017, 2017, vol 2017-Jan, pp 1–9

2.3 Problem Statement Definition

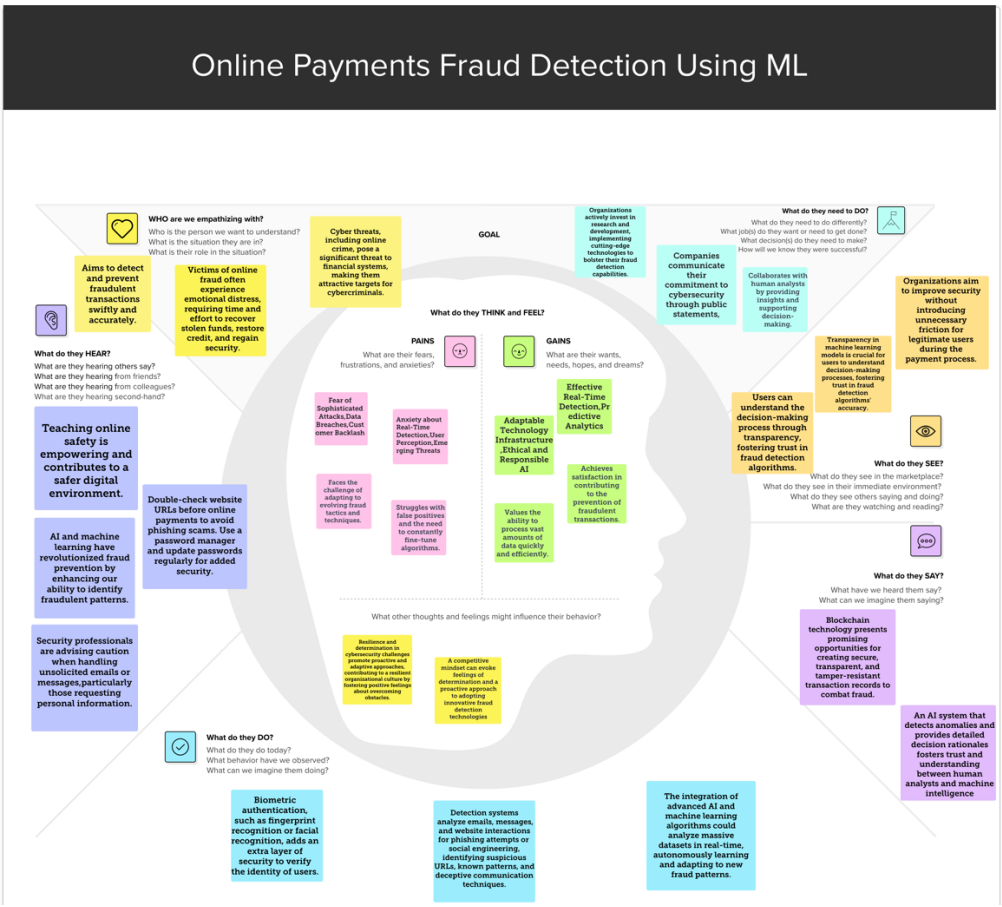
The evolving landscape of online payments presents a significant challenge in fraud detection. The development of a robust and adaptive Online Payments Fraud Detection system utilizing Machine Learning (ML) techniques is crucial to enhance the security of online payment systems and maintain a seamless user experience. The project aims to address several problem areas, including an imbalanced dataset, dynamic fraud patterns, false positives and negatives, adversarial attacks, real-time processing, model interpretability, seamless integration with existing systems, privacy concerns, model adaptability, and regulatory compliance.

The imbalanced dataset poses a challenge in training models that can distinguish between legitimate and fraudulent transactions. ML models must be capable of learning and recognizing new and sophisticated fraud tactics in real-time, ensuring a proactive defense against evolving threats. Achieving a balance between minimizing false positives and false negatives is essential to prevent unnecessary disruptions for legitimate users while ensuring effective identification of fraudulent transactions.

