CREDIT CARD FRAUD DETECTION

AI MINI PROJECT REPORT

18CSC305J - ARTIFICIAL INTELLIGENCE

Submitted by

Ananya Rana (RA2011003010831)

Deepti Busennagari (RA2011003010853)

Under the guidance of

M. Rajalakshmi

Assistant Professor, Department of Computer Science and Engineering

In partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

of

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

S.R.M. Nagar, Kattankulathur, Chengalpattu District

MAY'2023

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(UnderSection3ofUGCAct,1956)

BONAFIDE CERTIFICATE

Certified that Mini project report titled **"Credit Card Fraud Detection"**is the bona fide work of **Ananya Rana (RA2011003010831) and Deepti Busennagari (RA2011003010853)** who carried out the minor project under my supervision. Certified further, that to the best of my knowledge, the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE	**SIGNATURE**
M. Rajalakshmi	Dr. M. Pushpalatha
GUIDE	**HEAD OF THE DEPARTMENT**
Assistant Professor	Professor & Head
Department of Computing Technologies	Department of Computing Technologies

ACKNOWLEDGEMENT

We express our heartfelt thanks to our honorable **Vice Chancellor Dr. C. MUTHAMIZHCHELVAN,** for being the beacon in all our endeavors.

We would like to express my warmth of gratitude to our **Registrar Dr. S. Ponnusamy**, for his encouragement

We express our profound gratitude to our **Dean (College of Engineering and Technology) Dr. T. V. Gopal**, for bringing out novelty in all executions.

We would like to express my heartfelt thanks to Chairperson, School of Computing **Dr. Revathi Venkataraman**, for imparting confidence to complete my course project

We wish to express my sincere thanks to **Course Audit Professor Dr. Annapurani Panaiyappan, Professor** and **Head, Department of Networking and Communications and Course Coordinators** for their constant encouragement and support.

We are highly thankful to our my Course project Faculty **M. Rajalakshmi, Associate Proffesor, Department of Computing Technologies,** for his/her assistance, timely suggestion and guidance throughout the duration of this course project.

We extend my gratitude to our **HoD Dr. Pushpa Latha, Professor , Department of computing technologies** and my Departmental colleagues for their Support.

Finally, we thank our parents and friends near and dear ones who directly and indirectly contributed to the successful completion of our project. Above all, I thank the almighty for showering his blessings on me to complete my Course project.

ABSTRACT

This project aims to address the issue of data availability and misclassified data in credit card fraud detection, which makes it difficult to identify fraudulent transactions. The project targets the online credit card fraud segment, as successful fraud cases not only cost businesses money but also affect customer experience, potentially leading to customer loss. The project focuses on developing a model that can detect fraudulent transactions while minimizing incorrect fraud classifications to ensure that customers are not charged for items they did not purchase.

TABLE OF CONTENTS

1. INTRODUCTION

Our Projects main purpose is to make Credit Card Fraud Detection model identifying online credit card frauds. The main point of credit card fraud detection system is necessary to make our online transactionssecure. With this system, fraudsters don't have thechance to make multiple transactions on a stolen or counterfeit card before the cardholder is aware of the fraudulent activity. This model is used to identify whether a new transaction is fraudulent or not. Our aim here is to detect fraudulent transactions while minimizing the incorrect fraud classifications. Efforts are made to identify fraudulent credit card transactions so that the customers of credit card companies are not charged for items that they did not purchase.

1.1 Need of the Project

Despite advances in technology, credit card fraud continues to be a problem. Credit card fraud costs companies billions of dollars each year. While consumers are never on the hook for fraudulent charges and reporting thefts, and replacing cards can be aggravating and time consuming.

1.2 Approach

Machine learningapproach is used, as it canbring significantimprovements to theprocess of rule-based systems that were writtenby experts. Both supervised and unsupervised learning will be used for our model. The supervised learning will be used for training the model and unsupervised learning for detecting the anomalies.

1.3 Benefit

Higher accuracy of frauddetection.

Less manual work neededfor additional verification.

Fewer false declines.

Ability to identify newpatterns and adapt tochanges.

1.4 Competition

Already the industry isinvesting in a lot ofdifferent projects andapproaches for credit cardfraud. One of theapproaches can be takingeven more stringentmethods for credit card fraud prevention.

