CREDIT CARD FRAUD DETECTION USING MACHINE LEARNING

Priya Navale

Date: 17 May 2024

Abstract

In today's digital age, credit card fraud poses a significant threat to financial security, impacting both consumers and businesses worldwide. With the increasing sophistication of fraudulent activities, traditional methods of fraud detection are proving insufficient. Therefore, this project focuses on harnessing the power of machine learning algorithms to develop an innovative solution for detecting and preventing credit card fraud.

The primary objective of this project is to create a robust and adaptive system capable of accurately identifying fraudulent transactions in real-time while minimizing false positives. By leveraging machine learning algorithms, we aim to analyse vast amounts of transaction data to discern intricate patterns and anomalies indicative of fraudulent behaviour.

The importance of this project cannot be overstated. As fraudulent activities continue to evolve and become more sophisticated, there is an urgent need for advanced technologies to combat this growing threat. By employing machine learning algorithms, we can build intelligent systems capable of detecting fraudulent transactions with a high degree of accuracy, thereby mitigating financial losses and preserving the trust and integrity of financial institutions.

This introduction provides a glimpse into the objectives of our project, highlighting the significance of utilizing machine learning algorithms in the fight against credit card fraud. Through innovative approaches and cutting-edge technologies, we aim to develop a solution that not only enhances security but also empowers businesses and consumers to transact with confidence in an increasingly digital world.

1.Problem Statement

The problem we're tackling is credit card fraud, where people try to make unauthorized transactions using stolen card information. Our goal is to build a smart computer system that can quickly spot these fraudulent transactions while making sure not to mistake regular transactions as fraud. By doing this, we aim to protect both businesses and individuals from financial losses and maintain trust in the credit card system.

2. Introduction

Nowadays Credit card usage has been drastically increased across the world, now people believe in going cashless and are completely dependent on online transactions. The

credit card has made the digital transaction easier and more accessible. A huge number of dollars of loss are caused every year by the criminal credit card transactions. Fraud is as old as mankind itself and can take an unlimited variety of different forms. The PwC global economic crime survey of 2017 suggests that approximately 48% of organizations experienced economic crime. Therefore, there's positively a necessity to unravel the matter of credit card fraud detection. Moreover, the growth of new technologies provides supplementary ways in which criminals may commit a scam. The use of credit cards is predominant in modern day society and credit card fraud has been kept on increasing in recent years. Huge Financial losses have been fraudulent effects on not only merchants and banks but also the individual person who are using the credits. Fraud may also affect the reputation and image of a merchant causing non-financial losses that. For example, if a cardholder is a victim of fraud with a certain company, he may no longer trust their business and choose a competitor. Fraud Detection is the process of monitoring the transaction behaviour of a cardholder to detect whether an incoming transaction is authentic and authorized or not otherwise it will be detected as illicit. In a planned system, we are applying the random forest algorithm for classifying the credit card dataset. Random Forest is an associate in the nursing algorithmic program for classification and regression. Hence, it is a collection of decision tree classifiers. The random forest has an advantage over the decision tree as it corrects the habit of over fitting to their training set. A subset of the training set is sampled randomly so that to train each individual tree and then a decision tree is built, each node then splits on a feature designated from a random subset of the complete feature set. Even for large data sets with many features and data instances, training is extremely fast in the random forest and because each tree is trained independently of the others. The Random Forest algorithm has been found to provide a good estimate of the generalization error and to be resistant to overfitting.

3. Market/Customer/Business Need Assessment:

Protecting Finances: Banks and businesses need to safeguard their money from fraudulent transactions that can cause significant financial losses.

Building Trust: Customers want to feel safe using credit cards. They need assurance that their transactions are secure and protected from fraud.

Following Rules: Financial institutions must comply with regulations and laws about protecting customer data and preventing fraud.

Saving Money: Preventing fraud saves companies money by avoiding losses from unauthorized transactions and reducing the costs of investigating false alarms.

Staying Ahead: Businesses want to stand out by offering advanced security features that outshine competitors and keep customers safe.

Working Efficiently: Automated fraud detection systems help businesses work faster and smarter by catching fraud in real-time and reducing manual work.

Protecting Reputation: Companies need to maintain a good reputation by preventing fraud and showing customers they care about their security and trust.

4.Target Specifications and Characterization

Our target customers are small and medium-sized enterprises (SMEs) across various industries, including retail, e-commerce, hospitality, and services. These businesses typically lack dedicated resources for fraud detection and prevention and prioritize solutions that are scalable, cost-effective, and seamlessly integrated into their existing workflows. They require user-friendly systems that require minimal manual intervention and can adapt to their evolving needs and transaction volumes.

Characterization:

- Size: Small and medium-sized enterprises (SMEs).
- Industries: Retail, e-commerce, hospitality, services, etc.
- Resources: Limited dedicated resources for fraud detection and prevention.
- Priorities: Scalable, cost-effective solutions integrated seamlessly into existing workflows.
- Challenges: Vulnerability to financial losses, reputational damage, and operational disruptions due to fraudulent activities.
- Objectives: Mitigate fraud risks, protect financial assets, preserve customer trust, and maintain operational efficiency to sustain business growth.