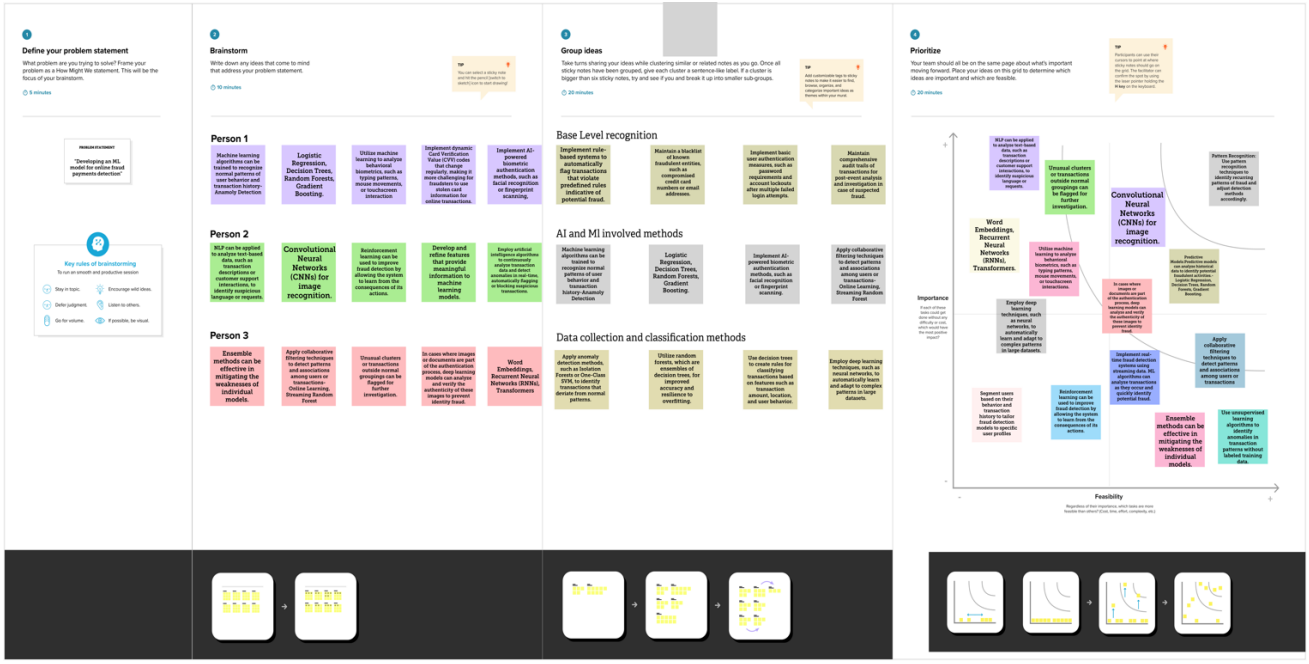
The system must also be resilient to adversarial attacks and adhere to diverse and evolving regulatory frameworks related to online payments and fraud detection. By combining advanced ML techniques, continuous monitoring, collaboration with domain experts, and a commitment to staying ahead of evolving fraud tactics, the successful implementation of a robust Online Payments Fraud Detection system will contribute to creating a secure and trustworthy environment for users and financial institutions engaged in digital commerce**.**

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 **Ideation & Brainstorming**



4. REQUIREMENT ANALYSIS

4.1 Functional requirement

The functional requirements for an Online Payments Fraud Detection system using Machine Learning outline the system's capabilities and features to tackle identified challenges and achieve project objectives.

The system must collect and preprocess a diverse dataset, including transactional data, user behavior, and contextual information, to train accurate machine learning models. Feature engineering techniques are used to select and transform relevant variables for effective fraud detection. Multiple ML models are trained, including Logistic Regression, Decision Trees, Random Forests, Neural Networks, and Ensemble Methods, to improve overall fraud detection accuracy. Real-time transaction monitoring is essential for identifying and preventing fraudulent activities. Alerts are generated when suspicious activities or potential fraud are detected, facilitating quick intervention. An adaptive model update mechanism is implemented to maintain high accuracy over time. The system must integrate seamlessly with various online payment platforms and systems, providing interpretability and explanation mechanisms for model decisions. User profile and behavior analysis is also necessary to identify deviations that may indicate fraudulent activity. Privacy protection measures, such as anonymization and encryption, are implemented to safeguard user data. Scalability is essential for handling varying transaction volumes and maintaining system performance. An audit trail of model decisions is maintained, facilitating analysis, debugging, and compliance reporting. Compliance with regulatory standards is crucial for legal and ethical operation and preventing regulatory penalties**.**

