SRI VENKATESWARA ENGINEERING COLLEGE

SPONSORED BY THE EXHIBITION SOCIETY,HYDERABAD

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Industrial Oriented Mini Project On

FRAUD DETECTION IN BANKING DATA BY MACHINE LEARNING TECHNIQUES

BATCH-1

SHAIK SUMAYA (21631A0554)
MOHAMMED FARAAZ (21631A0520)
TELAPUTTA RAJU (21631A0539)
VEESALA PAVAN (21631A0535)

Under the Guidance of:
Mr.P.Rataiah M.Tech
Sr.Assistant Professor,CSE

Head Of The Department:
Mr.P.Rataiah M.Tech
Sr.Assistant Professor,CSE

CONTENTS:

INTRODUCTION:

- Online banking has grown quickly, leading to more fraud.
- Changing fraud tactics make it hard to detect and stop fraud.
- Technology can help both prevent and enable fraud, so quick detection is important.
- **Machine learning techniques** like Bayesian optimization and ensemble methods improve credit card fraud detection accuracy.

ABSTRACT:

- The study uses machine learning to find fraud in banking data, which is important for the financial sector.
- Class weight-tuning hyperparameters help the model better distinguish between fraudulent and authorized transactions.
- **CatBoost**, **LightGBM**, and **XGBoost** algorithms are used to improve the fraud detection method's performance, each offering unique advantages.
- Enhanced the accuracy of fraudulent transaction detection and reduced false positives, resulting in better performance metrics such as precision, recall, and F1-score.
- Tests with real-world data show that combining

PROBLEM STATEMENT:

- The banking sector faces challenges in detecting and preventing fraud like unauthorized transactions and identity theft.
- As digital banking grows, fraudsters keep changing their methods, making it hard to prevent losses.
- The aim is to create effective fraud detection methods that can adapt to new fraud tactics and constant updates and protect banks and customers.
- A strong fraud detection system is essential for maintaining trust in banks and keeping customer assets safe.
- It helps minimize financial losses caused by fraudulent activities.

OBJECTIVE:

- To develop a adaptable fraud detection system for banking data to prevent fraudulent activities.
- Objective is to ef ciently analyze banking data through the application of machine learning algorithms.
- To enhance dataset analysis by exploring diverse sampling and scaling techniques.
- To implement hybrid models as an extension to improve fraud detection accuracy.
- To create an extension of a user-friendly interface using Flask for user testing and authentication.

LITERATURE SURVEY:

- K. Gupta, K. Singh, G. V. Singh, M. Hassan, G. Himani, and U. Sharma review credit card fraud detection methods using machine learning for real-time solutions.
- The authors discuss dealing fraud through the selection of effective machine learning models.
- Predictive analysis is used to detect fraud instantly via an API module, as highlighted by the authors.
- The approach by Gupta et al. focuses on handling data efficiently to improve fraud detection.
- The proposed system by the authors aims to prevent credit card fraud in digital transactions.
- The authors note challenges such as development costs, false positives, privacy concerns, and evolving fraud tactics.

EXISTING SYSTEM:

- The system **uses rule-based detection** for possible fraudulent transactions.
- It establishes **limits** that trigger alerts for suspicious transactions.
- Human analysts have to **check** detected transactions, adding extra steps.
- The system finds it hard to **adapt** to new fraud methods as they develop ,making it difficult to handle changing patterns.
- The System has a high rate of **false positives**, that leads to customer dissatisfaction.

DISADVANTAGES :

- The system can't easily **adjust** to new fraud tactics because it relies on fixed rules.
- Human analysts need to **check** detected transactions, which causes delays and higher costs.
- The system often incorrectly **marks** authorized transactions as fraud, annoying customers and raising investigation costs.
- It has trouble spotting new **types** of fraud, making it less effective as tactics change.
- The system f nds it hard to **maintain** accuracy because there are many more authorized transactions than fraudulent ones.

PROPOSED SYSTEM:

- The project creates a adaptive fraud detection system to detect fraud in banking data using machine learning .
- It adjusts how important different types of transactions are to improve accuracy.
- This method is used to find the best settings for the system to work better.
- The system uses strong algorithms like **CatBoost**, **LightGBM**, and **XGBoost** for better fraud detection.
- Advanced deep learning techniques are used to further enhance the system.

ADVANTAGES:

- The system significantly enhances fraud detection accuracy compared to traditional methods.
- Adapts to evolving fraud tactics, maintaining its effectiveness over time.
- The system efficiently manages unbalanced data without compromising accuracy.
- Evaluation with multiple performance metrics ensures a complete assessment of its real-world effectiveness.

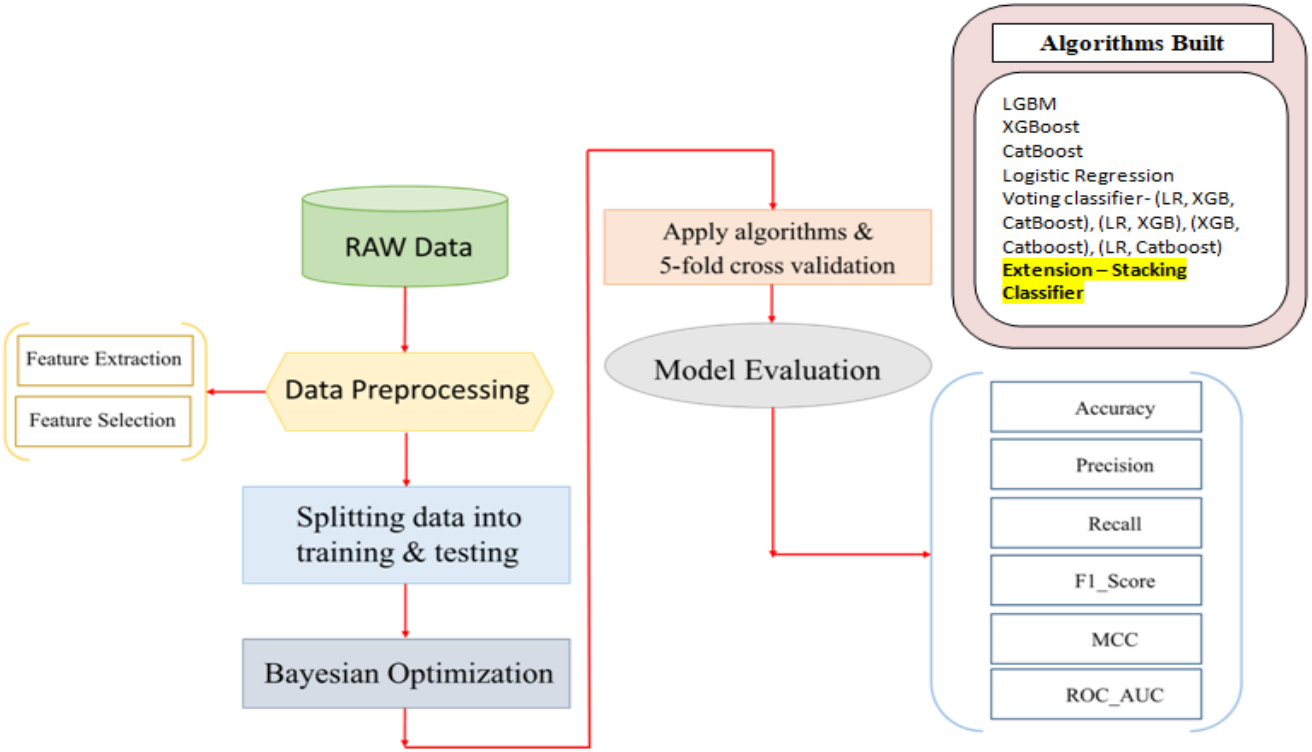
EXTENSION:

- The project uses a Stacking Classifier that combines predictions from **RandomForest** and LightGBM, with a gradient boosting classifier for better accuracy.
- It includes a Flask framework with SQLite for secure signup and signin, improving user testing and usability in fraud detection.