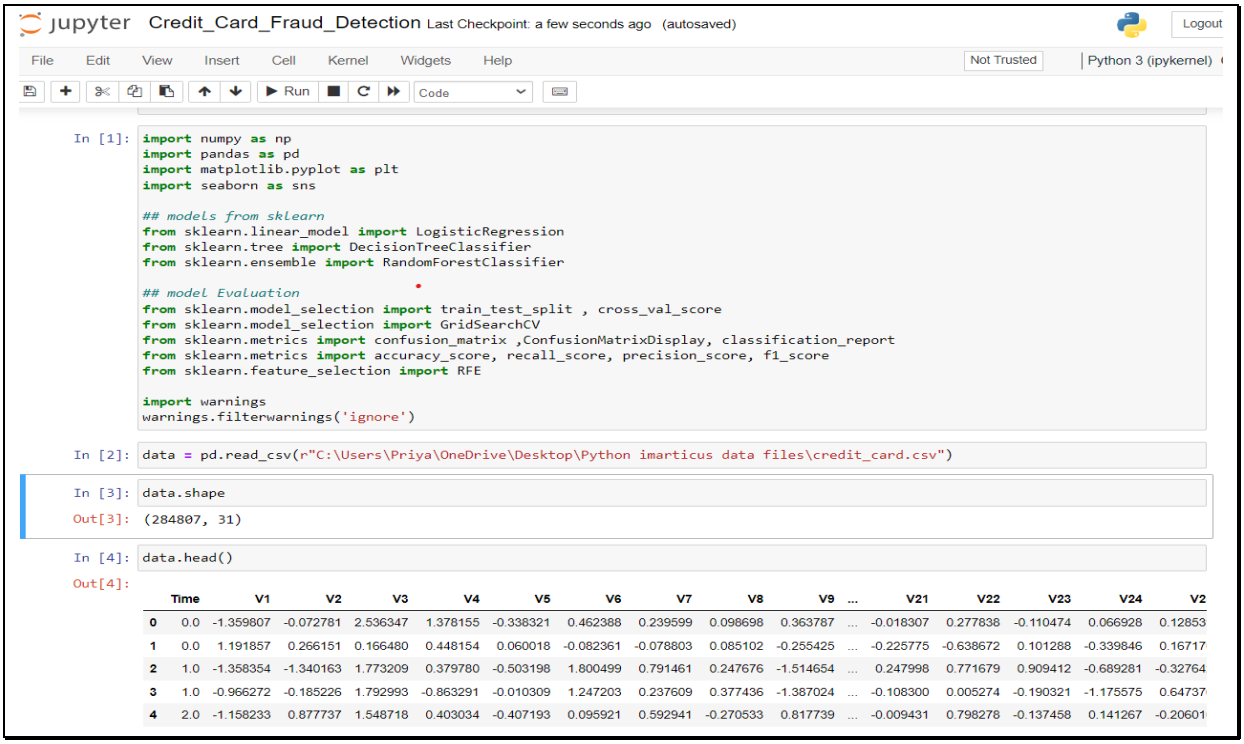
5. External Search (information sources/references)

I use the Credit card fraud detection dataset for this project Dataset can be found here: https://www.kaggle.com/code/codeaesthete/credit-card-fraud-detection-deep-neural-networks/input
The Dataset can be found on the kaggle. The dataset consists of feature vectors belonging to 2,84,807 sessions.

Relevant Papers/reports:
https://sjcit.ac.in/wp-content/uploads/2022/11/00905519CS40319CS404.pdf