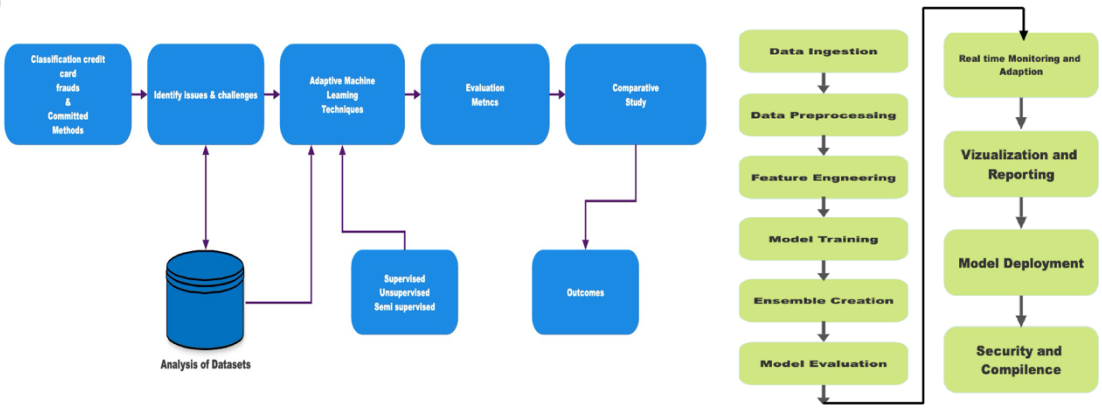
4.2 Non-Functional requirements

Non-functional requirements are essential for the success and reliability of an Online Payments Fraud Detection system. These requirements describe the qualities and characteristics of the system, rather than defining specific behaviors or features. Scalability is paramount for handling an increasing volume of transactions without compromising performance. Performance is essential for real-time fraud detection, ensuring minimal downtime and robust operation. Accuracy is paramount for effective fraud detection while minimizing false positives and negatives. Adaptability is crucial for maintaining the effectiveness of the system over time and addressing evolving fraud patterns. Security is paramount to protect sensitive transactional data and maintain the integrity of the fraud detection system. Compliance with data protection regulations and industry-specific standards for online payments and fraud detection is essential for legal and ethical operation. Privacy-preserving mechanisms are crucial for building and maintaining trust in the system. Interpretability is essential for

fostering trust and confidence in the system. Usability is essential for efficient system management and analysis of fraud-related information. Maintainability ensures the long-term viability and adaptability of the fraud detection system. Integration is crucial for incorporating the fraud detection system into the existing online payment infrastructure without disrupting user transactions. Auditability is essential for compliance reporting, debugging, and accountability. By addressing these non-functional requirements, the Online Payments Fraud Detection system can effectively detect and prevent fraudulent activities while operating seamlessly, securely, and in compliance with relevant regulations.

5. PROJECT DESIGN

5.1 Data Flow Diagrams & User Stories

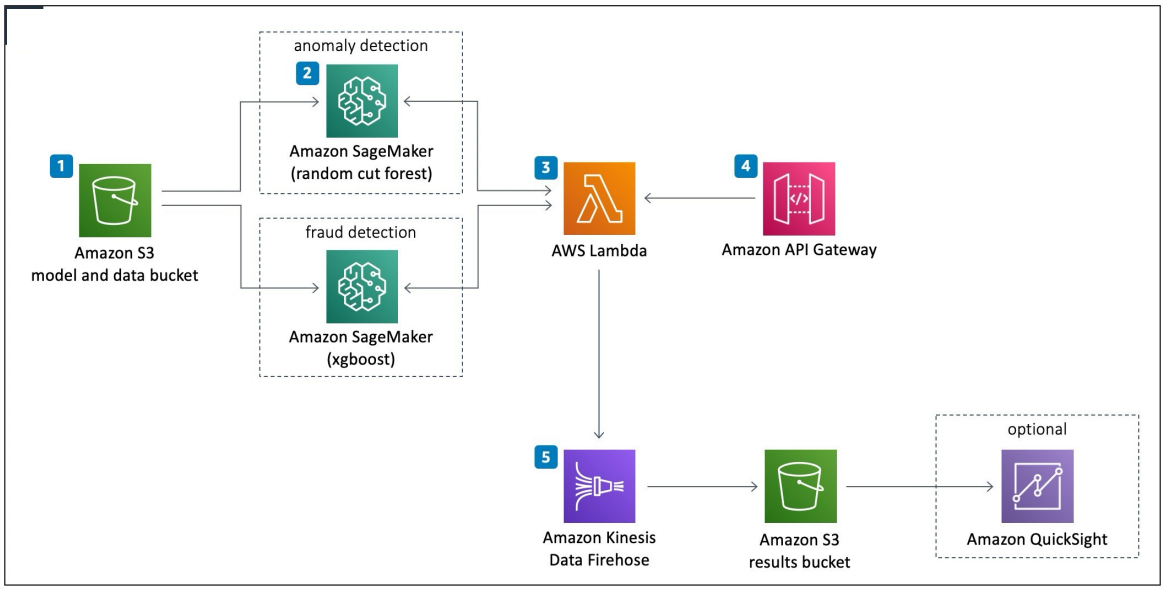


User Stories :

Use the below template to list all the user stories for the product.