ADVANTAGES:

- The Stacking Classifier (RandomForest + LightGBM) performs better than other models in accurately detecting fraud.
- The Stacking Classifier combines different algorithms, making the fraud detection system stronger and more reliable.
- Flask integration provides a secure and easy-to-use signup and signin process, improving usability.
- The combination of the Stacking Classifier and Flask creates a complete fraud prevention system that boosts accuracy and enhances user interaction for better protection of banking data.

SYSTEM ARCHITECTURE:



MODULES:

1. Importing required Packages.
2. Exploring the dataset- Credit Card Fraud Data.
3. Visualization using seaborn & matplotlib.
4. Data Processing - Using Pandas Data frame.
5. Feature Extracting.
6. Feature Selection – using information gain.
7. Train & Test Split.

MODULES:

8. Training and Building the model.
9. As an extension we have built a stacking classifier by combining the predictions of multiple individual models to produce a more robust and accurate final prediction.
10. And we have built front end using flask framework for user testing and with user authentication.
11. Flask Framework with SQLite for signup and signin.
12. User gives input.
13. The given input is preprocessed and trained models are used for predictions.
14. Final outcome is displayed through the front-end.

TECHNIQUES USED:

1.Bayesian Optimization.

2.Cross-Validation (Stratified K-Fold).

3.SMOTE Sampling.

4.Under Sampling.

5.Class Weighting.

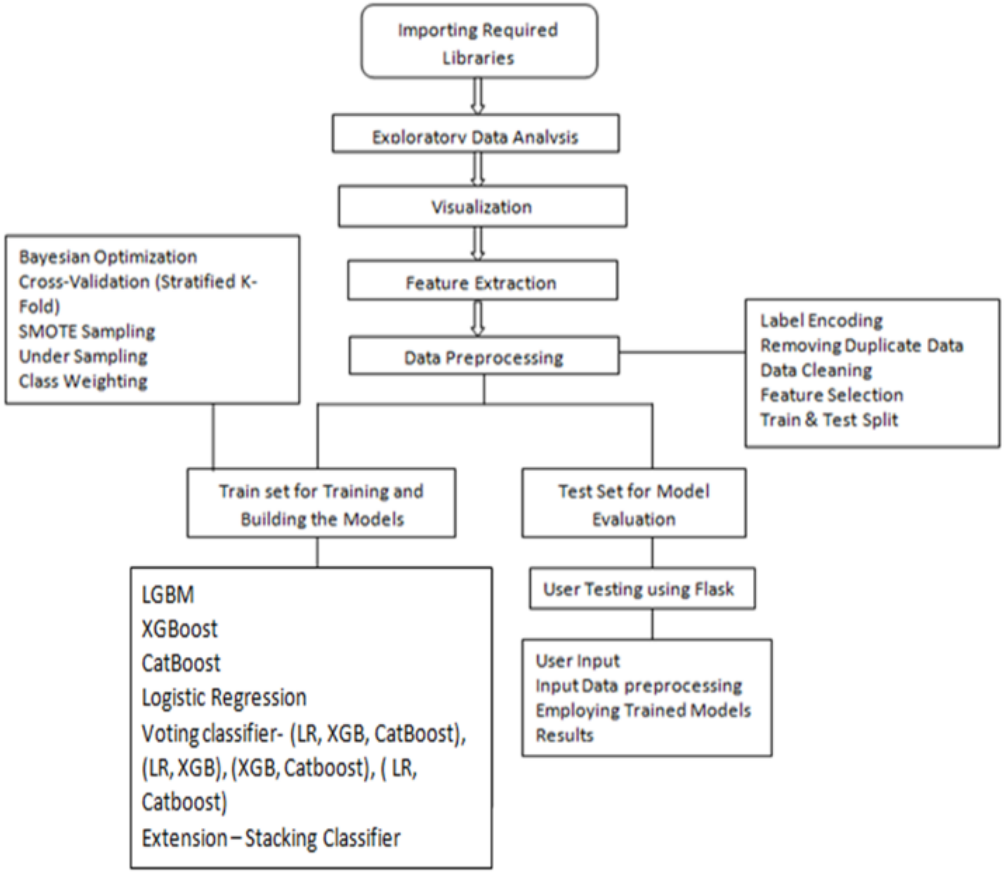
SOFTWARE/HARDWARE REQUIREMENTS:

Software Requirements

1. Software : Anaconda
2. Primary Language : Python 3.7 or Higher
3. Frontend Framework : Flask
4. Backend Framework : Jupyter Notebook
5. Database : Sqlite3
6. Front-End Technologies : HTML, CSS, JavaScript and Bootstrap4

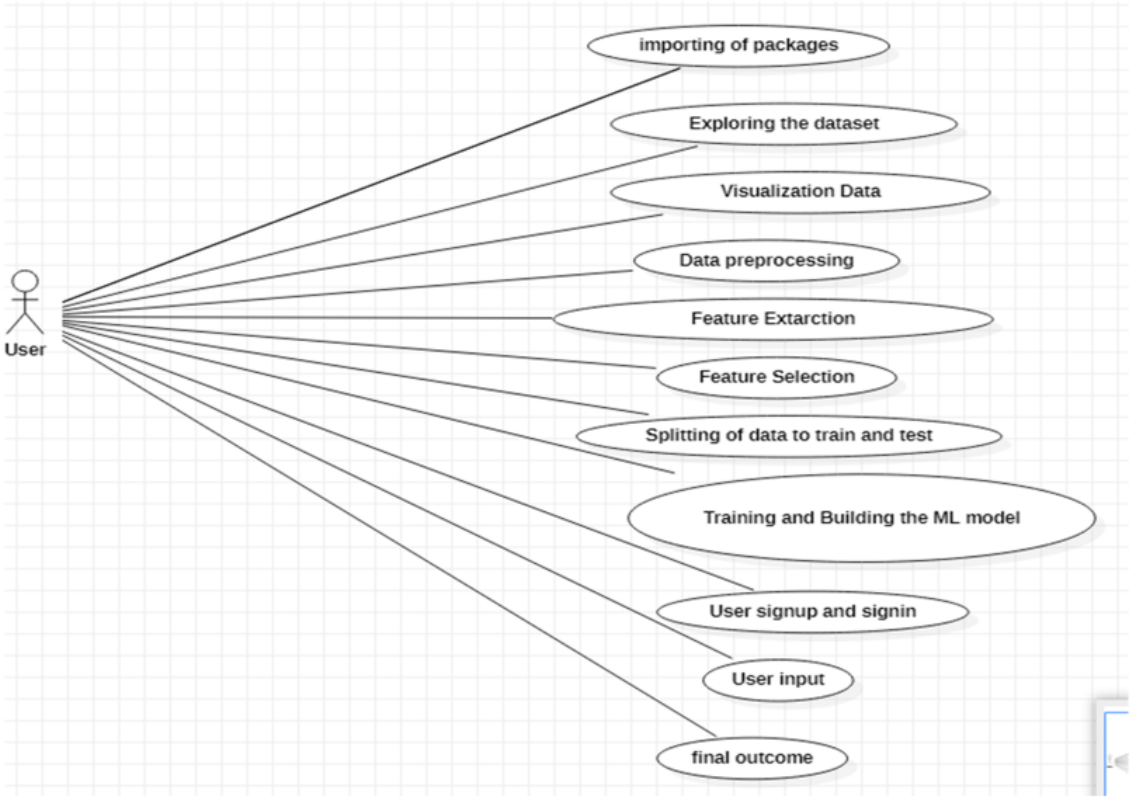
Hardware Requirements

1. Operating System : Windows ,MacOS, Linux
2. Processor : i5 and above
3. Ram : 8GB and above
4. Hard Disk : 25 GB and above