2. CUSTOMER DISCOVERY

2.1 Problem sizing

Problem:

Issue is data availability, because majority of the transactions (99.8%) are private and that makes it difficult to discover thefraudulent ones due to imbalanced data, which means that majority of the transactions are not fraudulent.

Misclassified Data is another significant problem because not all fraudulent transactions are discovered and reported.

The scammers utilize adaptive methods against the model.

User Segment:

If the credit card fraud was online andsuccessful, the business will have to reimburse the customer, plus pay achargeback fee to the bank. What is not usually appreciated by peopleoutside fraud prevention is that credit card fraud has a knock-on effect on theconsumer in terms of customer experience. All this, in turn, affects thebusiness in a different way - because itloses customers, or frustratescustomers.

Legitimate:

- Number- 11,384
- Percent - 99.24%

Fraud:

- Number- 87
- Percent - 0.76%

Total:

- Number - 11,471
- Percent - 100%

Severity:

•In 2021 total amount of cardtransactions were about 56.3 billion in numbers and out of that about 0.036%of all the transactions were fraudulent

ie. 20.268 million transactions thatsums up to almost 810.72millionineuros to the victims.

•A recent Nielson Report predicts thatcard fraud will cost the global tech industry $408.50 billion within the nextdecade.

Evolution:

- Boosters:

  Identity theft and credit card fraud are two of the most common financial crimes, and each of them saw significant growth in 2020. Partof the growth was due to the pandemic, as government benefitssuch as stimulus checks and unemployment insurance paymentswere targets of fraudsters.
- Setbacks:

While the traditional method todetect fraud worked for many years, and was the best optionavailable to banks and other service providers, it can’t keep upwith the advances in technology and our variety of modern-dayfinancial transactions. Moreover, scammers are getting moresophisticated in their fraudulent transactions.

2.2 Problem validation

Do you use credit cards?

For what purpose do you usually use your credit card for?

Do you keep track of your monthly expenses?

Do you keep track of your credit card expenses?

How often do you use your credit card for online transactions?

How frequently do you cross check your bank messages?

How aware are you about online fraud?

What precautions do you take against online fraud?

Have you ever been a victim of online fraud?

Have you noticed any unusual transaction?

What steps have you taken after the faulty transaction?

Are you aware of the steps that need to be taken after your credit card has been accessed by someone other than yourself?

3. PROJECT DESCRIPTION

The digital payments market is soaring as the world shifts towards online and card-based payment methods at a faster rate. With such a shift comes the growing issue of cybersecurity and fraud, which is more common than ever. According to a recent report, credit card fraud within the next 5 years will cause global losses of about $43 billion. Another study revealed that as many as 80% of the US credit cards currently in use have been compromised.

Enhancing credit card fraud detection is a priority for all banks and financial organizations. Thanks to machine learning (ML), credit card fraud detection is becoming easier and more efficient. ML-based fraud detection solutions can track patterns and prevent abnormal transactions.

Machine learning models can recognize unusual credit card transactions and fraud. The first and foremost step involves collecting and sorting raw data, which is then used to train the model to predict the probability of fraud. The solutions offered by machine learning for credit card fraud detection involve:

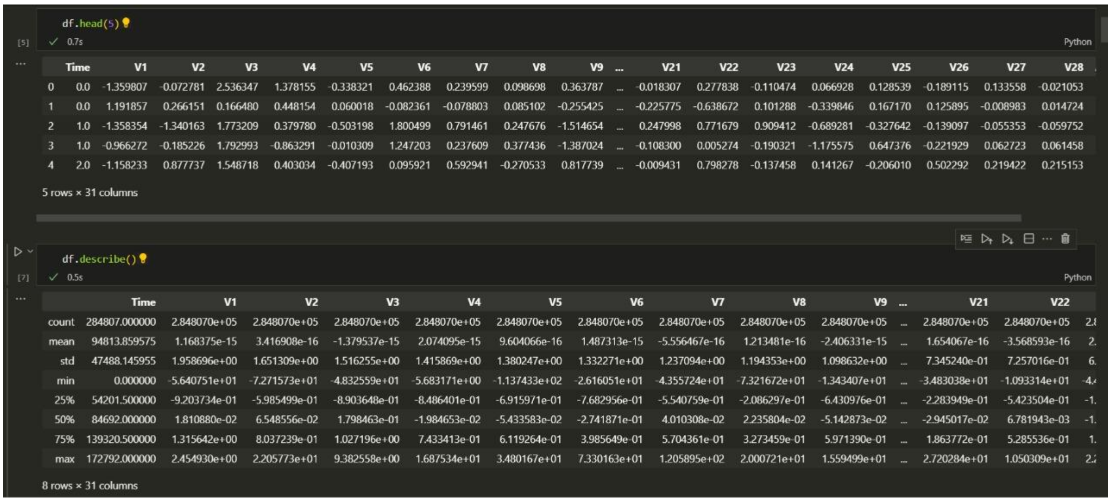
Classifying whether credit card transactions are authentic or fraudulent using algorithms such as logistic regression, random forests, Decision Trees (DTs), Extreme Gradient Boosting (XGBoost), support vector machines (SVMs), deep neural networks along with autoencoders, long short-term memory (LSTM) networks, and convolutional neural networks (CNNs)