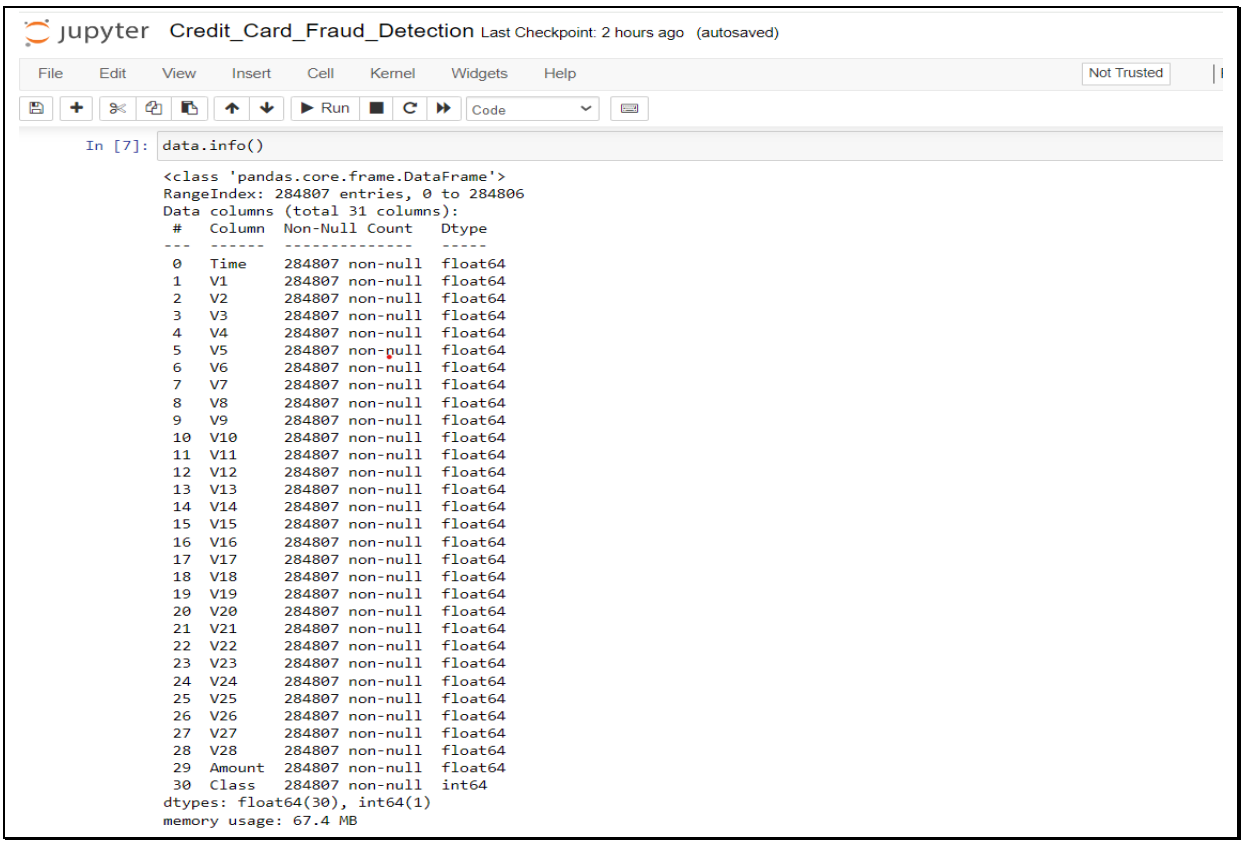
- Credit card fraud techniques: https://spd.tech/machine-learning/credit-card-fraud-detection/
- https://www.geeksforgeeks.org/ml-credit-card-fraud-detection/
- https://www.sciencedirect.com/science/article/pii/S2772662223000036
- https://www.infosysbpm.com/blogs/bpm-analytics/machine-learning-for-credit-card-fraud-detection.html
- https://link.springer.com/article/10.1007/s44230-022-00004-0
- https://stripe.com/in/resources/more/how-machine-learning-works-for-payment-fraud-detection-and-prevention

Let’s view dataset:



	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V2
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	...	-0.018307	0.277838	-0.110474	0.066928	0.12853
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	...	-0.225775	-0.638672	0.101288	-0.339846	0.16717
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	...	0.247998	0.771679	0.909412	-0.689281	-0.32764
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	...	-0.108300	0.005274	-0.190321	-1.175575	0.64737
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	...	-0.009431	0.798278	-0.137458	0.141267	-0.20601

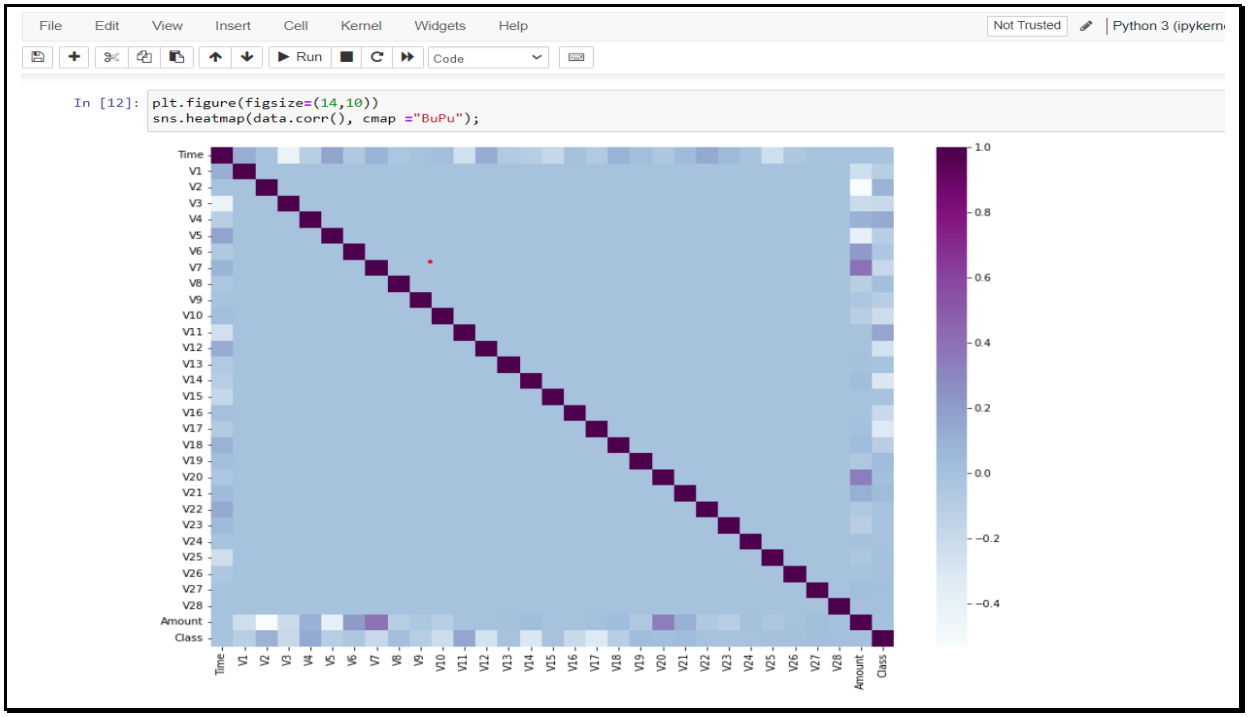
See some information about our dataset:



6. Benchmarking

- Accuracy Benchmarking: Compare the accuracy of the developed fraud detection system with industry benchmarks or established standards. This involves evaluating the system's ability to correctly classify transactions as fraudulent or legitimate. Industry benchmarks or standards may vary, but achieving a high level of accuracy (e.g., above 95%) is generally desirable.
- Performance Benchmarking: Assess the system's performance metrics such as precision, recall and F1-score. Benchmark these metrics against industry standards or existing solutions to ensure that the developed system meets or exceeds performance
- Expectations Scalability Test: We see how well our system handles more and more transactions. It should keep working smoothly even as the number of transactions grows.
- Adaptability Test: We test if our system can learn about new fraud tricks and adapt to them. It needs to stay ahead of the bad guys.
- Threshold Optimization Benchmarking: Benchmark different classification thresholds to optimize the system's sensitivity to false positives and false negatives. Evaluate the impact of threshold adjustments on performance metrics such as precision, recall, and overall accuracy.
- Understanding Test: We make sure our system can explain why it thinks a transaction might be fraudulent. It should be easy to understand how it makes decisions.
- Following Rules Test: We check if our system follows all the regulations and laws about protecting customer data and preventing fraud.
- Compatibility Test: We make sure our system works well with other systems and databases that banks and companies already use.
- Cost-Effectiveness Test: We check if our system is worth the money. It should do a great job of catching fraud without costing too much to run

Correlation Matrix for Credit Card Data: The correlations between all the of the attributes within the dataset are presented in the figure below.



7. Applicable Regulations

- Payment Card Industry Data Security Standard (PCI DSS): PCI DSS sets forth requirements for the secure processing, transmission, and storage of cardholder data. Compliance with PCI DSS is crucial for any entity involved in credit card transactions, including shopkeepers and vendors collecting payment data.

- General Data Protection Regulation (GDPR) (if applicable): GDPR mandates the protection of personal data of individuals within the European Union (EU). Compliance with GDPR is necessary when collecting, processing, or storing personal data, including credit card information, of EU residents.

- Fair Credit Reporting Act (FCRA) (if applicable): FCRA regulates the collection, dissemination, and use of consumer credit information. Compliance with FCRA is necessary when using credit reports or credit-related data for fraud detection or credit scoring purposes.

- Electronic Funds Transfer Act (EFTA): EFTA establishes rights, liabilities, and responsibilities of consumers and financial institutions regarding electronic fund transfers, including credit card transactions. Compliance with EFTA is essential for ensuring fair and transparent electronic payment systems.

- Anti-Money Laundering (AML) Regulations: AML regulations aim to prevent the use of financial systems for money laundering and terrorist financing activities. Compliance with AML regulations may involve implementing measures to detect and report suspicious transactions, including fraudulent credit card transactions.

- Consumer Protection Laws: Various consumer protection laws may apply to credit card transactions, ensuring fair treatment of consumers and prohibiting deceptive or unfair practices. Compliance with consumer protection laws is crucial for maintaining trust and credibility in credit card transactions.

- Contractual Agreements: Any contractual agreements between payment processors, financial institutions, and merchants may impose specific obligations and requirements related to credit card fraud detection and data security. Compliance with contractual agreements is essential for maintaining business relationships and avoiding legal disputes.

8. Applicable Constraints

- Limited physical area for equipment/setup.
- Financial limitations on resources.
- Lack of technical knowledge or skills.
- Limited time for implementation or training.
- Difficulty in accessing relevant data.
- Compliance with industry regulations.
- Ability to expand or adjust system as needed.
- Limitations in existing technology infrastructure.

- Compatibility with existing systems.
- Ensuring data and system security measures.

9. Business Opportunity

- Enhanced Security Solutions: By implementing advanced machine learning algorithms, businesses can offer enhanced security solutions to their customers, protecting them from fraudulent activities. This not only safeguards the financial interests of customers but also strengthens the reputation of the business as a trusted provider of secure payment services.

- Reduced Financial Losses: Detecting and preventing credit card fraud in real-time can significantly reduce financial losses for businesses. By implementing effective fraud detection systems, organizations can minimize the impact of fraudulent transactions on their bottom line, leading to improved financial performance and stability.

- Improved Customer Trust: Investing in robust fraud detection measures demonstrates a commitment to customer security and safety. This fosters trust and confidence among customers, leading to increased loyalty and retention rates. Customers are more likely to continue using the services of businesses they trust to protect their financial information.

- Competitive Advantage: Businesses that excel in fraud detection can gain a competitive advantage in the market. By offering superior security features and demonstrating a proactive approach to fraud prevention, organizations can differentiate themselves from competitors and attract new customers who prioritize security in their financial transactions.