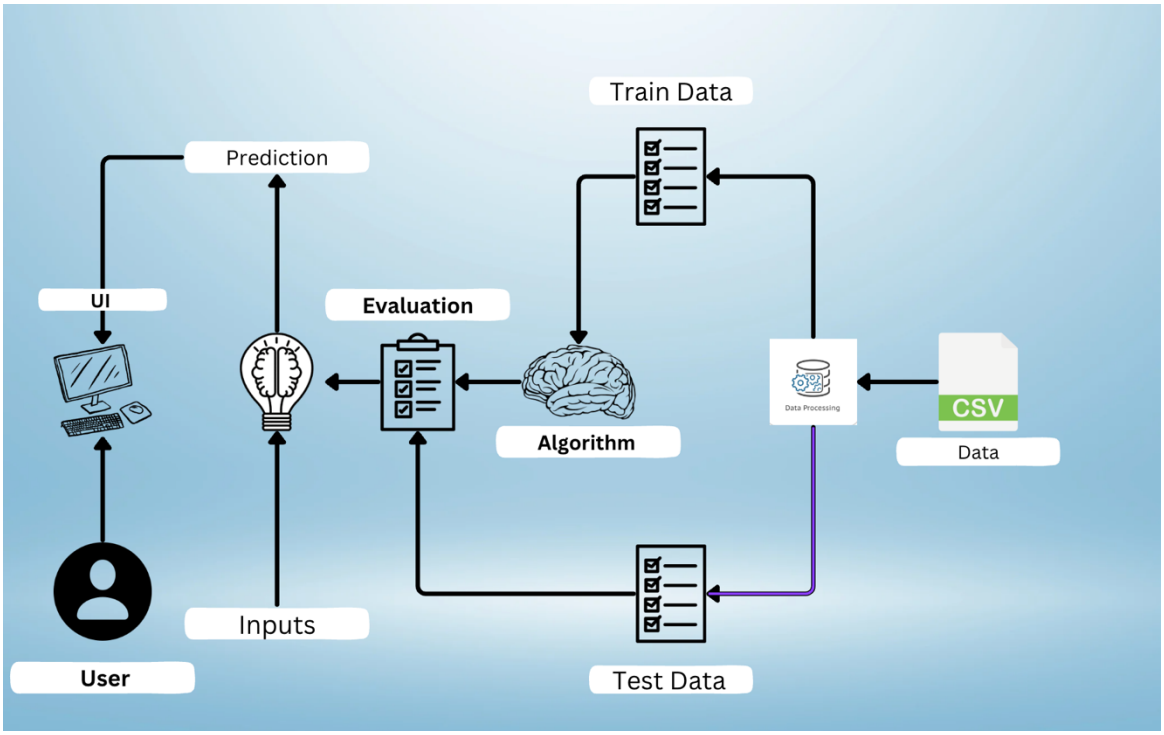
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Project setup & Infrastructure	USN-1	To Setup an Online Payments Fraud Detection using ML	1	High	Ruksana
Sprint-1	Development environment	USN-2	Gather a diverse dataset of Text that represents the details of the Transaction made by the Costumer.	2	High	Omprakash
Sprint-2	Data collection	USN-3	Users want assurance of secure handling and transparency in data collection practices. Concerns about privacy and potential misuse of personal information can lead to apprehension	3	High	Karthik
Sprint-2	data preprocessing	USN-4	Explore and evaluate different deep learning architectures (e.g., svm, random forest, decision tree) to select the most suitable model for online payments fraud detection	4	High	Omprakash
Sprint-3	model development	USN-5	train the selected Machine learning model using the preprocessed dataset and monitor its performance on the validation set.	5	High	Karthik
Sprint-3	Training	USN-6	incorporate data training using handing null values,handling of outliers ,separating test and train data to enhance the model's resilience and boost its accuracy.	6	Medium	Ruksana
Sprint-4	model deployment & Integration	USN-7	deploy the trained Machine learning model as an API or web service to make it accessible for online payment fraud detection. Integrate the model's API into a user-friendly web interface for users to Know the fraud payments.	2	Medium	Karthik
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.	1	Medium	Omprakash Ruksana

