Low Level Design (LLD)

Fraud Transaction Detection

Gautam Sharma

Document Version Control

Date Issued	Version	Description	Author
17th August 2020	V3	Deployes	Gautam Sharma

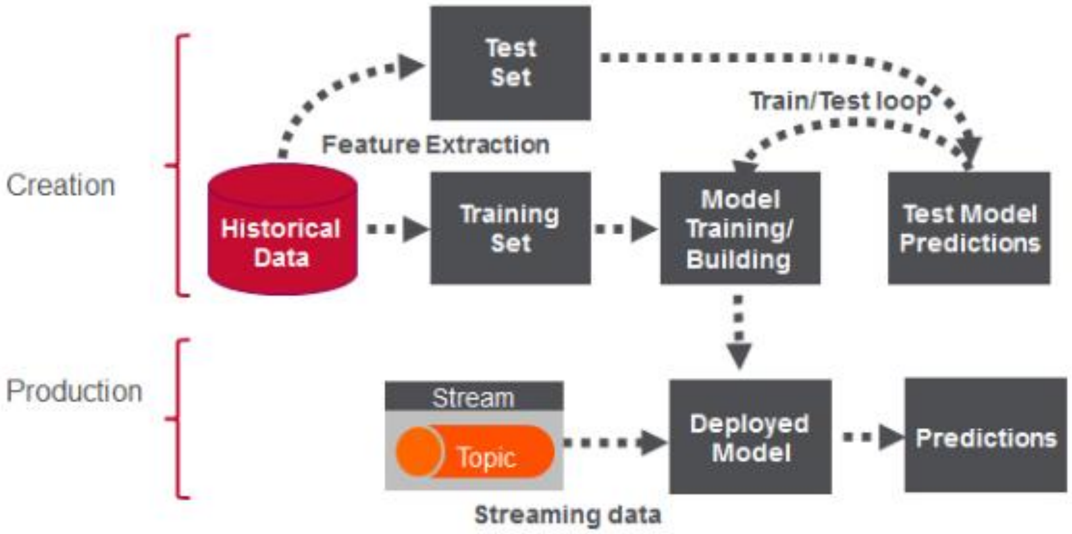
Contents

Abstract

Credit card frauds are easy and friendly targets. E-commerce and many other online sites have increased the online payment modes, increasing the risk for online frauds. Increase in fraud rates, researchers started using different machine learning methods to detect and analyse frauds in online transactions. The main aim of the paper is to design and develop a novel fraud detection method for Streaming Transaction Data, with an objective, to analyse the past transaction details of the customers and extract the behavioural patterns. Where cardholders are clustered into different groups based on their transaction amount.

1 Introduction

- Credit card generally refers to a card that is assigned to the customer (cardholder), usually allowing them to purchase goods and services within credit limit or withdraw cash in advance. Credit card provides the cardholder an advantage of the time, i.e., it provides time for their customers to repay later in a prescribed time, by carrying it to the next billing cycle.
- Credit card frauds are easy targets. Without any risks, a significant amount can be withdrawn without the owner's knowledge, in a short period. Fraudsters always try to make every fraudulent transaction legitimate, which makes fraud detection very challenging and difficult task to detect.
- With different frauds mostly credit card frauds, often in the news for the past few years, frauds are in the top of mind for most the world's population. Credit card dataset is highly imbalanced because there will be more legitimate transaction when compared with a fraudulent one.



1.1 Scope

In this we developed a novel method for fraud detection, where customers are grouped based on their transactions and extract behavioural patterns to develop a profile for every cardholder. Then different classifiers are applied on three different groups later rating scores are generated for every type of classifier. This dynamic changes in parameters lead the system to adapt to new cardholder's transaction behaviours timely.

1.2 Constraints

We will only be selecting a dataset from kaggle.

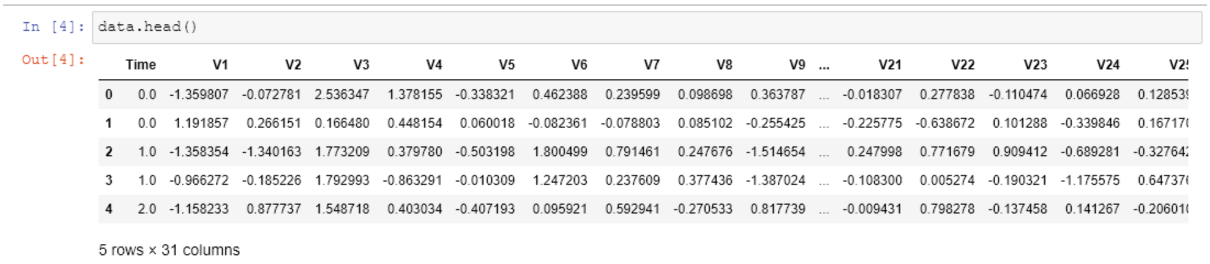
2 Technical specifications

2.1 Dataset

Sr. No.	**Featutre**	Description
1.	Time	Time in seconds to specify the elapses between the current transaction and first transaction.
2.	Amount	Transaction amount
3.	Class	0 - not fraud 1 – fraud