DATA FLOW DIAGRAM:

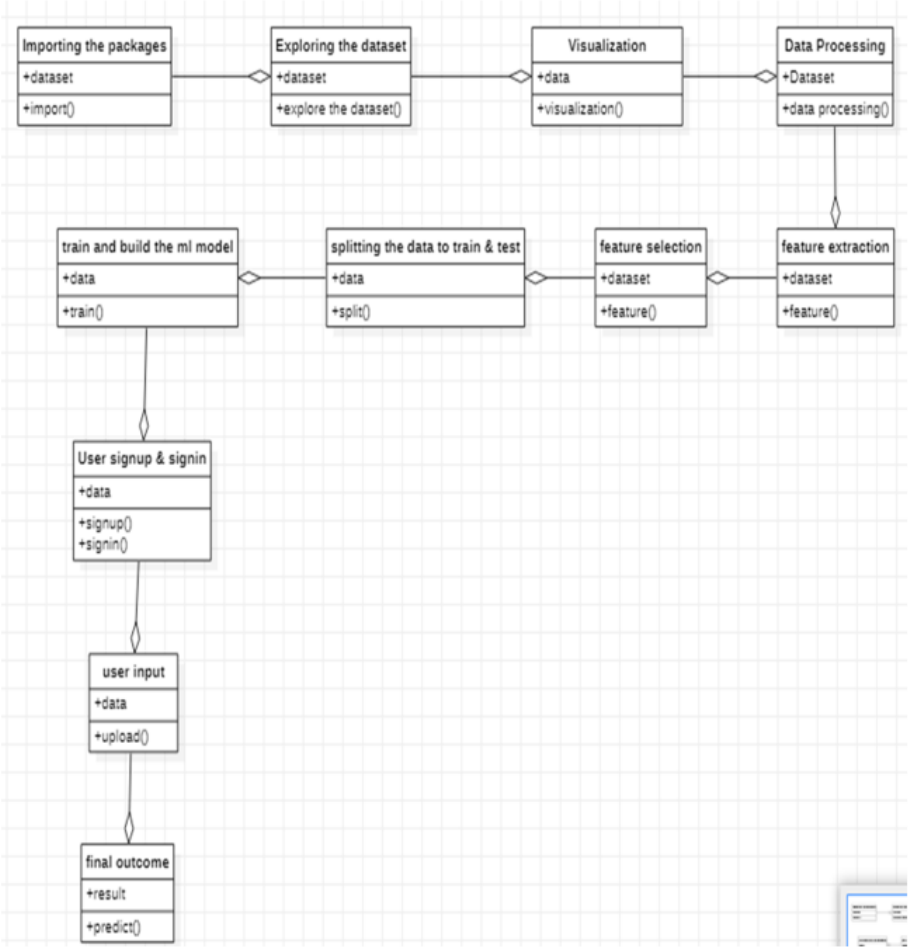
- Data Flow diagrams(DFDs) show how data moves through a system and how it is processed.
- Use Circles(processes),rectangles(entities), and arrows(data flow) to represent how data is input, processed, and output in the system.



UML DIAGRAMS:

❑USE CASE DIAGRAM:

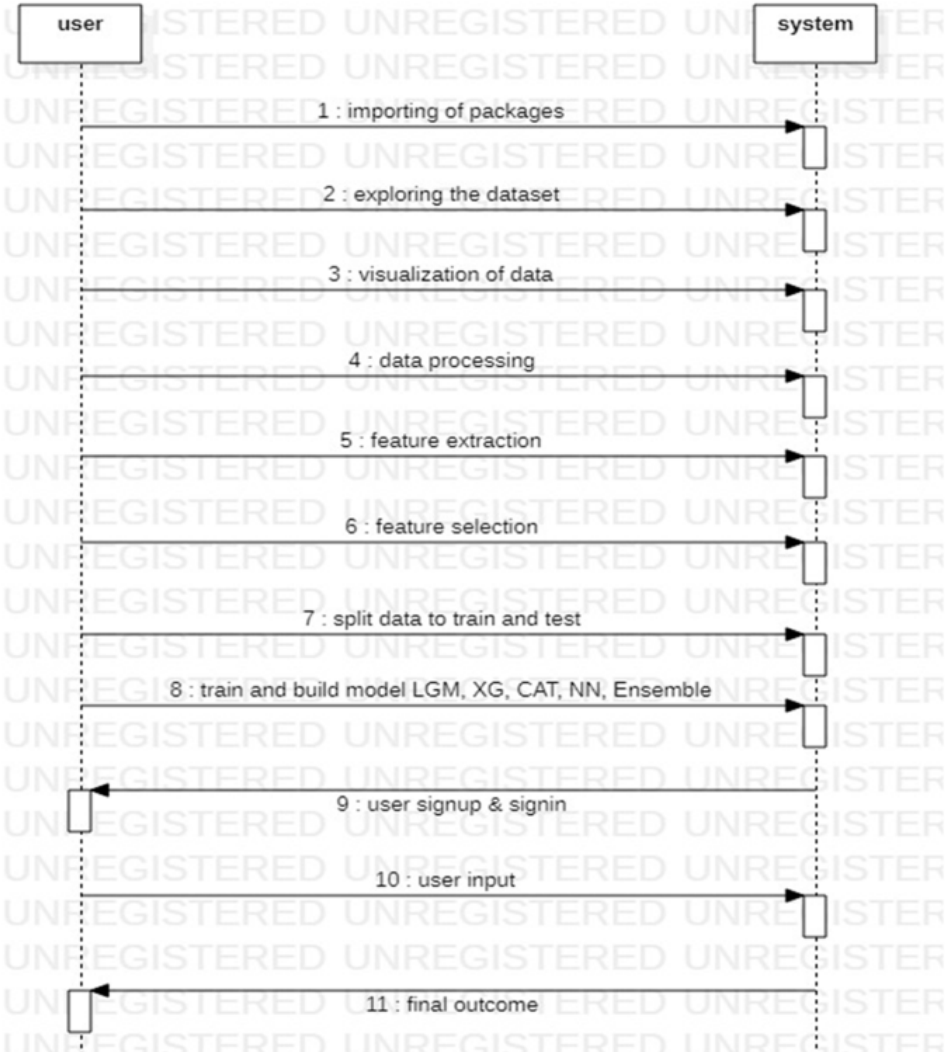
- ❖Use case diagrams describe the interactions between the system and its users (actors).
- ❖Create actors representing the users (e.g., end-users, administrators).