Predicting whether it is the cardholders or the fraudsters using the credit cards through credit card profiling

Using outlier detection methods to identify considerably different transactions (or 'outliers') from regular credit card transactions to detect credit card fraud.

3.1 Illustration Input/Output:

Input:



```
df.head(5)
[5] ✓ 0.7s                                                        Python
```

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V25	V26	V27	V28
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	0.128539	-0.189115	0.133558	-0.021053
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	0.167170	0.125895	-0.008983	0.014724
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	-0.327642	-0.139097	-0.055353	-0.059752
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	0.647376	-0.221929	0.062723	0.061458
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	-0.206010	0.502292	0.219422	0.215153

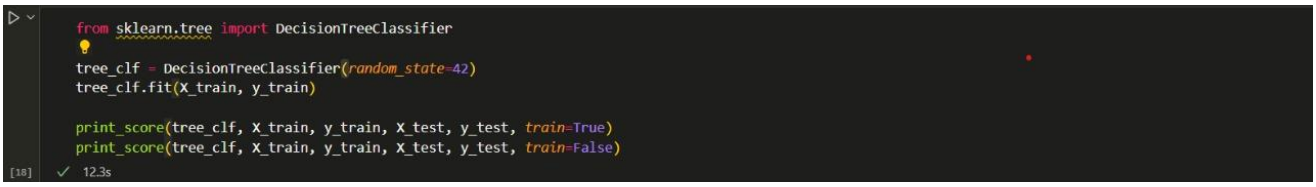
5 rows × 31 columns

```
df.describe()
[7] ✓ 0.5s                                                        Python
```

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	
count	284807.000000	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	...	2.848070e+05	2.848070e+05	2.8
mean	94813.859575	1.168375e-15	3.416908e-16	-1.379537e-15	2.074095e-15	9.604066e-16	1.487313e-15	-5.556467e-16	1.213481e-16	-2.406331e-15	...	1.654067e-16	-3.568593e-16	2.
std	47488.145955	1.958696e+00	1.651309e+00	1.516255e+00	1.415869e+00	1.380247e+00	1.332271e+00	1.237094e+00	1.194353e+00	1.098632e+00	...	7.345240e-01	7.257016e-01	6.
min	0.000000	-5.640751e+01	-7.271573e+01	-4.832559e+01	-5.683171e+00	-1.137433e+02	-2.616051e+01	-4.355724e+01	-7.321672e+01	-1.343407e+01	...	-3.483038e+01	-1.093314e+01	-4.4
25%	54201.500000	-9.203734e-01	-5.985499e-01	-8.903648e-01	-8.486401e-01	-6.915971e-01	-7.682956e-01	-5.540759e-01	-2.086297e-01	-6.430976e-01	...	-2.283949e-01	-5.423504e-01	-1.
50%	84692.000000	1.810880e-02	6.548556e-02	1.798463e-01	-1.984653e-02	-5.433583e-02	-2.741871e-01	4.010308e-02	2.235804e-02	-5.142873e-02	...	-2.945017e-02	6.781943e-03	-1.
75%	139320.500000	1.315642e+00	8.037239e-01	1.027196e+00	7.433413e-01	6.119264e-01	3.985649e-01	5.704361e-01	3.273459e-01	5.971390e-01	...	1.863772e-01	5.285536e-01	1.
max	172792.000000	2.454930e+00	2.205773e+01	9.382558e+00	1.687534e+01	3.480167e+01	7.330163e+01	1.205895e+02	2.000721e+01	1.559499e+01	...	2.720284e+01	1.050309e+01	2.2

8 rows × 31 columns

Output 1 code:

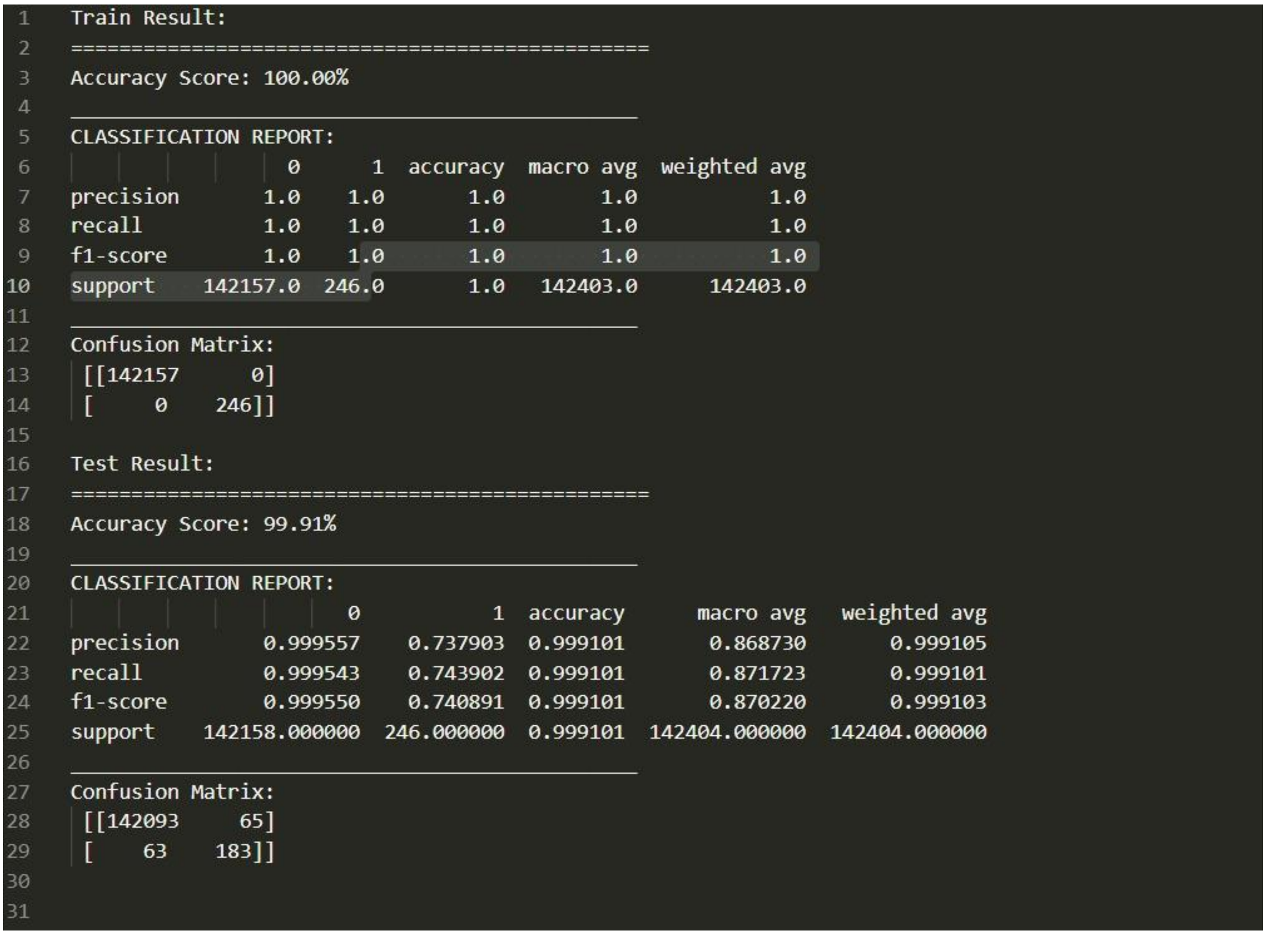


```
from sklearn.tree import DecisionTreeClassifier

tree_clf = DecisionTreeClassifier(random_state=42)
tree_clf.fit(X_train, y_train)

print_score(tree_clf, X_train, y_train, X_test, y_test, train=True)
print_score(tree_clf, X_train, y_train, X_test, y_test, train=False)
[18] ✓ 12.3s
```