2.1.1 Diabetes dataset overview

The table Consists of data Time And amount and as shown in below fig.

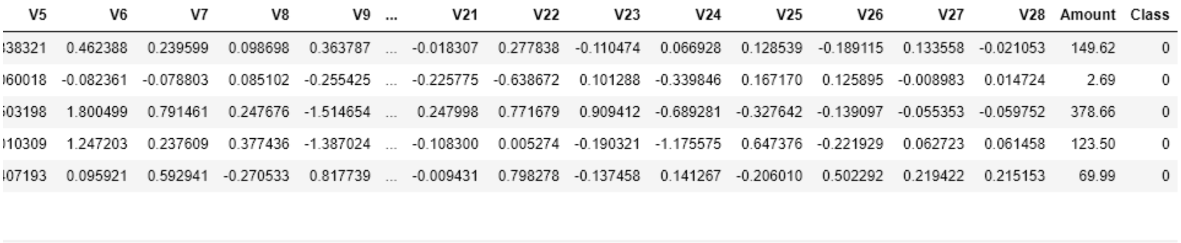


```
In [4]: data.head()
```

Out[4]:

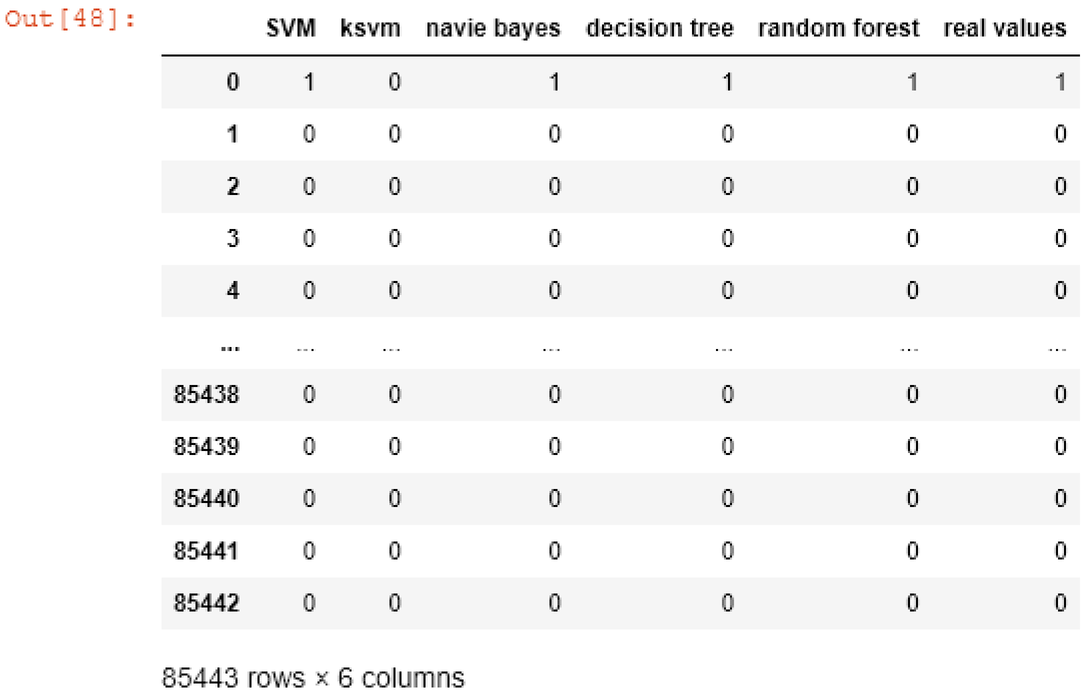
	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V2
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	...	-0.018307	0.277838	-0.110474	0.066928	0.12853
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	...	-0.225775	-0.638672	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