- Monetization Opportunities: Organizations with expertise in credit card fraud detection can capitalize on their knowledge by offering fraud prevention services to other businesses. This creates additional revenue streams and expands the market reach of the organization beyond its core offerings.

- Compliance with Regulations: Implementing effective fraud detection systems helps businesses comply with regulatory requirements related to data security and privacy. Adherence to regulations such as PCI DSS and GDPR not only mitigates legal risks but also enhances the reputation of the business as a responsible and trustworthy entity.

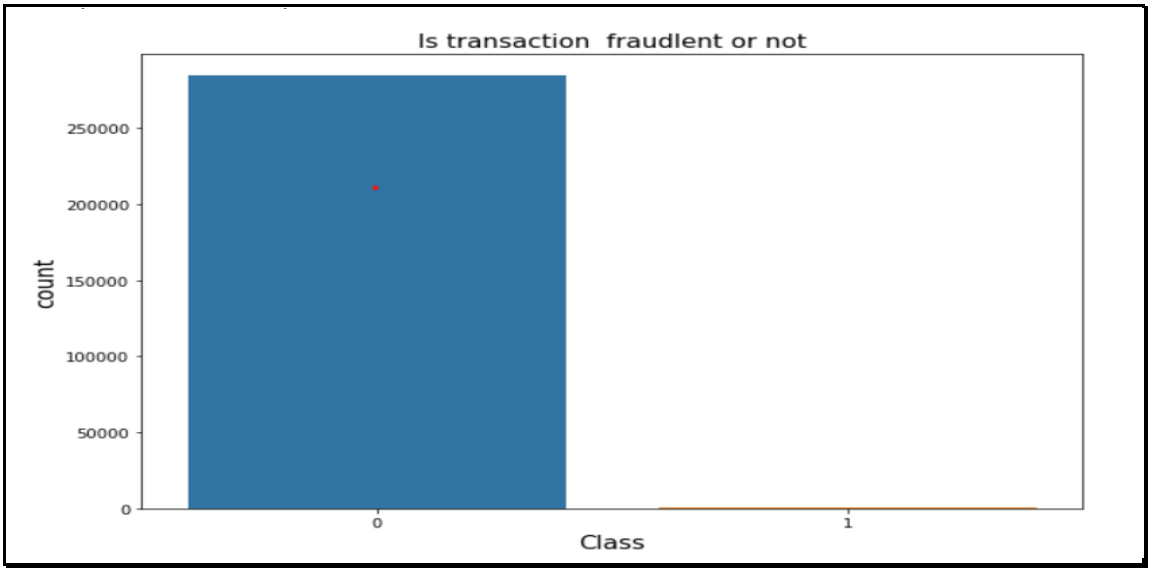
- Data Insights and Analytics: The data collected during the fraud detection process can provide valuable insights into customer behaviour and transaction patterns. By analysing this data, businesses can identify trends, detect emerging threats, and make informed decisions to optimize their fraud prevention strategies.

- Cross-Selling and Upselling Opportunities: Fraud detection projects can serve as a gateway to cross-selling and upselling opportunities, allowing businesses to offer additional security features or related services to clients concerned about fraud prevention.