5.2 Solution Architecture



6. P ROJECT PLANNING & SCHEDULING

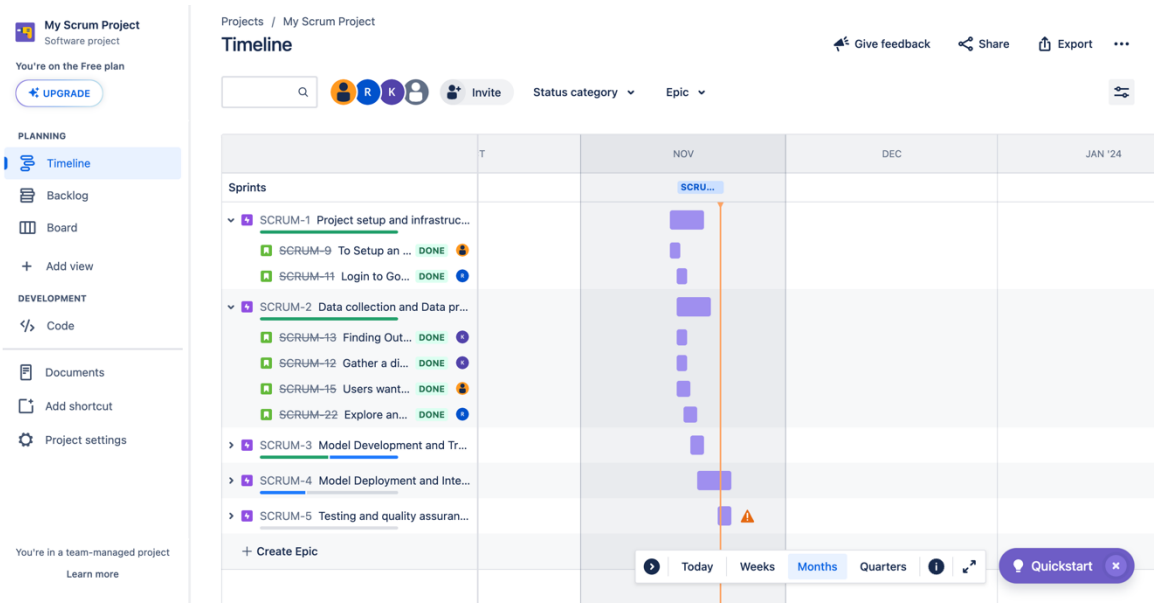
6.1 Technical Architecture



6.2 Sprint Planning & Estimation

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	1 Days	10 Nov 2023	11 Nov 2023	20	11 Nov 2023
Sprint-2	7	1 Days	11 Nov 2023	12Nov 2023		
Sprint-3	9	2 Days	13 Nov 2023	15 Nov 2023		
Sprint-4	2	2 Days	16 Nov 2023	18 Nov 2023		
Sprint-5	1	4 Days	17 Nov 2023	21 Nov 2023		

6.3 Sprint Delivery Schedule



7. CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1 Feature 1(xgboost)

A function named xgboost is created and train and test data are passed as the parameters. Inside the function, the xgboostClassifier algorithm is initialised and training data is passed to the model with the .fit() function. Test data is predicted with .predict() function and saved in a new variable. For evaluating the model, confusion matrix and classification report is done

xgboost Classifier

```
[ ] import xgboost as xgb
    from sklearn.metrics import accuracy_score
    xgb1=xgb.XGBClassifier()
    xgb1.fit(x_train,y_train1)
    y_test_predict5=xgb1.predict(x_test)
    test_accuracy=accuracy_score(y_test1,y_test_predict5)
    test_accuracy
```

```
0.997
```

```
[ ] y_train_predict5=xgb1.predict(x_train)
    train_accuracy=accuracy_score(y_train1,y_train_predict5)
    train_accuracy
```

```
1.0
```

```
[ ] pd.crosstab(y_test1,y_test_predict5)
```

col_0	0	1
row_0		
0	3	3
1	0	994

```
[ ] from sklearn.metrics import classification_report,confusion_matrix
    print(classification_report(y_test1,y_test_predict5))
```

```
              precision    recall  f1-score   support

           0       1.00      0.50      0.67         6
           1       1.00      1.00      1.00       994

    accuracy                           1.00      1000
   macro avg       1.00      0.75      0.83      1000
weighted avg       1.00      1.00      1.00      1000
```

7.2 Feature 2(decision tree)

Decision tree Classifier

A function named Decisiontree is created and train and test data are passed as the parameters. Inside the function, the DecisiontreeClassifier algorithm is initialised and training data is passed to the model with the .fit() function. Test data is predicted with the .predict() function and saved in a new variable. For evaluating the model, a confusion matrix and classification report is done.

