

CREDIT CARD FRAUD DETECTION

CSE 587

Phase 3

Team

Sri Harshitha Palla
Venkata Subrahmanyam

50469733
50465653

spalla3
vupputur

Problem Statement

With the increase in online transactions and e-commerce platforms, credit card fraud has become more common than before. The main objective of this project is to accurately detect fraudulent credit card transactions by using of various Machine Learning Algorithms. We will be using the Credit Card Transactions Fraud Detection Dataset from Kaggle[1] to achieve this.

A. DISCUSS THE BACKGROUND OF THE PROBLEM LEADING TO YOUR OBJECTIVES. WHY IS IT A SIGNIFICANT PROBLEM?

Credit card fraud can result in significant financial losses for both cardholders and financial institutions. Fraudulent transactions can be very large, and often go unnoticed until the victim receives their credit card statement. By detecting and preventing fraud, financial institutions can save themselves and their customers from financial losses. Overall, credit card fraud detection is significant because it helps financial institutions to protect themselves and their customers from financial losses, maintain their reputation, comply with laws and regulations, and provide a better customer experience.

B. EXPLAIN THE POTENTIAL OF YOUR PROJECT TO CONTRIBUTE TO YOUR PROBLEM DOMAIN. DISCUSS WHY THIS CONTRIBUTION IS CRUCIAL?

Financial institutions process large volumes of credit card transactions every day, and data science techniques can help them to quickly and accurately analyse this data to detect and prevent fraud. This project uses machine learning algorithms that can identify patterns and anomalies in credit card transactions that may be indicative of fraudulent activity. These techniques can help financial institutions to identify fraudulent transactions more accurately and efficiently than traditional manual methods.

Models used in the Product

Based on accuracies obtained, we have used the following top 3 models for predictions in our Final Product

- Logistic Regression,

	Accuracy	Precision	Recall	F1 Score
0	0.994549	0.037383	0.002944	0.005459

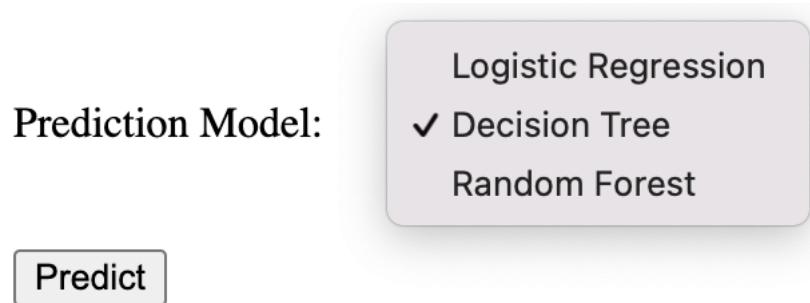
- Decision Tree and

	Accuracy	Precision	Recall	F1 Score
0	0.998632	0.918239	0.803178	0.856863

- Random Forest

	Accuracy	Precision	Recall	F1 Score
0	0.998604	0.962617	0.755501	0.846575

The choice of model is given to the user through the User-Interface in the form of a dropdown.



Storing Trained Models

We used `pickle.dump()` to save the trained model

1. Logistic Regression

Pickle Dump

```
import pickle

# Save the trained model to a pickle file
with open('logreg_model.pkl', 'wb') as f:
    pickle.dump(model, f)
```

2. Decision Tree

Pickle Dump

```
# Save the trained model to a pickle file
with open('decTree_model.pkl', 'wb') as f:
    pickle.dump(model, f)
```

3. Random Forest

Pickle Dump

```
# Save the trained model to a pickle file
with open('ranFor_model.pkl', 'wb') as f:
    pickle.dump(model, f)
```

Recommendations Related to Problem Statement

What can users learn from your product?

- Users can check if any transaction is fraud or not fraud
- They can input details such as – Credit card number, amount, purchase category, gender, date of transaction.
- They also get to choose any of the 3 different models available to them as a drop down
- The prediction then shows up as a pop up once they click on predict button
- They can also look at the trends in Fraud Transactions on our Trends & Visualisation page

How does it help them solve problems related to the problem statement?

- Users doubtful about a transaction online or in-store can come to our website to validate the truthfulness of the purchase
- Our product will act as a trust system for both users and credit card issuing entities
- They now do not have to wait until they receive their monthly statement
- They can check whenever and wherever immediately
- This saves not just the user but also financial institutions from incurring loss

Ideas of extension on project?