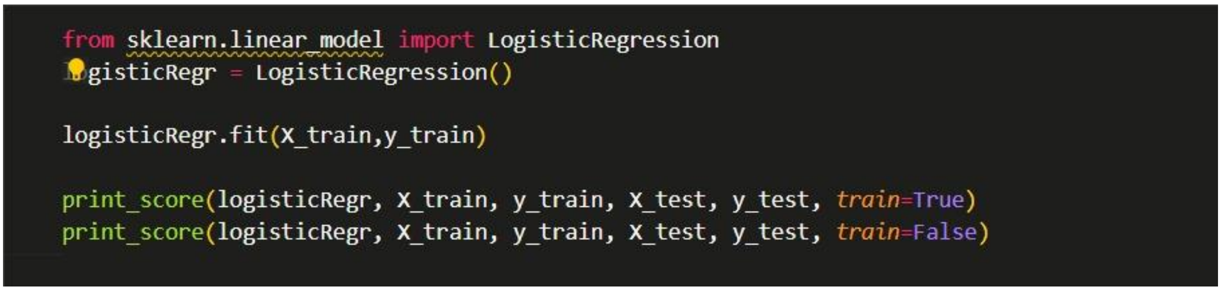
Output1:



```
Train Result:
================================================
Accuracy Score: 100.00%
_______________________________________________
CLASSIFICATION REPORT:
                  0      1  accuracy  macro avg  weighted avg
precision       1.0    1.0       1.0        1.0           1.0
recall          1.0    1.0       1.0        1.0           1.0
f1-score        1.0    1.0       1.0        1.0           1.0
support    142157.0  246.0       1.0   142403.0      142403.0
_______________________________________________
Confusion Matrix:
 [[142157      0]
 [     0    246]]

Test Result:
================================================
Accuracy Score: 99.91%
_______________________________________________
CLASSIFICATION REPORT:
                       0           1  accuracy      macro avg   weighted avg
precision       0.999557    0.737903  0.999101       0.868730       0.999105
recall          0.999543    0.743902  0.999101       0.871723       0.999101
f1-score        0.999550    0.740891  0.999101       0.870220       0.999103
support    142158.000000  246.000000  0.999101  142404.000000  142404.000000
_______________________________________________
Confusion Matrix:
 [[142093     65]
 [    63    183]]
```

Output 2 code:

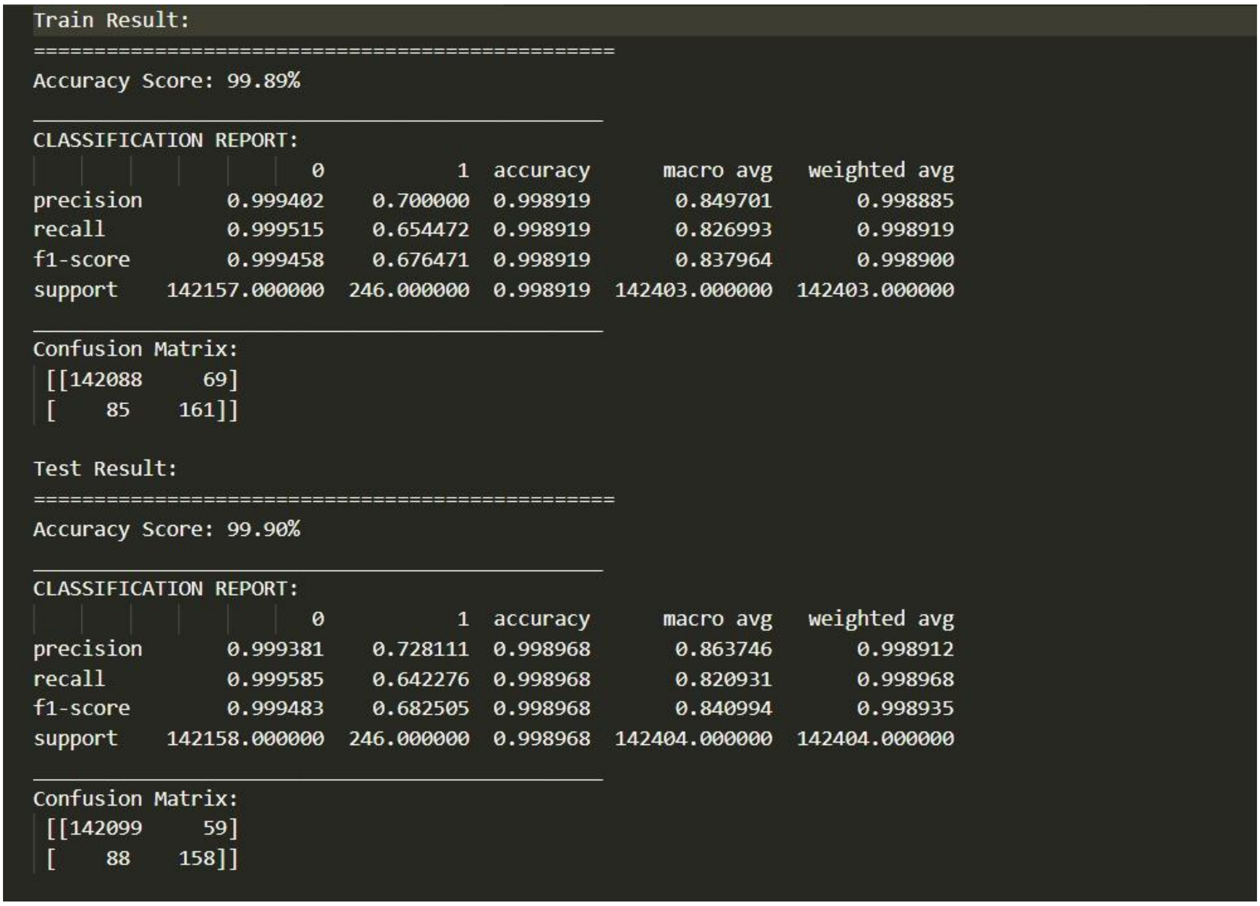


```
from sklearn.linear_model import LogisticRegression
gisticRegr = LogisticRegression()

logisticRegr.fit(X_train,y_train)

print_score(logisticRegr, X_train, y_train, X_test, y_test, train=True)
print_score(logisticRegr, X_train, y_train, X_test, y_test, train=False)
```