2)Decision Tree Classifier

```
[ ] from sklearn.tree import DecisionTreeClassifier
    from sklearn.metrics import accuracy_score
    dtc=RandomForestClassifier()
    dtc.fit(x_train,y_train)
    y_test_predict2=dtc.predict(x_test)
    test_accuracy=accuracy_score(y_test,y_test_predict2)
    test_accuracy
```

```
0.998
```

```
[ ] 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)
```

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

col_0	is Fraud	is not Fraud
isFraud		
is Fraud	4	2
is not Fraud	0	994

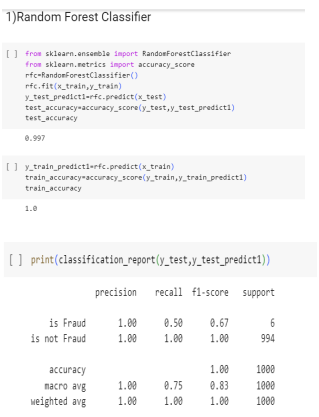
```
[ ] print(classification_report(y_test,y_test_predict2))
```

```
                precision    recall  f1-score   support

    is Fraud       1.00      0.67      0.80         6
is not Fraud       1.00      1.00      1.00       994

    accuracy                           1.00      1000
   macro avg       1.00      0.83      0.90      1000
weighted avg       1.00      1.00      1.00      1000
```

7.3 Database Schema (if Applicable)

8. PERFORMANCE TESTING

8.1 Performace Metrics

S.No.	Parameter	Values	Screenshot
1.	Metrics	**Regression Model:** MAE - , MSE - , RMSE - , R2 score - **Classification Model:** Confusion Matrix - , Accuray Score- & Classification Report -	**Random Forest Classifier :** 

1)Random Forest Classifier

```
[ ] from sklearn.ensemble import RandomForestClassifier
    from sklearn.metrics import accuracy_score
    rfc=RandomForestClassifier()
    rfc.fit(x_train,y_train)
    y_test_predict1=rfc.predict(x_test)
    test_accuracy=accuracy_score(y_test,y_test_predict1)
    test_accuracy
```

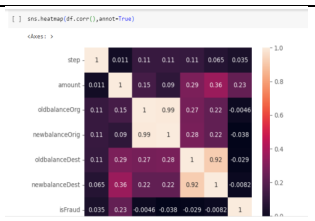
```
0.997
```

```
[ ] y_train_predict1=rfc.predict(x_train)
    train_accuracy=accuracy_score(y_train,y_train_predict1)
    train_accuracy
```

```
1.0
```

```
[ ] print(classification_report(y_test,y_test_predict1))
```

```
                precision    recall  f1-score   support

    is Fraud       1.00      0.50      0.67         6
is not Fraud       1.00      1.00      1.00       994

    accuracy                           1.00      1000
   macro avg       1.00      0.75      0.83      1000
weighted avg       1.00      1.00      1.00      1000
```



2) **Decision Tree Classifier:**

2)Decision Tree Classifier

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
dtc=RandomForestClassifier()
dtc.fit(x_train,y_train)
y_test_predict2=dtc.predict(x_test)
test_accuracy=accuracy_score(y_test,y_test_predict2)
test_accuracy
```

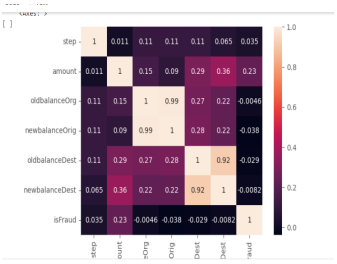
```
0.998
```

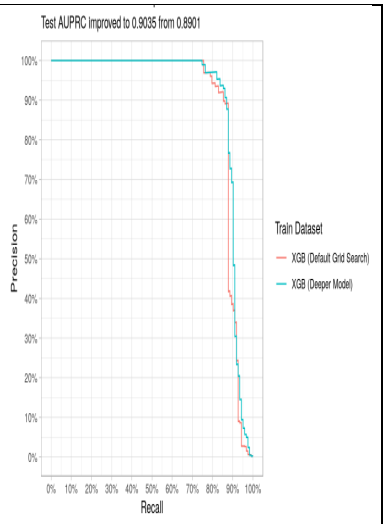
```
y_train_predict2=dtc.predict(x_train)
train_accuracy=accuracy_score(y_train,y_train_predict2)
train_accuracy
```