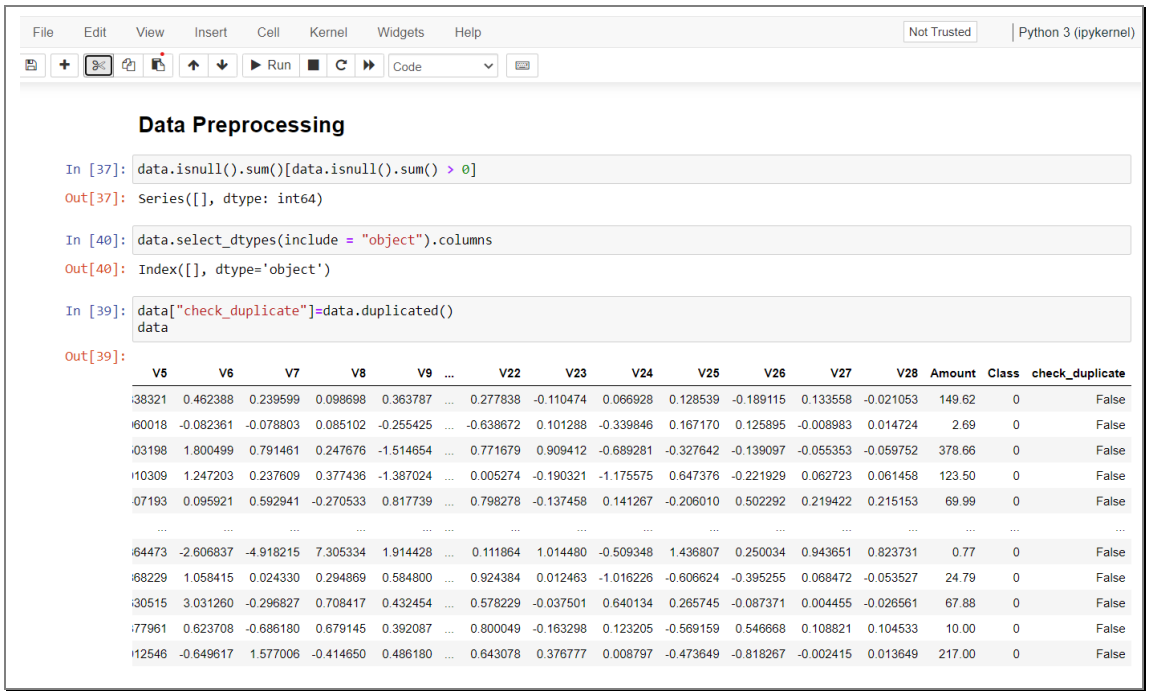
10. Concept Generation

This product requires the tool of machine learning models to be written from scratch in order to suit our needs. Tweaking these models for our use is less daunting than coding it up from scratch. A well trained model can either be repurposed or built. But building a model with the resources and data we have is dilatory but possible. Begin by understanding the nature and scope of credit card fraud, including common fraud schemes, typical fraudulent behaviours, and potential vulnerabilities in the payment process.

1. Data Analysis



2. Data Preprocessing



V5	V6	V7	V8	V9	...	V22	V23	V24	V25	V26	V27	V28	Amount	Class	check_duplicate
38321	0.462388	0.239599	0.098698	0.363787	...	0.277838	-0.110474	0.066928	0.128539	-0.189115	0.133558	-0.021053	149.62	0	False
60018	-0.082361	-0.078803	0.085102	-0.255425	...	-0.638672	0.101288	-0.339846	0.167170	0.125895	-0.008983	0.014724	2.69	0	False
03198	1.800499	0.791461	0.247676	-1.514654	...	0.771679	0.909412	-0.689281	-0.327642	-0.139097	-0.055353	-0.059752	378.66	0	False
10309	1.247203	0.237609	0.377436	-1.387024	...	0.005274	-0.190321	-1.175575	0.647376	-0.221929	0.062723	0.061458	123.50	0	False
07193	0.095921	0.592941	-0.270533	0.817739	...	0.798278	-0.137458	0.141267	-0.206010	0.502292	0.219422	0.215153	69.99	0	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
64473	-2.606837	-4.918215	7.305334	1.914428	...	0.111864	1.014480	-0.509348	1.436807	0.250034	0.943651	0.823731	0.77	0	False
68229	1.058415	0.024330	0.294869	0.584800	...	0.924384	0.012463	-1.016226	-0.606624	-0.395255	0.068472	-0.053527	24.79	0	False
30515	3.031260	-0.296827	0.708417	0.432454	...	0.578229	-0.037501	0.640134	0.265745	-0.087371	0.004455	-0.026561	67.88	0	False
77961	0.623708	-0.686180	0.679145	0.392087	...	0.800049	-0.163298	0.123205	-0.569159	0.546668	0.108821	0.104533	10.00	0	False
12546	-0.649617	1.577006	-0.414650	0.486180	...	0.643078	0.376777	0.008797	-0.473649	-0.818267	-0.002415	0.013649	217.00	0	False

3. Train_Test_Split the Data in train_x, test_x, train_y, test_y (Data Sampling)

Data Sampling

```
In [13]: from sklearn.model_selection import train_test_split

         data_train, data_test = train_test_split(data, test_size = 0.3)
```

```
In [14]: data_train_x = data_train.iloc[: , 0:-1]
         data_train_y = data_train.iloc[: , -1]

         data_test_x = data_test.iloc[: , 0:-1]
         data_test_y = data_test.iloc[: , -1]
```

4. Data Modelling

We will use five different models and we will finalize the model which will give good accuracy

1) Decision Tree Classifier

A decision tree is a type of supervised machine learning used to categorize or make predictions based on how a previous set of questions were answered. The model is a form of supervised learning, meaning that the model is trained and tested on a set of data that contains the desired categorization.

```
In [21]: ## classification report of DecisionTree

         print(classification_report(data_test_y , pred_test_dt) )
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     85292
           1       0.71      0.74      0.72       151

    accuracy                           1.00     85443
   macro avg       0.86      0.87      0.86     85443
weighted avg       1.00      1.00      1.00     85443
```

2) Logistic Regression

Logistic Regression model is statical model where evaluations are formed of the connection among dependent qualitative variable (binary or binomial logistic regression) or variable with three values or higher (multinomial logistic regression) and one independent explanatory variable or higher whether qualitative or quantitative

```
In [17]: ## classification report of Logistics Regression

         print(classification_report(data_test_y , pred_test) )
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     85292
           1       0.74      0.65      0.69       151

    accuracy                           1.00     85443
   macro avg       0.87      0.82      0.84     85443
weighted avg       1.00      1.00      1.00     85443
```

3) Random Forest Classifier

Random forest is a supervised machine learning algorithm based on ensemble learning. Ensemble learning is an algorithm where the predictions are derived by assembling or bagging different models or similar model multiple times. The random forest algorithm works in a similar way and uses multiple algorithms i.e. multiple decision trees, resulting in a forest of trees, hence the name "Random Forest". The random forest algorithm can be used for both regression and classification tasks.

```
In [25]: ## classification report of RandomForestClassifier
         print(classification_report(data_test_y , pred_test_rfc) )
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     85292
           1       0.96      0.77      0.85       151

    accuracy                           1.00     85443
   macro avg       0.98      0.88      0.93     85443
weighted avg       1.00      1.00      1.00     85443
```