UML DIAGRAMS:

CLASS DIAGRAM:

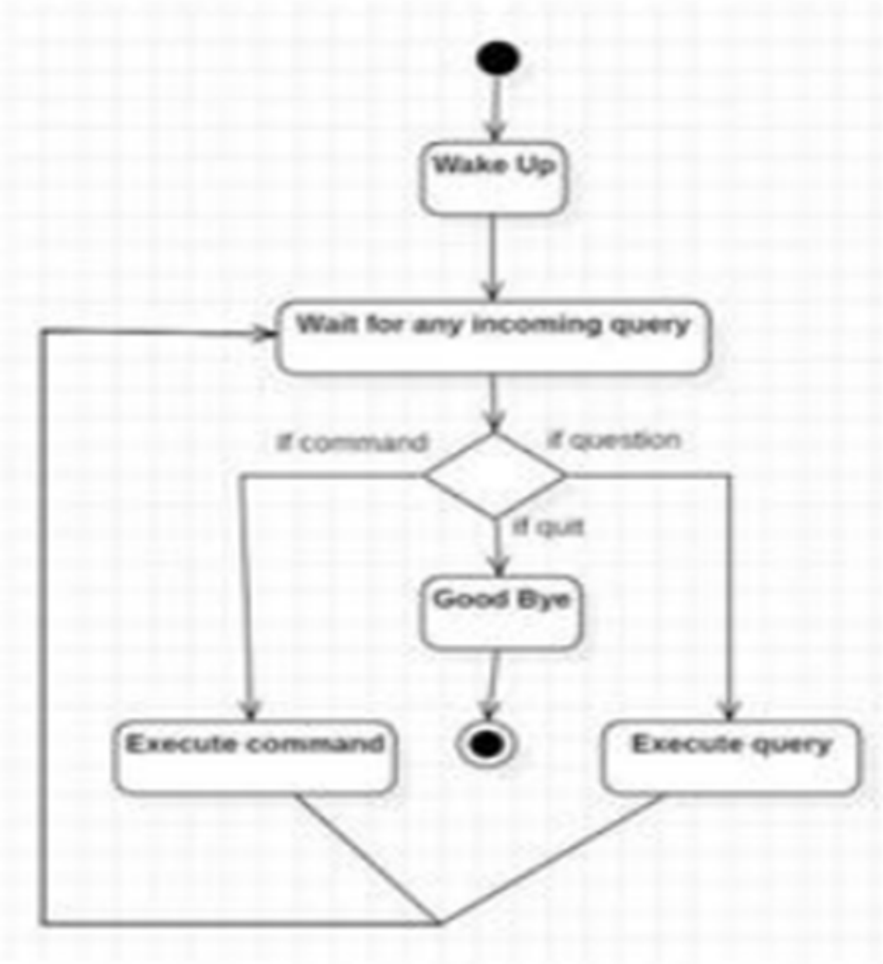
- Class diagrams show the static structure of your software, including classes, attributes, and relationships.
- Identify the main classes in your Fraud detection project (e.g., Data Preprocessing, Feature Extraction etc.).
- Add attributes and methods to each class.



UML DIAGRAMS:

❑SEQUENCE DIAGRAM:

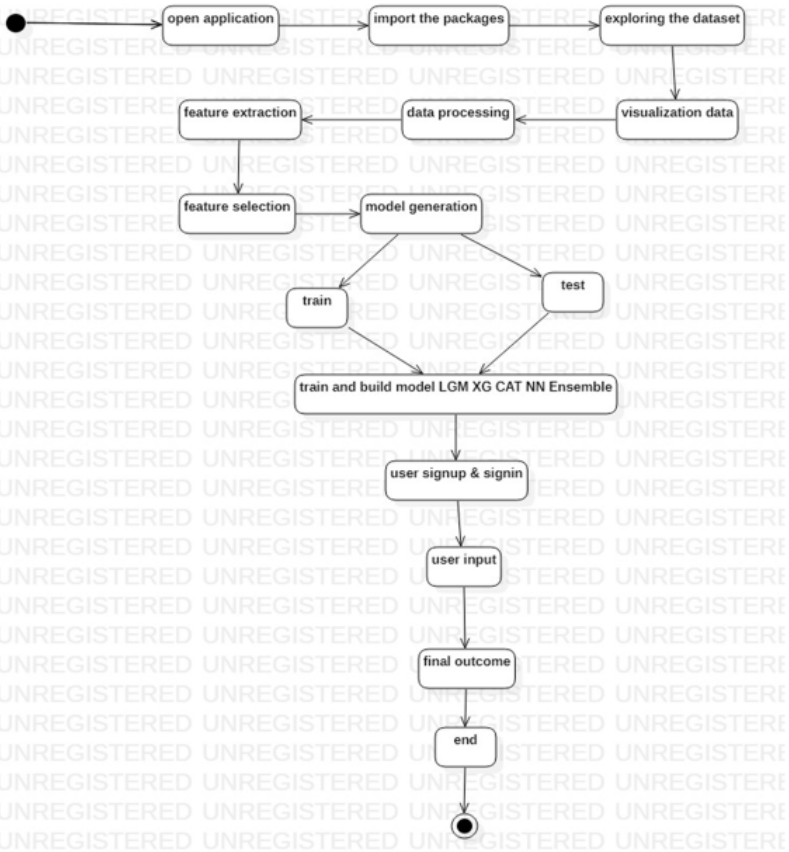
- ❖Sequence diagrams illustrate the interactions between objects over time.
- ❖Create a sequence diagram to visualize how various components of your fraud detection system interact during specific scenarios.
- ❖Use lifelines to represent objects (e.g., User, Fraud detection system) and arrows to represent messages and their order.



UML DIAGRAMS:

❑COLLABORATION DIAGRAM:

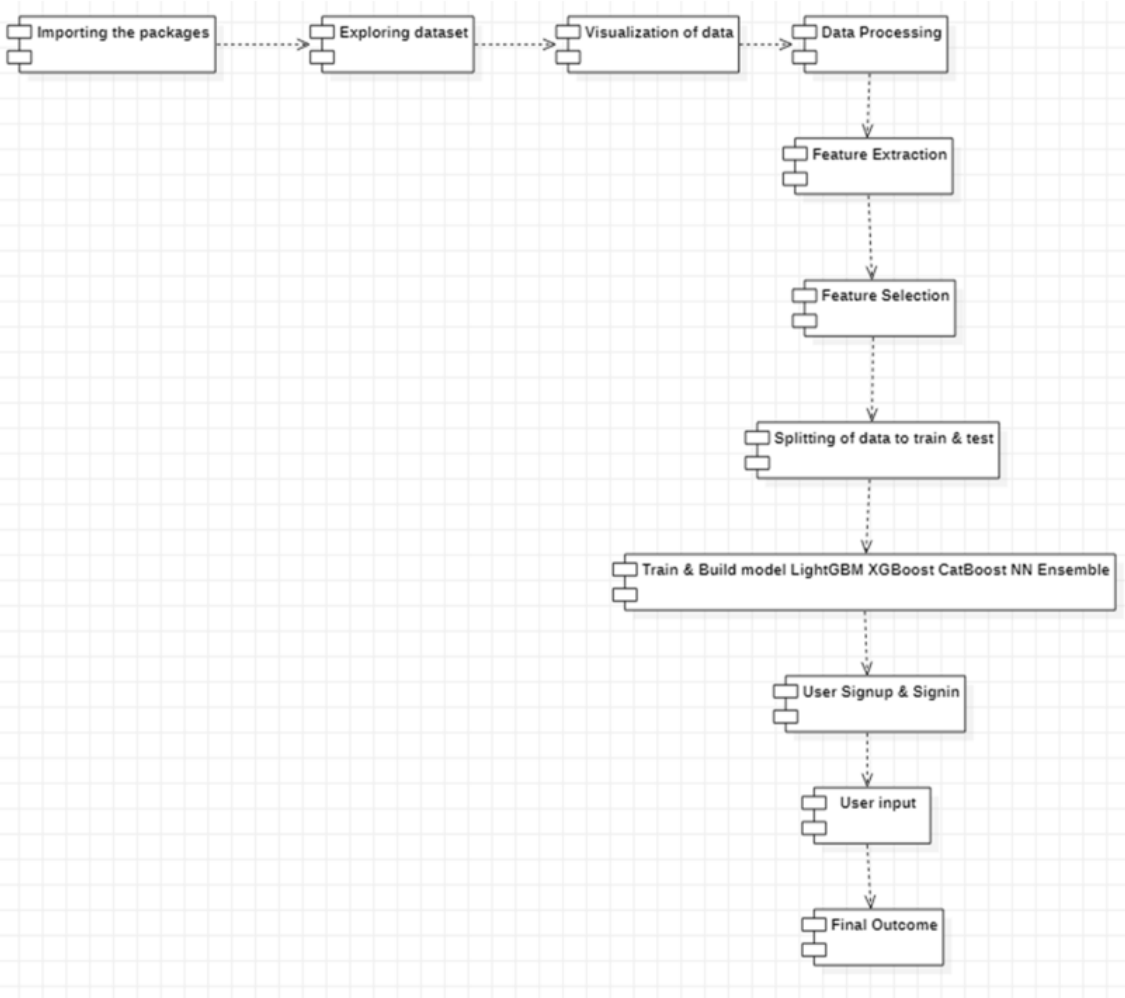
- ❖Collaboration diagrams show the interactions between objects but focus on the relationships and organization of objects.
- ❖Create a collaboration diagram to describe how various components of your fraud detection system interact during a scenario like a user transaction check.
- ❖Use numbered arrows to represent the sequence of messages and show how objects(e.g.,User,Fraud detection system) are associated.



UML DIAGRAMS:

ACTIVITY DIAGRAM:

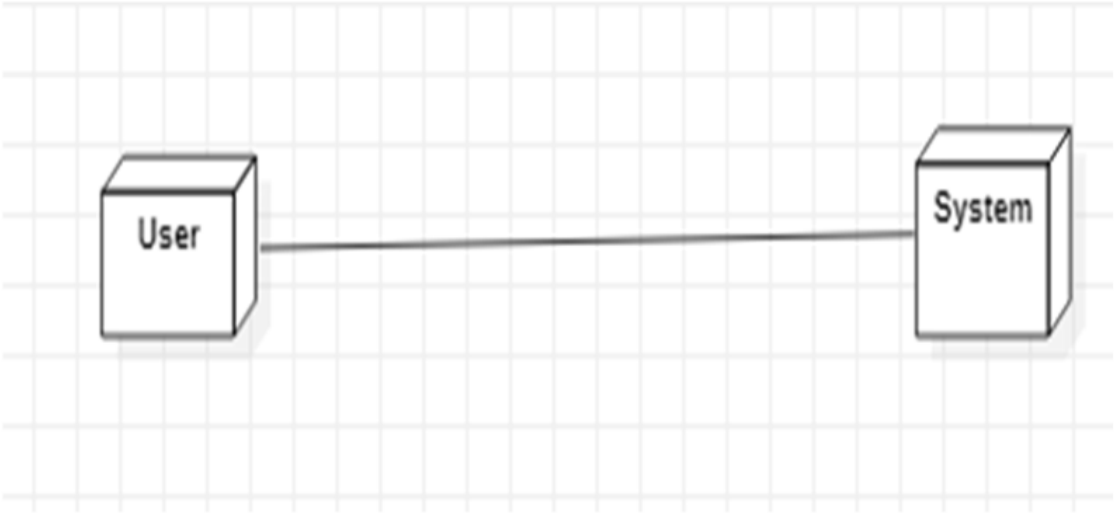
- Activity diagrams show the flow of activities and actions within a system.
- This activity diagrams describes the flow of interactions when users with your fraud detection system.
- Include decision points(e.g. data validation, forks(represents the use of different algorithms, and overall path through the system.



UML DIAGRAMS:

❑COMPONENT DIAGRAM:

- ❖Component diagrams represent the high-level structure of a system by showing its components and their relationships.
- ❖Create a Component Diagram to visualize how various parts of your fraud detection system work together during process of a transaction.



UML DIAGRAMS:

❑DEPLOYMENT DIAGRAM:

- ❖Deployment diagrams show how a system's parts are placed on physical devices /hardware.
- ❖Use boxes(nodes) to represent devices(e.g., phone, server, database) and lines to show communication between them.

ALGORITHMS USED:

1. **LGBM(Light Gradient Boosting Machine):** is known for its efficiency and speed in building decision trees.
2. **XGBoost(Extreme Gradient Boosting):** is a high-performance gradient boosting algorithm that uses parallel and distributed computing to handle large datasets efficiently.
3. **CatBoost(Categorical Boosting):** is specifically designed for handling categorical data and offers automatic feature transformation without requiring hyperparameter tuning.
4. **Logistic Regression:** uses the sigmoid function to map inputs to a probability score and applies a threshold to classify data, learning coefficients during training to fit the data effectively.
5. **Voting Classifiers:** combine the predictions of multiple models, such as Logistic Regression, XGBoost, and CatBoost, to leverage the strengths of each algorithm and improve prediction accuracy.
6. **Neural Network:** is a computational model inspired by the human brain, capable of capturing complex data patterns through adjusting weights during training.
7. **Stacking Classifier:** combines multiple base classifiers (RandomForest and LightGBM) with a final estimator (GradientBoosting) to make an optimized final prediction.

CREDIT CARD FRAUD DETECTION DATASET:

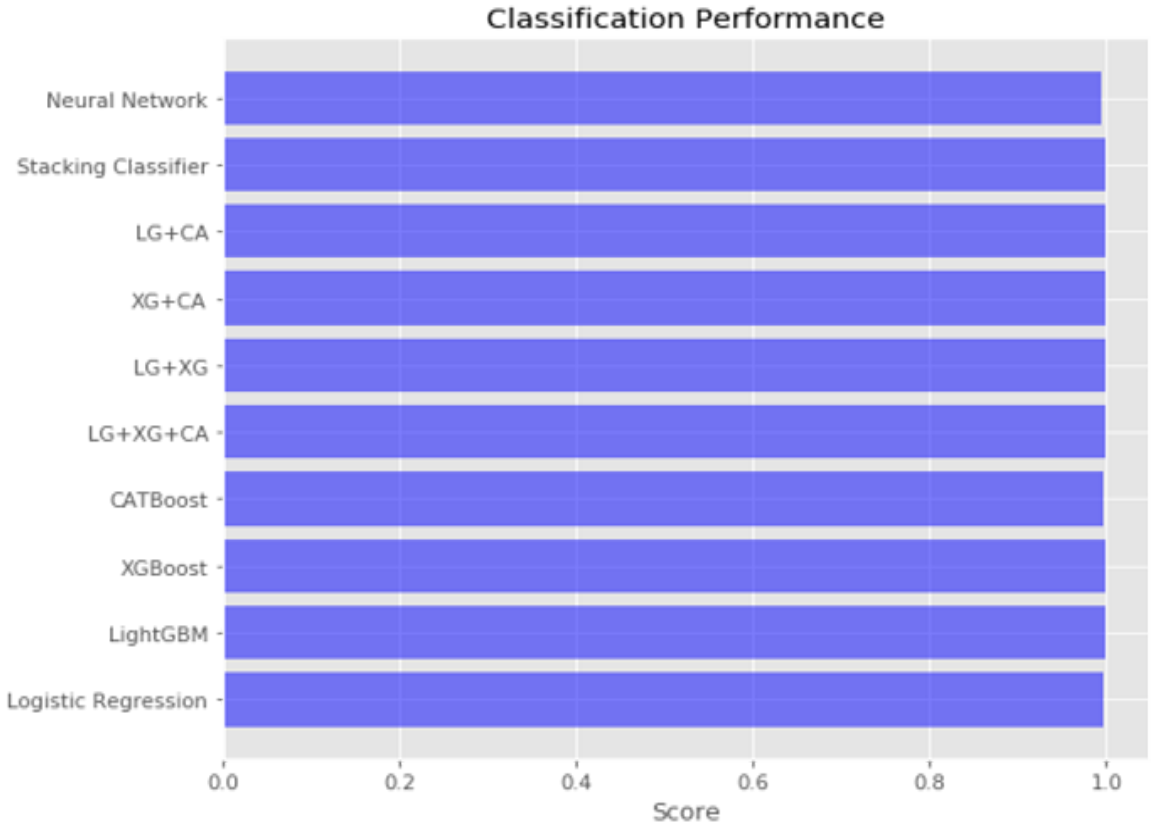
- We have used Credit Card Fraud Detection dataset taken from Kaggle to train machine learning algorithms.
- The dataset originally had various transaction-related features, like "Amount," "Time," and "V1" t o "V28."
- Details about the original features were kept confidential to safeguard sensitive information.

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	...	V23	V24	V25	V26	V27	V28	Amount	Class	hour	second
169876	-0.611712	-0.769705	-0.149759	-0.224877	2.028577	-2.019887	0.292491	-0.523020	0.358468	0.070050	...	0.380739	0.023440	-2.220686	-0.201146	0.066501	0.221180	1.79	0	16	1107
127467	-0.814682	1.319219	1.329415	0.027273	-0.284871	-0.653985	0.321552	0.435975	-0.704298	-0.600684	...	0.090660	0.401147	-0.261034	0.080621	0.162427	0.059456	1.98	0	10	2740
137900	-0.318193	1.118618	0.969864	-0.127052	0.569563	-0.532484	0.706252	-0.064966	-0.463271	-0.528357	...	-0.123884	-0.495687	-0.018148	0.121679	0.249050	0.092516	0.89	0	11	3182
21513	-1.328271	1.018378	1.775426	-1.574193	-0.117696	-0.457733	0.681867	-0.031641	0.383872	0.334853	...	-0.239197	0.009967	0.232829	0.814177	0.098797	-0.004273	15.98	0	4	2917
134700	1.276712	0.617120	-0.578014	0.879173	0.061706	-1.472002	0.373692	-0.287204	-0.084482	-0.696578	...	-0.076738	0.258708	0.552170	0.370701	-0.034255	0.041709	0.76	0	11	1723

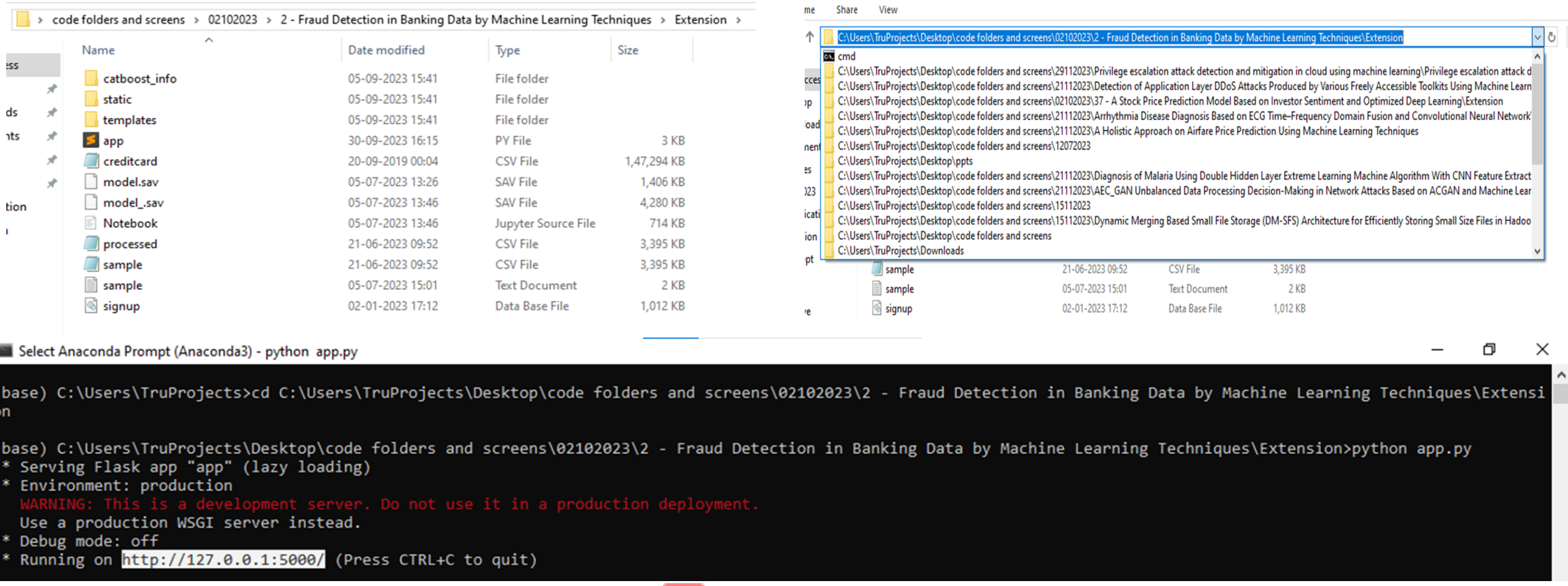
5 rows × 32 columns

PERFORMANCE EVALUATION:

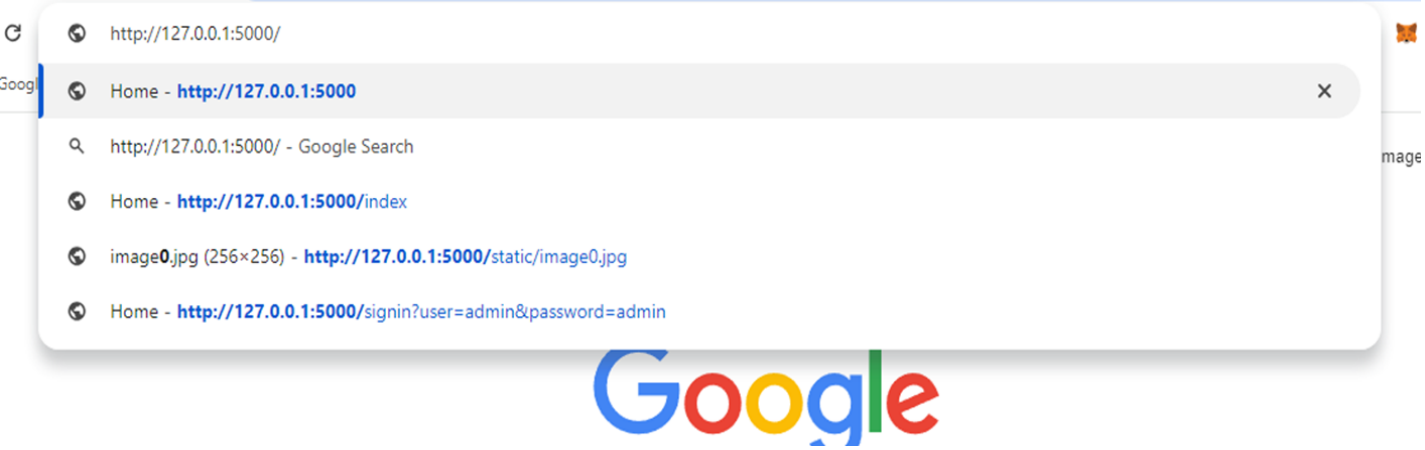
	Model	accuracy	precision	recall	f1
0	LGBM	0.996908	0.344762	0.413333	0.355455
1	XGB	0.999122	0.883232	0.588889	0.682707
2	CatBoost	0.999262	0.848312	0.713333	0.768263
3	Vot_Lg,Xg,Ca	0.996908	0.344762	0.413333	0.355455
4	Vot_Lg,Xg	0.999122	0.883232	0.588889	0.682707
5	Vot_Xg,Ca	0.999262	0.848312	0.713333	0.768263
6	Vot_Lg,Ca	0.999227	0.830345	0.733333	0.764458
7	**Stacking**	**0.999332**	**0.85101**	**0.753333**	**0.795105**

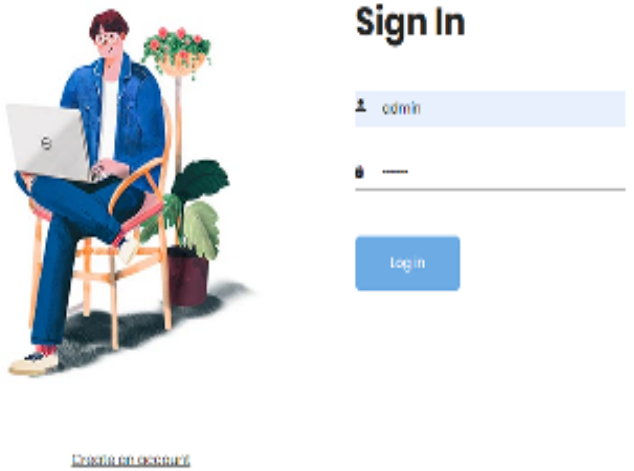


OUTPUT SCREEENSHOTS:

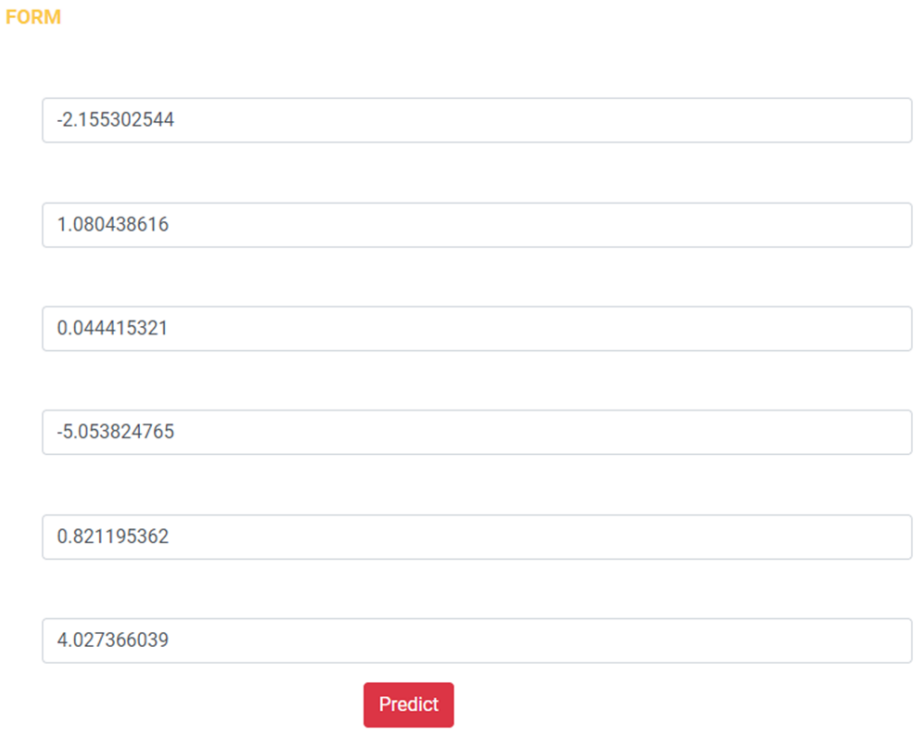


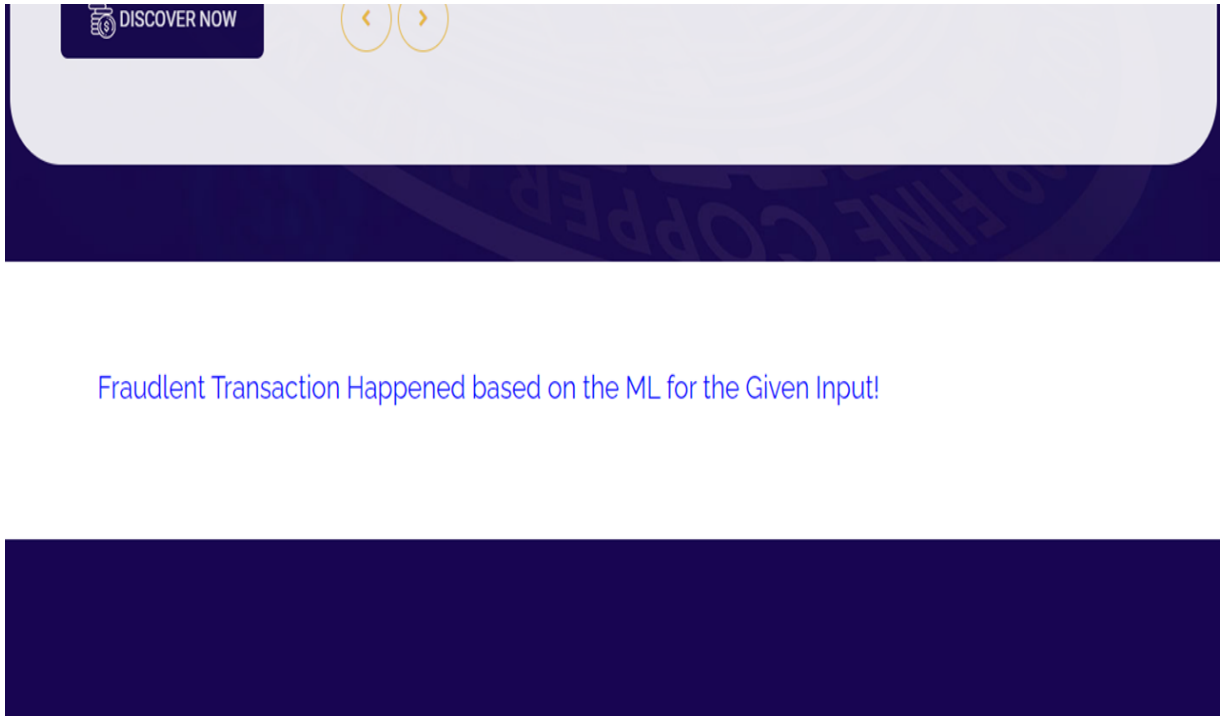
OUTPUT SCREEENSHOTS:



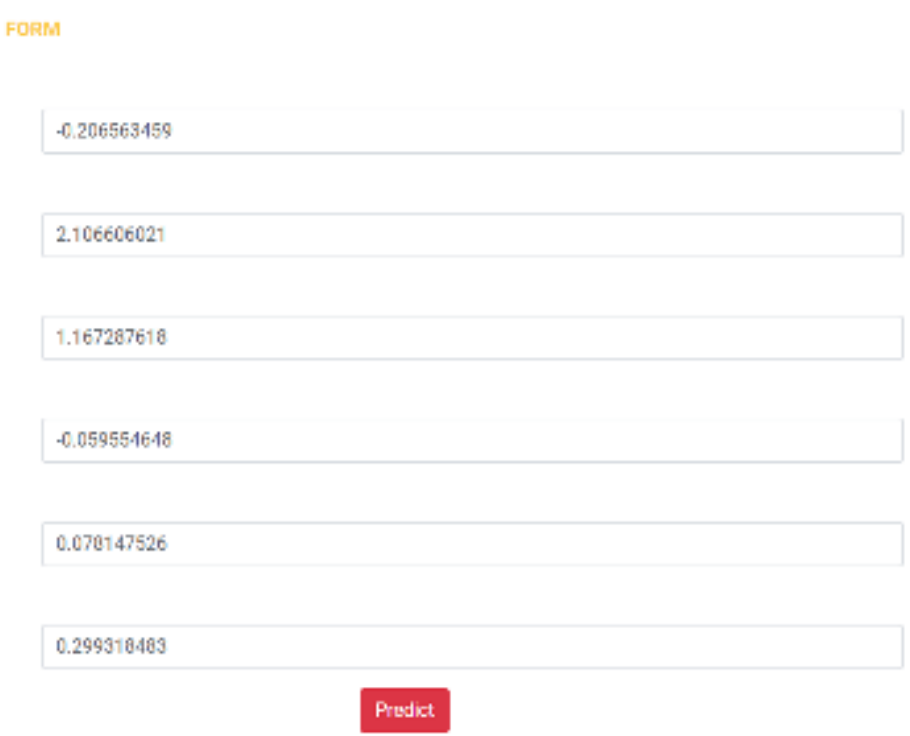


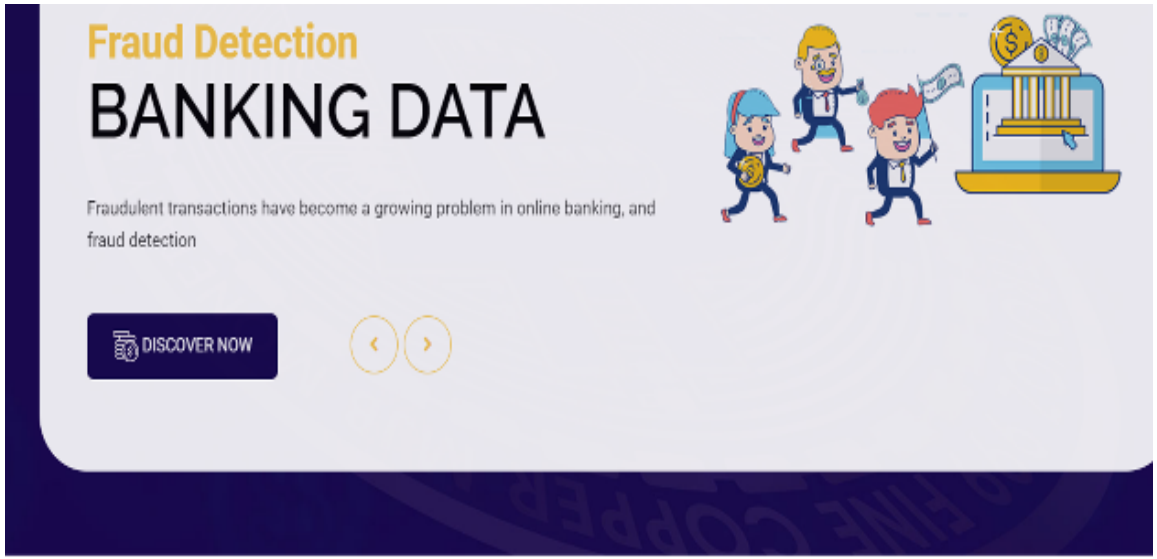
OUTPUT SCREEENSHOTS:





OUTPUT SCREEENSHOTS:





NON-Fraudlent Transaction Happened based on the ML for the Given Input!

CONCLUSION:

- The Stacking Classifier performed exceptionally well, achieving the highest accuracy compared to other models.
- The project showed strong performance across various machine learning models, including LightGBM, XGBoost, CatBoost, and neural networks.
- Using different sampling and scaling techniques improved the accuracy of fraud detection.
- By using an ensemble technique like the Stacking Classifier, we enhanced the accuracy further.
- A user-friendly Flask interface with secure authentication was added to improve the user experience during testing.
- The project's results show promise for applying advanced machine learning techniques to detect fraud in the banking sector.
- The project can be continuously improved by exploring more ensemble techniques and optimizations.
- The project helps the banking industry by strengthening fraud detection, reducing financial losses, and ensuring secure transactions.

FUTURE SCOPE:

- Future research will look into combining more hybrid models with CatBoost to improve fraud detection.
- Future studies will focus on adjusting hyperparameters, especially the number of trees in CatBoost, to optimize performance.
- Future work will aim to develop strategies to keep up with changing fraud patterns to maintain the model's effectiveness.
- Ongoing research will explore ways to integrate real-time data to improve the system's responsiveness and adaptability.
- Future efforts will aim to make the model easier to understand, providing clearer insights into how it makes decisions.

22

REFERENCES:

I. Matloob, S. A. Khan, R. Rukaiya, M. A. K. Khattak, and A. Munir, “A sequence mining-based novel architecture for detecting fraudulent transactions in healthcare systems,” IEEE Access, vol. 10, pp. 48447–48463, 2022.

K. Gupta, K. Singh, G. V. Singh, M. Hassan, G. Himani, and U. Sharma, “Machine learning based credit card fraud detection—A review,” in Proc. Int. Conf. Appl. Artif. Intell. Comput. (ICAAIC), 2022, pp. 362–368.

R. Almutairi, A. Godavarthi, A. R. Kotha, and E. Ceesay, “Analyzing credit card fraud detection based on machine learning models,” in Proc. IEEE Int. IoT, Electron. Mechatronics Conf. (IEMTRONICS), Jun. 2022, pp. 1–8.

THANK YOU