```
1.0
```

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

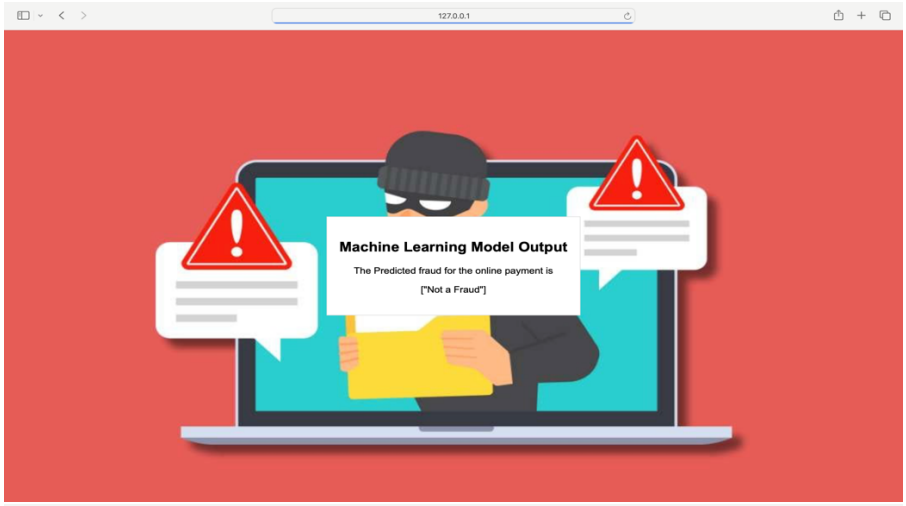
	precision	recall	f1-score	support
is Fraud	1.00	0.67	0.80	6
is not Fraud	1.00	1.00	1.00	994
accuracy			1.00	1000
macro avg	1.00	0.83	0.90	1000
weighted avg	1.00	1.00	1.00	1000

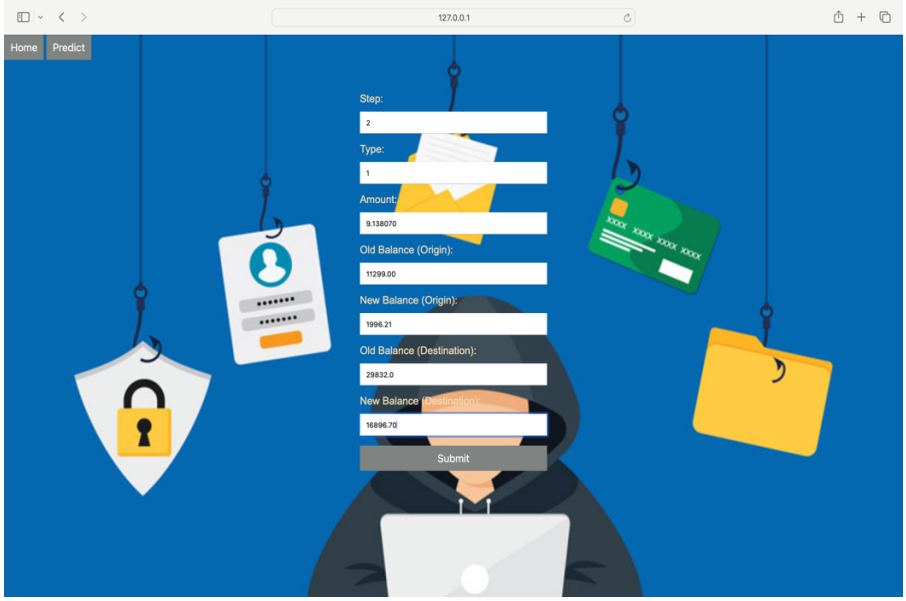


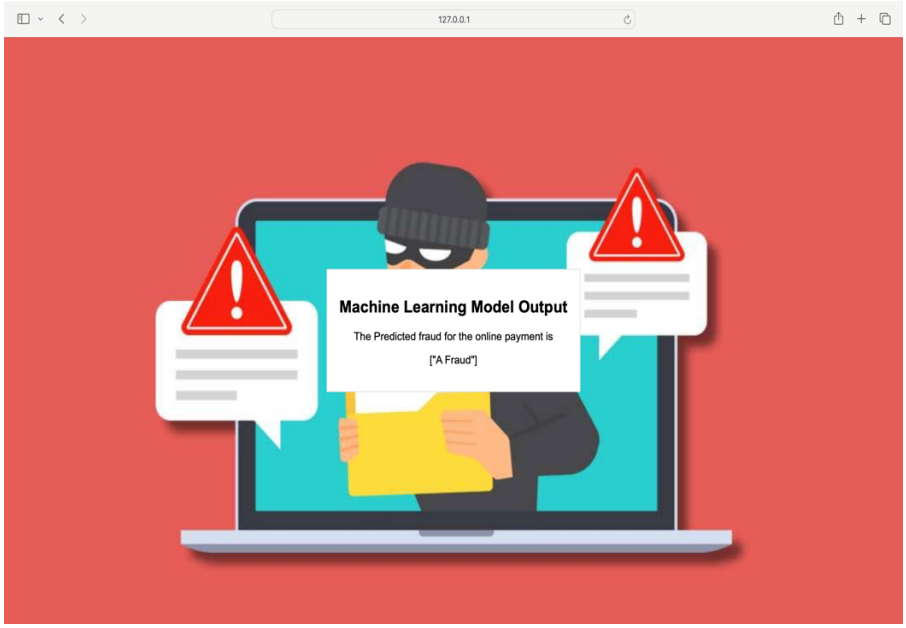
2.	Tune the Model	Hyperparameter Tuning - Validation Method -	

9. RESULTS

9.1 Output Screenshots


127.0.0.1
Machine Learning Model Output
The Predicted fraud for the online payment is
["Not a Fraud"]


127.0.0.1
Home
Predict
Step:
2
Type:
1
Amount:
9.138070
Old Balance (Origin):
11299.00
New Balance (Origin):
1996.21
Old Balance (Destination):
29832.0
New Balance (Destination):
16896.70
Submit


127.0.0.1
Machine Learning Model Output
The Predicted fraud for the online payment is
["A Fraud"]

10. ADVANTAGES & DISADVANTAGES

ADVANTAGES

1) Detection of Anomalies

Machine learning fraud detection offers faster detection of anomalies in an ever-evolving threat landscape, enabling businesses to quickly learn from their data and identify unusual patterns indicating fraud.

2) Better Predictions

Machine learning fraud detection algorithms handle large datasets, enhancing prediction accuracy and reducing human error by learning from more data, thereby reducing the need for manual data analysis.

3) Saves Time and Money

Additionally, machine learning can help businesses save time and money by reducing the number of manual reviews that need to be conducted. By utilizing machine learning fraud detection, businesses can reduce the number of false positives and false negatives that result from manual reviews, which can save businesses time and money.

DISADVANTAGES

1) Inaccurate Prediction

Machine learning algorithms require a large amount of high-quality data to be effective. If the data used to train the algorithm is biased or lacks sufficient detail, the algorithm's predictions may be inaccurate.

2) Difficult To Interpret

Machine learning algorithms can be difficult to interpret and understand, especially for people who are not familiar with the technical details of how they work. This can make it difficult for people to understand why the algorithm is flagging certain transactions as potentially fraudulent.

3) Expensive

Machine learning algorithms can be expensive to implement and maintain, especially if a company does not have in-house expertise in this area.

4) Lack of Human Intelligence

Even the most advanced technology cannot replace the expertise and judgment of a human when it comes to evaluating and interpreting data to determine the risk of questionable activity. The psychological analysis and