Output 2:



```
Train Result:
================================================
Accuracy Score: 99.89%
_______________________________________________
CLASSIFICATION REPORT:
                       0           1  accuracy      macro avg   weighted avg
precision       0.999402    0.700000  0.998919       0.849701       0.998885
recall          0.999515    0.654472  0.998919       0.826993       0.998919
f1-score        0.999458    0.676471  0.998919       0.837964       0.998900
support    142157.000000  246.000000  0.998919  142403.000000  142403.000000
_______________________________________________
Confusion Matrix:
 [[142088     69]
 [    85    161]]

Test Result:
================================================
Accuracy Score: 99.90%
_______________________________________________
CLASSIFICATION REPORT:
                       0           1  accuracy      macro avg   weighted avg
precision       0.999381    0.728111  0.998968       0.863746       0.998912
recall          0.999585    0.642276  0.998968       0.820931       0.998968
f1-score        0.999483    0.682505  0.998968       0.840994       0.998935
support    142158.000000  246.000000  0.998968  142404.000000  142404.000000
_______________________________________________
Confusion Matrix:
 [[142099     59]
 [    88    158]]
```

3.2 Technical components

Algorithms used:

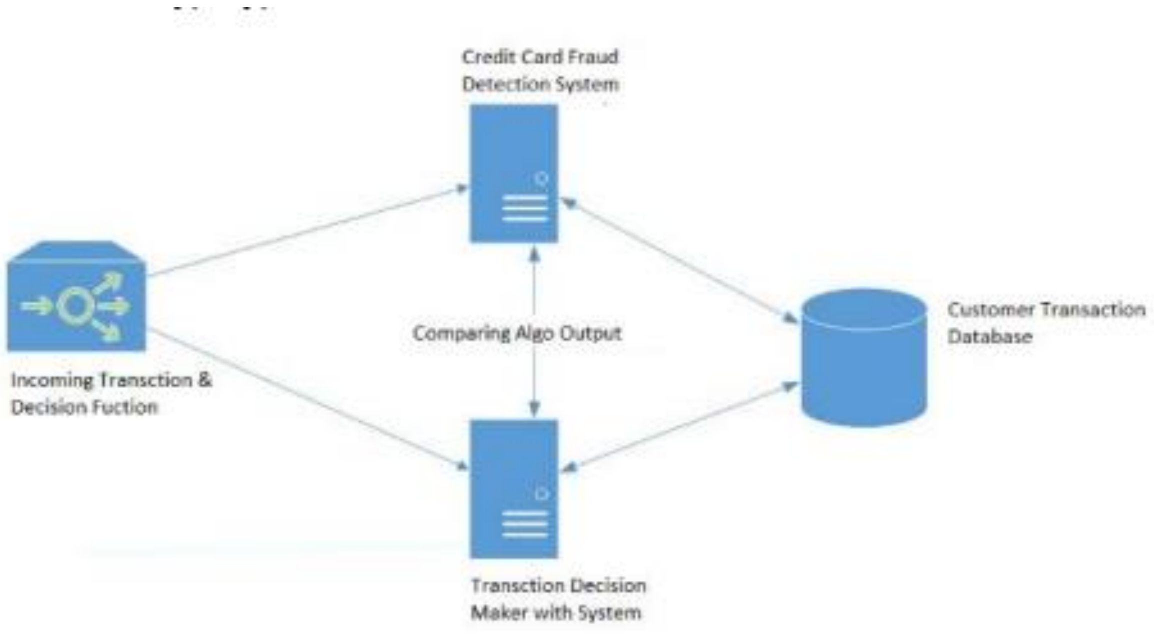
- Decision Trees (DTs): Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