5 rows × 31 columns



V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
38321	0.462388	0.239599	0.098698	0.363787	...	-0.018307	0.277838	-0.110474	0.066928	0.128539	-0.189115	0.133558	-0.021053	149.62	0
60018	-0.082361	-0.078803	0.085102	-0.255425	...	-0.225775	-0.638672	0.101288	-0.339846	0.167170	0.125895	-0.008983	0.014724	2.69	0
03198	1.800499	0.791461	0.247676	-1.514654	...	0.247998	0.771679	0.909412	-0.689281	-0.327642	-0.139097	-0.055353	-0.059752	378.66	0
10309	1.247203	0.237609	0.377436	-1.387024	...	-0.108300	0.005274	-0.190321	-1.175575	0.647376	-0.221929	0.062723	0.061458	123.50	0
07193	0.095921	0.592941	-0.270533	0.817739	...	-0.009431	0.798278	-0.137458	0.141267	-0.206010	0.502292	0.219422	0.215153	69.99	0

2.1.2 Experimental Results

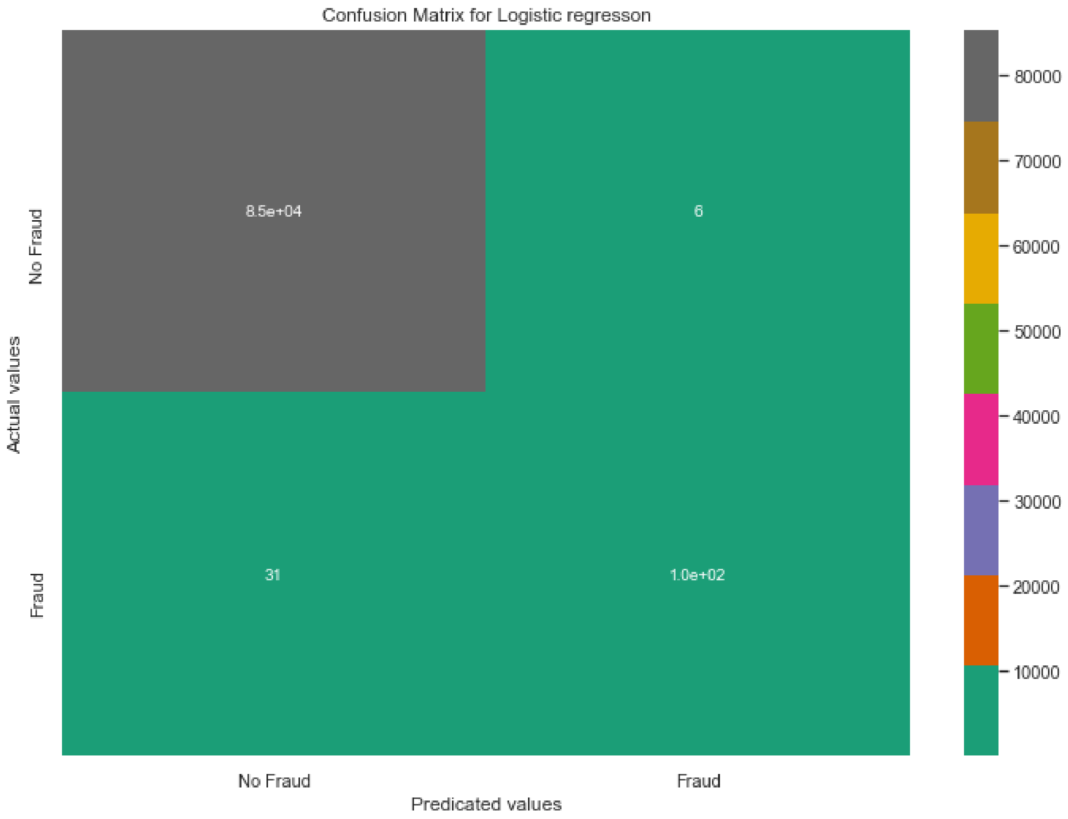


Out[48]:

	SVM	ksvm	navie bayes	decision tree	random forest	real values
0	1	0	1	1	1	1
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
...	...	...	...	...	...	...
85438	0	0	0	0	0	0
85439	0	0	0	0	0	0
85440	0	0	0	0	0	0
85441	0	0	0	0	0	0
85442	0	0	0	0	0	0

85443 rows × 6 columns

2.2 Predicting Frauds or Not



2.3 Database

System needs to store every request into the database and we need to store it in such a way that it is easy to retrain the model as well.