- As this idea has the potential to act as a trust building system, various online websites and businesses can use our product as an intermediate trust layer before the completion of a purchase
- This will be helpful in proving their trustworthiness to the customers
- This can also be made as an extension to our web browser so that customers can manually check the trustworthiness of the website if they are not already using our product

Steps to run the code

1. Open .ipynb in your IDE or Jupyter Lab
2. Run all cells in the .ipynb file
3. Run *!pip install flask*
4. Run *pip install flask_cors*
5. Running flask api cell will start the flask server on <http://127.0.0.1:5000>
6. Open home.html file with a local browser
7. Play around with the website!

Product Screenshots

1. Home Page



- This is a welcome page.
- It contains static data that introduces our system to the users
- We have the menu bar to the left
- User can navigate by clicking on the menu items on the left
- The 3 menu items available are – Home, Validate Transactions, trends and Visualisations

2. Validate Transactions Page

Credit Card Fraud Detection

Home **Enter Transaction Details**

Validate Transaction Credit Card Number:

Trends & Visualisations Amount:

Hour of Transaction:

Age:

Purchase Category:

Gender (Male):

Day of Transaction:

Prediction Model:

User inputs:

- Credit card number

Credit Card Number:

- Amount

Amount:

- Hour of Transaction

Enter Transaction Details

Credit Card Number: 1891234567890123

Amount: \$100.00

Hour of Transaction: 11

Age: 29

Purchase Category: Electronics

Gender (Male): Yes

Day of Transaction: 15

Prediction Model: Random Forest

- Age

Age:

- Purchase Category

Hour of Transaction:

Age:

Purchase Category:

Gender (Male): Yes

Day of Transaction:

Prediction Model:

Food & Dining
Gas & Transport
Grocery (Online)
Grocery (Offline)
Health & Fitness
 Home
Kids & Pets
Misc (Online)
Misc (Offline)
Personal Care
Shopping (Online)
Shopping (Offline)
Travel

- Gender

Gender (Male):

Yes
No

- Day of Transaction

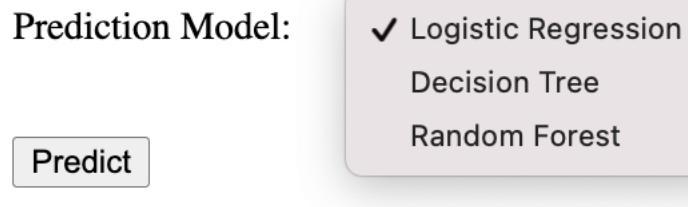
Day of Transaction:

Prediction Model:

Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday

Predict

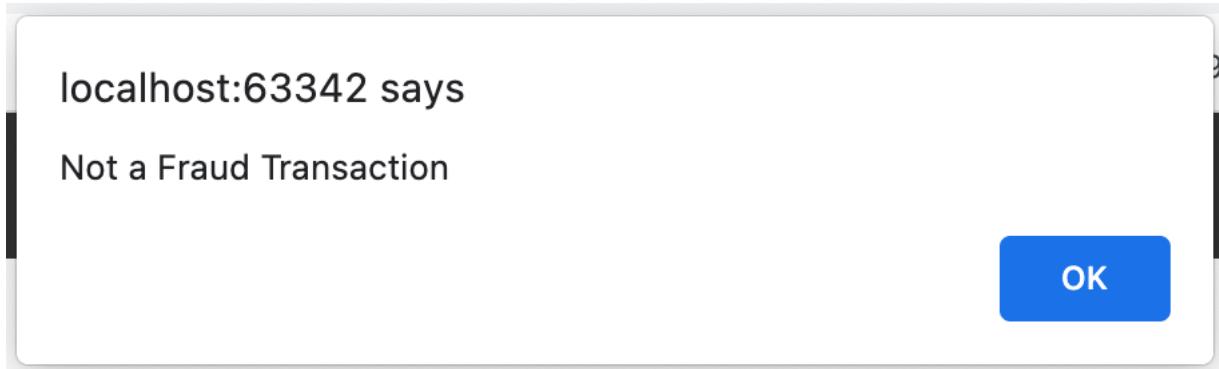
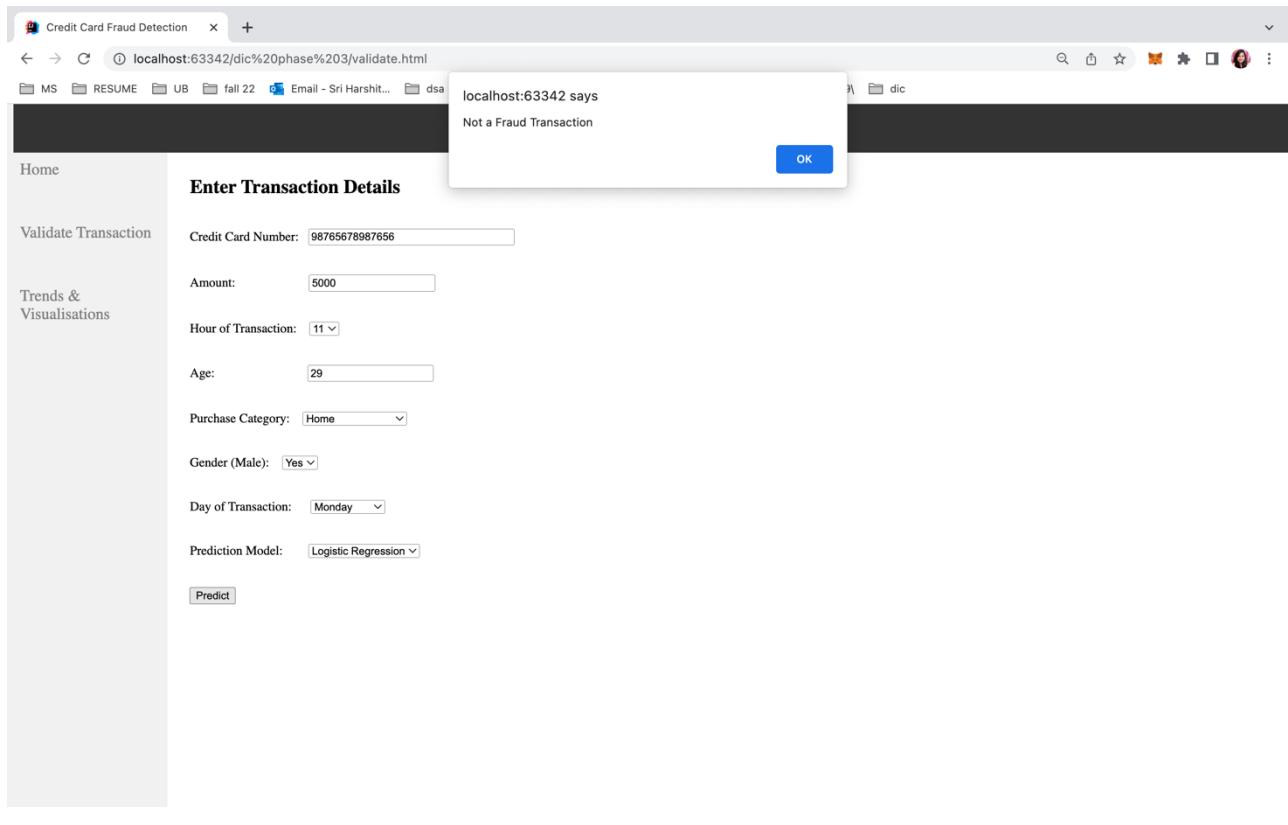
- Prediction Model