- Extreme Gradient Boosting (XGBoost): The XGBoost (eXtreme Gradient Boosting) is a popular and efficient open-source implementation of the gradient boosted trees algorithm. Gradient boosting is a supervised learning algorithm that attempts to accurately predict a target variable by combining an ensemble of estimates from a set of simpler and weaker models.
- Light Gradient Boosting Machine(LightGBM): LightGBM, short for light gradient-boosting machine, is a free and open-source distributed gradient-boosting framework for machine learning, originally developed by Microsoft. It is based on decision tree algorithms and used for ranking, classification, and other machine learning tasks.
- CatBoost: CatBoost is an algorithm for gradient boosting on decision trees. Developed by Yandex researchers and engineers, it is the successor of the MatrixNet algorithm that is widely used within the company for ranking tasks, forecasting and making recommendations.
- Logistic regression: Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.

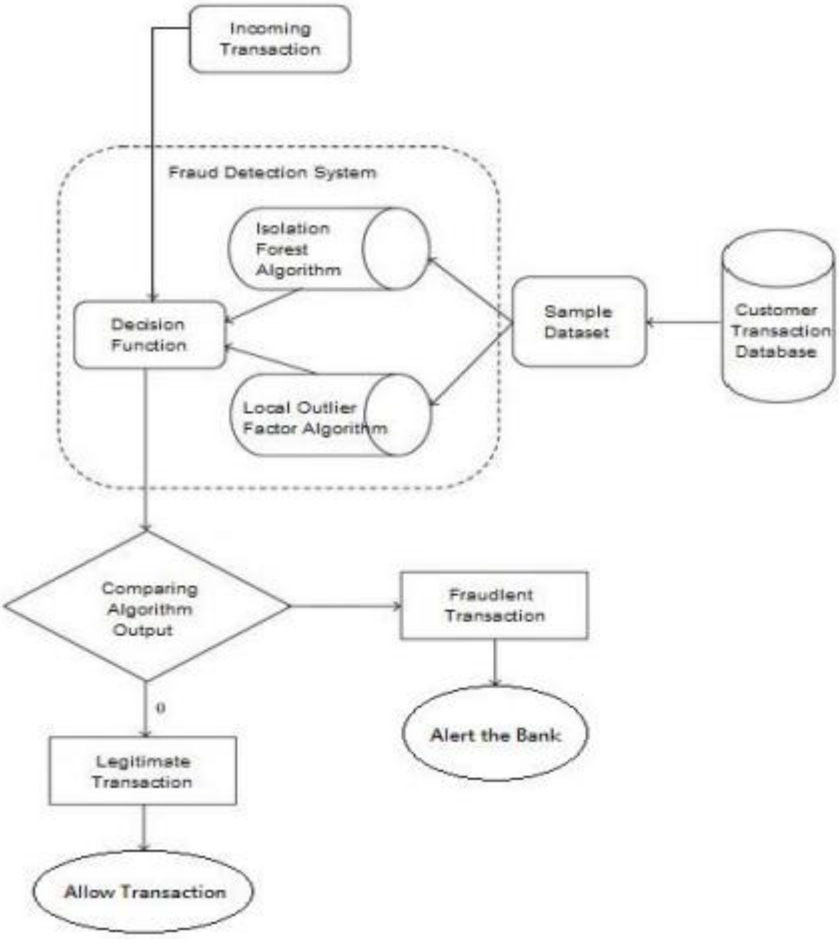
- Language used -Python
- Libraries used –
    - pandas - for data analysis and manipulation