4) Support Vector Machine

SVM works by mapping data to a high-dimensional feature space so that data points can be categorized, even when the data are not otherwise linearly separable. A separator between the categories is found, then the data are transformed in such a way that the separator could be drawn as a hyperplane Training regression model and finding out the best one.

```
In [187]: ## classification report of SVM

          print(classification_report(data_test_y , pred_test_svm) )
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     85308
           1       0.96      0.63      0.76       135

    accuracy                           1.00     85443
   macro avg       0.98      0.81      0.88     85443
weighted avg       1.00      1.00      1.00     85443
```

5) K-Nearest Neighbours (KNN)

The K-Nearest Neighbour algorithm (KNN) is a supervised ML technique that can be applied in both scenario instances, classification instances along with regression instances.

```
In [33]: ## classification report of KNN

         print(classification_report(data_test_y , pred_test_knn) )
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     85292
           1       0.90      0.06      0.11       151

    accuracy                           1.00     85443
   macro avg       0.95      0.53      0.56     85443
weighted avg       1.00      1.00      1.00     85443
```

By analysing all the models, we can say that Random Forest is giving good results.

Sr. No	Models	Accuracy
1	Decision Tree Classifier	99.90
2	Logistic Regression	99.89
3	Random Forest Classifier	99.95
4	Support Vector Machine	99.82
5	K-Nearest Neighbor (KNN)	99.83

So we Finalize Random Forest model For Our Model Training And Model Deployment

3.Random Forest

```
In [21]: from sklearn.ensemble import RandomForestClassifier
         rfc = RandomForestClassifier()

         rfc.fit(data_train_x , data_train_y)
         pred_test_rfc = rfc.predict(data_test_x)

         mat_rfc = confusion_matrix(data_test_y , pred_test_rfc)
         mat_rfc
```

```
Out[21]: array([[85268,     6],
                [   34,   135]], dtype=int64)
```

```
In [22]: rfc_acc = accuracy_score(data_test_y , pred_test_rfc)
         rfc_acc
```

```
Out[22]: 0.9995318516437859
```

```
In [23]: ## classification report of RandomForestClassifier
         print(classification_report(data_test_y , pred_test_rfc) )
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     85274
           1       0.96      0.80      0.87       169

    accuracy                           1.00     85443
   macro avg       0.98      0.90      0.94     85443
weighted avg       1.00      1.00      1.00     85443
```

```
In [40]: var1 = ConfusionMatrixDisplay(mat_rfc , display_labels = ['Non_Fraudulent', 'Fraudulent'])
         var1.plot();
```

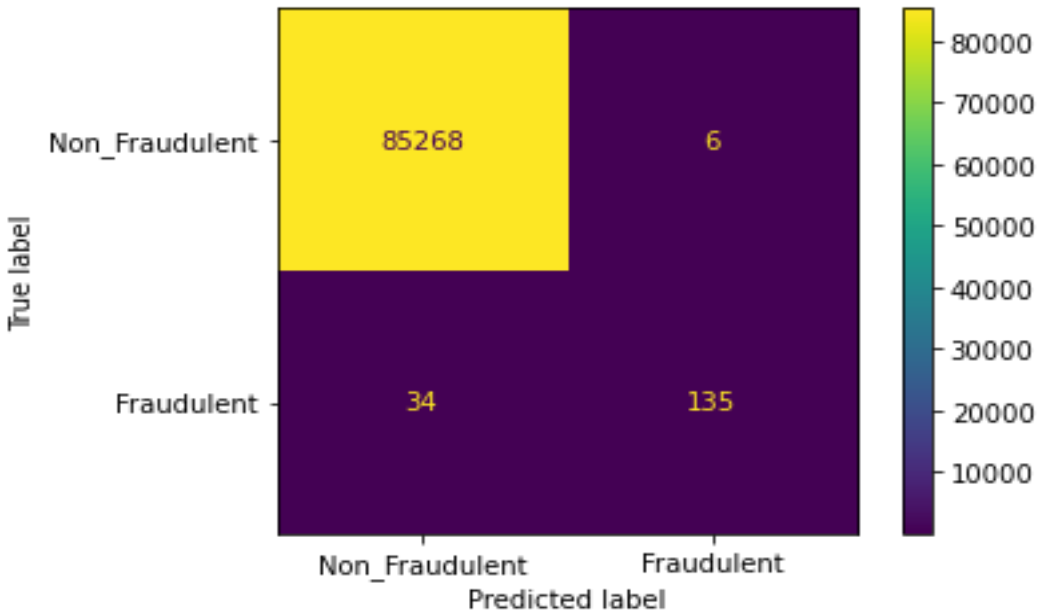


Fig. Confusion matrix for test dataset

11. Concept Development

1. Problem Identification: Identify the increasing cases of credit card fraud globally, which cause financial losses for both customers and banks.

2. Objective Definition: Develop a machine learning system to accurately detect fraudulent transactions in real-time, minimizing financial losses and maintaining trust in the banking system.

3. Data Collection: Gather a large dataset of credit card transactions, including both non-fraudulent and fraudulent ones, from various sources.

4. Data Preprocessing: Clean the data, handle missing values, and normalize features to prepare it for machine learning algorithms.