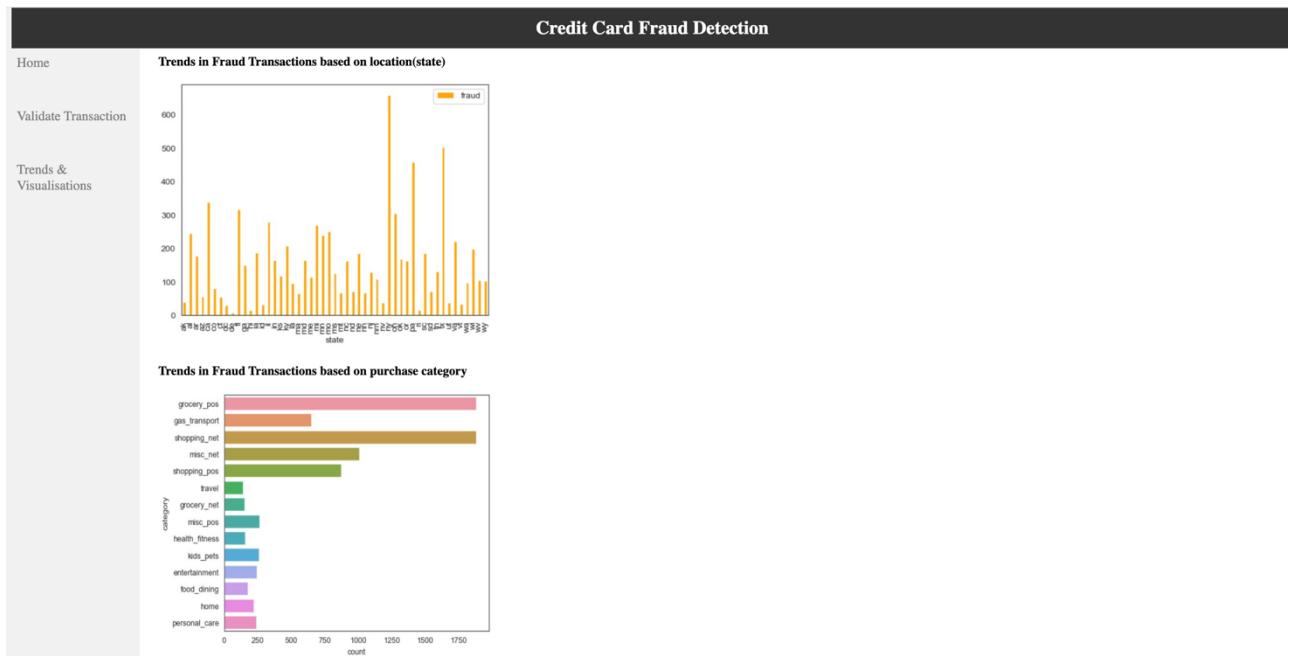
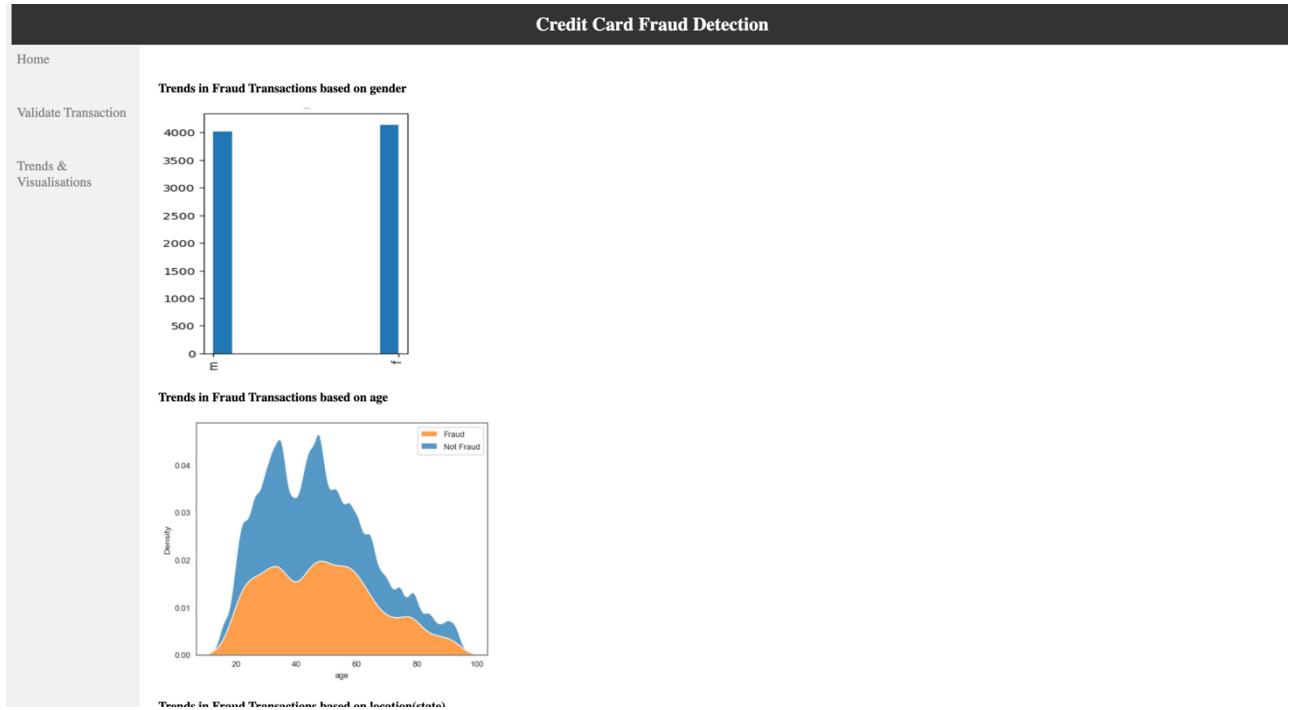
- Prediction button