- matplotlib - for visualizing data using graphs
- missingno - for visualizing missing data, completeness (or lack thereof) of the dataset
- plotly - for generating graphs and visualizing data
- seaborn - for making graphs generated by matplotlib more attractive and informative
- scikit-learn - for building machine learning models and generating confidence scores

3.3 System Architecture:



3.4 Data Flow in the System:



4. BUSINESS PLAN

4.1 Key Activities:

- Updating Information and transactions
- Notifying the user and bank about every Transaction
- Upgrading the model from time to time
- Development and enhancing the software platform on the basis of review

4.2 Key Resources:

- technology talent
- improving its algorithms and data analysis
- brand image

4.3 Key Partners:

- Payment processors (Banks)
- companies providing financial services
- investors
- other partners

4.4 Value Propositions:

For users - no fraud

For service providers - no hassle of rectifying

For authorities:Easy to detect people committing fraud

4.5 Cost structure:

- Maintenance cost
- Customer Acquisition Cost (CAC)
- Legal and settlement costs
- Infrastructure cost
- Customer support
- Optimization of algorithm

4.6 Revenue Streams:

- Commission based charges
- Selling user data (online spending behaviour)

4.7 Customer Segment:

- Payment processors
- Payment gateways
- End users

4.8 Customer Relationship:

- The financial service provider is the critical component
- The same is the case with users

4.9 Channels:

- Social Media
- Digital ads

5. FINANCE

5.1 Growth Strategy

- Pursuing most used payment processors and payment gateways to integrate our api.
- Targeting small and medium sized businesses for integrating our api in their payment portal to get access to markets outside of popular payment gateways.
- promoting the brand outside of the payment niche and marketing it as a brand of trust in financial markets.

5.2 Traction

- utilizing access to excess data to train our model and provide customers with more accurate predictions than any service in the market.
- lightweight service, does not slow the performance of the servers
- easy to integrate in any payment service
- brand of trust for end users

5.3 Financials

- Enlist and describe the various heads of income, expenditure
  - ➔ Expenditure
    - ◆ Customer Acquisition Cost (CAC)
    - ◆ Marketing Cost
    - ◆ Employee salaries Cost
    - ◆ Maintenance cost
    - ◆ Legal and settlement costs
    - ◆ Infrastructure cost

- Customer support
- Optimization of algorithm

➔ Income

- commission charges from our users for providing our api
- Selling user data (online spending behaviour)

- Draw a table of Financials for three years

	2022	20XX	20XX	
End Users	~60,000	200,000	1,100,000	
Jobs	25	120	450	
Average value / transaction	₹ 2,700	₹ 2,900	₹ 3,200	
Average price per api request	0.3% per transaction	0.4% per transaction	0.5% per transaction	
COMPANY REVENUE	₹ 4,86,000	₹ 23,20,000	₹ 1,76,00,000	
Gross Profit	₹ 4,86,000	₹ 23,20,000	₹ 1,76,00,000	
OPEX				
- Sales & Marketing	1,50,000	2,55,000	8,33,500	70%
- Customer Service	87,500	96,250	1,05,875	10%
- Product Development	5,62,500	8,90,625	14,20,156	5%
- Misc.	60,000	80,200	1,20,000	2%
TOTAL OPEX	8,60,000	13,22,075	24,79,531	
EBIT	**-3,74,000**	**9,97,925**	**1,51,20,469**	

6. CONCLUSION

Credit card fraud is without a doubt an act of criminal dishonesty. This project has worked on the most common methods of fraud along with their detection methods. The model has also shown, how machine learning can be applied to get better results in fraud detection. While using various models such as Decision Trees (DTs), Extreme Gradient Boosting (XGBoost), Light Gradient Boosting Machine (LightGBM), CatBoost, Logistic regression the algorithm does reach over 99.96% accuracy, in testing. This high percentage of accuracy is to be expected due to the huge imbalance between the number of valid and number of genuine transactions. Since the entire dataset consists of only two days' transaction records, it's only a fraction of data that can be made available if this project were to be used on a commercial scale. Being based on machine learning algorithms, the program will only increase its efficiency over time as more data is put into it.