understanding that a human can bring to the table are crucial in accurately filtering and interpreting data to determine the meaning of a risk score.

11. CONCLUSION

Machine learning (ML) classification algorithms are a promising tool for online payments fraud detection. They offer accuracy and efficiency in processing large datasets, pattern recognition, adaptability, automation, and scalability. ML algorithms can identify complex patterns and relationships within data, making them essential for detecting subtle and evolving fraud patterns. They can also automate and scale fraud detection, enabling financial institutions to handle the growing number of online transactions. ML algorithms can also identify anomalies in transaction data, reducing false positives and maintaining a positive user experience. However, challenges include imbalanced datasets, adversarial attacks, interpretability issues, data privacy concerns, and the dynamic fraud landscape. Addressing these challenges is crucial for the continued success of ML algorithms in online payments fraud detection. With ongoing research and advancements, ML algorithms are poised to play a pivotal role in securing online transactions and safeguarding financial ecosystems from fraudulent activities.

12. FUTURE SCOPE

The future of online payments fraud detection using machine learning (ML) algorithms is expected to be dynamic, driven by technological advancements, data availability, and the evolving nature of fraud. Key directions include deep learning and neural networks, explainable AI (XAI), unsupervised learning for anomaly detection, ensemble learning, continuous learning models, blockchain and cryptocurrency integration, biometric and behavioral authentication, federal learning, external data integration, AI-powered fraud investigation tools, and adversarial robustness. Deep learning models can capture intricate patterns and relationships in data, while explainable AI can provide transparency and interpretability. Ensemble learning methods can enhance predictive accuracy and generalization, while continuous learning models allow systems to adapt to evolving fraud patterns and tactics. The integration of biometric and behavioral authentication methods, federal learning, and external data integration can enhance predictive capabilities and help financial institutions collaborate and improve fraud detection models.

13. APPENDIX Source Code

GitHub & Project Demo Link

GitHub Link:-

https://github.com/smartinternz02/SI-GuidedProject-613758-1699949779

Project Demo Link:-

https://drive.google.com/drive/folders/1_w_LohxNRo6Nrx56zCyHMKOTTRzFUHwU?usp=drive_link