5. Feature Engineering: Extract relevant features from the transaction data, such as transaction amount, location, time, frequency, and previous transaction history.

6. Model Selection: Choose appropriate machine learning algorithms such as logistic regression, decision trees, random forests, or deep learning models like neural networks.

7. Model Training: Train the selected models on the pre-processed data, using techniques like cross-validation to optimize performance and prevent overfitting.

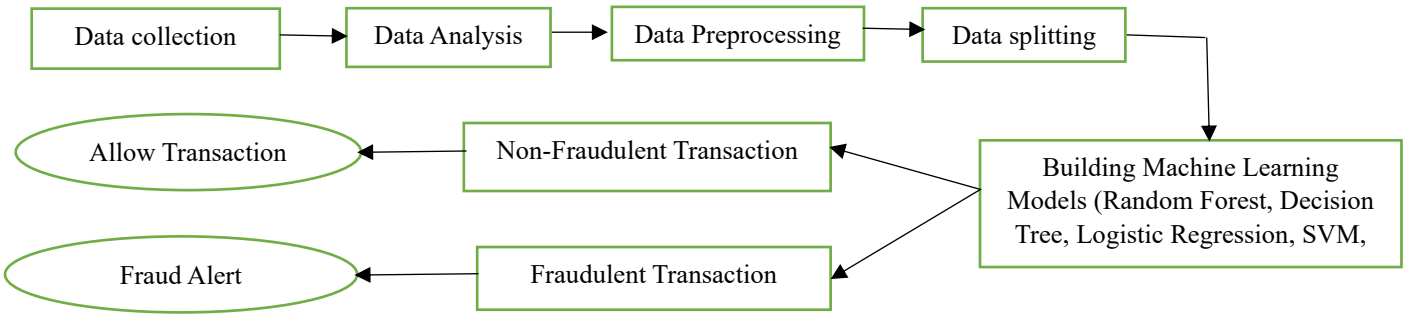
8. Evaluation Metrics: Define evaluation metrics such as accuracy, precision, recall, and F1-score to assess the performance of the trained models.

9. Model Evaluation: Evaluate the performance of each model using the defined metrics and select the one with the highest accuracy and lowest false positive rate.

10. Deployment: Deploy the selected model into the banking system's infrastructure, integrating it with real-time transaction processing systems for continuous monitoring.

11. Monitoring and Maintenance: Continuously monitor the performance of the deployed model, retraining it periodically with new data to adapt to evolving fraud patterns and maintain effectiveness over time.

The concept can be developed by using the appropriate API (flask in this case) and using Django as framework for the same and for its deployment, The cloud services has to be chosen accordingly to the need.



12. Final Report Prototype

The final product prototype for credit card fraud detection is a comprehensive system that combines advanced machine learning algorithms, real-time monitoring capabilities, and robust security measures. This prototype aims to detect and prevent fraudulent activities in credit card transactions efficiently and effectively. It includes a user-friendly interface for monitoring transactions, a backend system for data processing and analysis, and sophisticated fraud detection models. Additionally, the prototype incorporates multi-factor authentication, encryption techniques, and fraud alerts to enhance security and enable timely intervention. The schematic diagram below illustrates the key components and data flow within the prototype.

The product takes the following functions to perfect and provide a good result.

Back-end

Model Development: This must be done before releasing the service. A lot of manual supervised machine learning must be performed to optimize the automated tasks.

1. Performing EDA to realize the dependent and independent features.

2. Algorithm training and optimization must be done to minimize overfitting of the model and hyperparameter tuning.

Front End

1. Different user interface: The user must be given many options to choose form in terms of parameters. This can only be optimized after a lot of testing and analysis all the edge cases.
2. Interactive visualization the data extracted from the trained models will return raw and inscrutable data. This must be present in an aesthetic and an “easy to read” style.
3. Feedback system: A valuable feedback system must be developed to understand the customer’s needs that have not been met. This will help us train the models constantly.

13. Product details - How does it work?

The Tracker Web App simplifies finance management by enabling users to register, log expenses, and analyse spending patterns. Users can categorize expenses, set monthly budgets, and receive alerts when nearing limits. The app provides visualizations and reports to offer insights into spending habits, empowering users to make informed financial decisions. Additionally, users can export data for further analysis and customize settings to personalize their experience.

14. Code Implementation:

This is a Github link - https://github.com/priya2928/Feynn-Labs-Projects

15. Conclusion:

Nowadays, in the global computing environment, online payments are important, because online payments use only the credential information from the credit card to fulfill an application and then deduct money. Due to this reason, it is important to find the best solution to detect the maximum number of frauds in online systems. Accuracy, Error-rate, Sensitivity and Specificity are used to report the performance of the system to detect the fraud in the credit card. We use five machine learning algorithms are developed to detect the fraud in credit card system. To evaluate the algorithms, 70% of the dataset is used for training and 30% is used for testing. Accuracy, error rate, sensitivity and specificity are used to evaluate for different variables for five algorithms.

The comparative results show that the Random Forest performs better than the SVM, Decision Tree, KNN and Logistic Regression techniques.

With the help of web app prevents financial losses by detecting and blocking fraudulent credit card transactions in real-time, ensuring security for both issuers and cardholders.