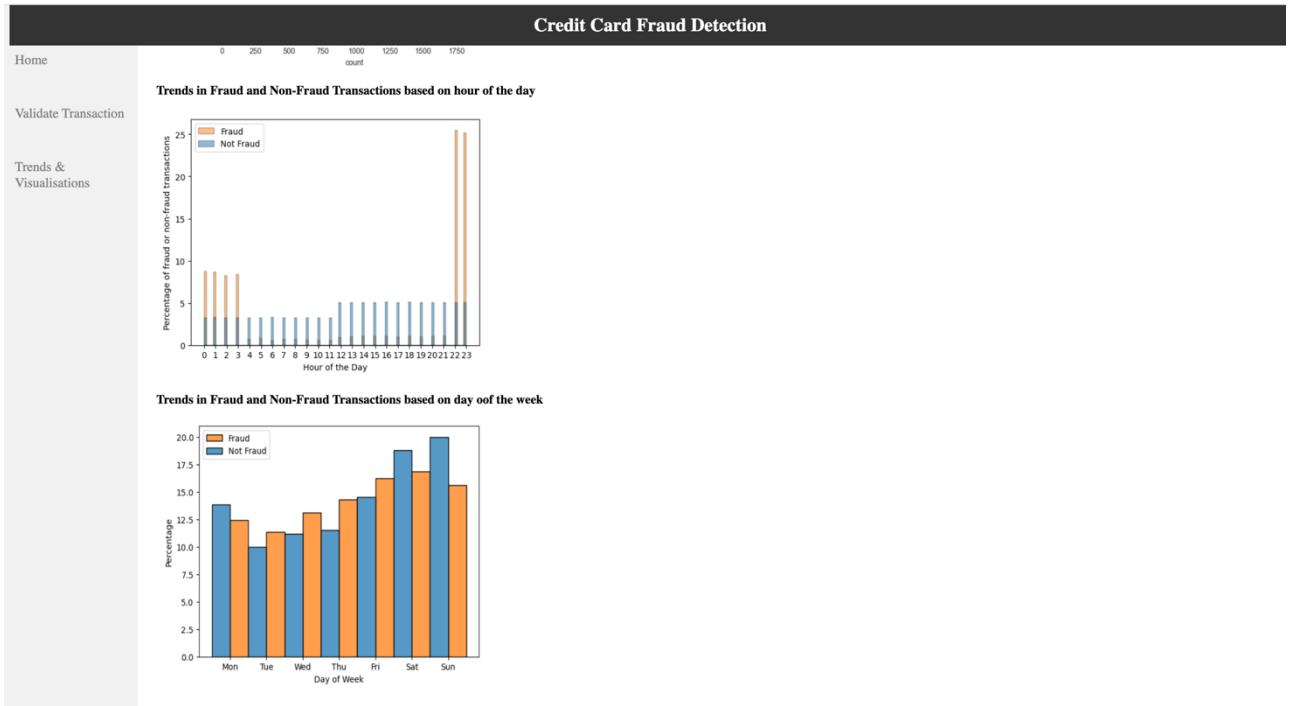
A screenshot of a user interface showing a single "Predict" button.

Prediction:



3. Trends & Visualisations Page





How could a user use it to help them solve a problem/answer a question?

- User can take a look at the graphs
- Clearly, the trends indicate a particular gender or day of transaction or a day of week or some ate to most probably be a fraud transaction
- This provides insights to the user to avoid few transactions in such situations
- For example, since NY indicates highest number of Fraud transactions, the user can be extra careful while transacting there
- Our system clearly exhibits its purpose in case of offline transactions as well with the above provided example

References

- [1] <https://www.kaggle.com/datasets/kartik2112/fraud-detection>
- [2] <https://nordicapis.com/how-to-create-an-api-using-the-flask-framework/>
- [3] <https://pythontic.com/modules/pickle/dump>
- [4] https://www.w3schools.com/jsref/met_element_addeventlistener.asp
- [5] https://www.w3schools.com/html/html_attributes.asp
- [6] [python - How to use pickle to save sklearn model - Stack Overflow](#)
- [7] [pickle — Python object serialization — Python 3.11.3 documentation](#)
- [8] [JavaScript Fetch API Explained By Examples \(javascripttutorial.net\)](#)