1. The User chooses the time and Amount
2. The User gives required information.

2.4 Deployment

1. Heroku

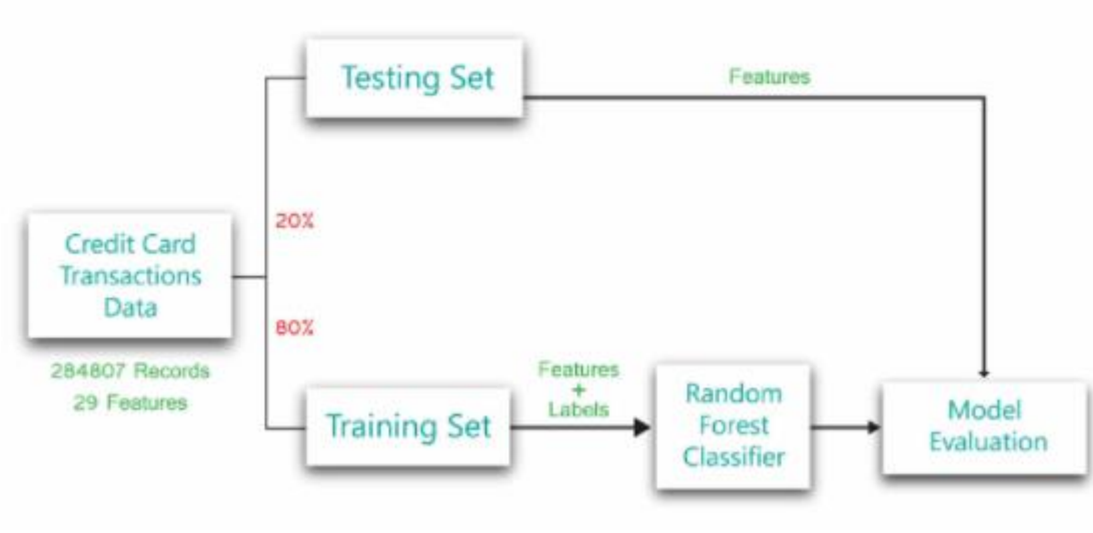
3 Technology stack

Front End	HTML/CSS/JS/React
Backend	Python Flask
Database	Kaggle
Deployment	Heroku

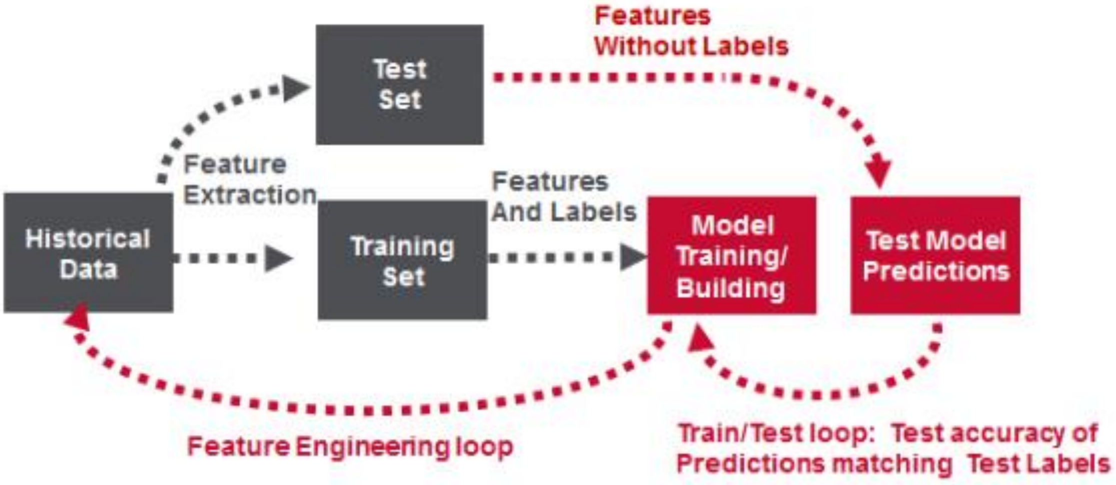
4 Proposed Solution

In this we developed a novel method for fraud detection, where customers are grouped based on their transactions and extract behavioural patterns to develop a profile for every cardholder. Then different classifiers are applied on three different groups later rating scores are generated for every type of classifier. This dynamic changes in parameters lead the system to adapt to new cardholder's transaction behaviours timely. Followed by a feedback mechanism to solve the problem of concept drift. We observed that the Matthews Correlation Coefficient was the better parameter to deal with imbalance dataset. MCC was not the only solution. By applying the SMOTE, we tried balancing the dataset, where we found that the classifiers were performing better than before. The other way of handling imbalance dataset is to use one-class classifiers like one-class SVM. We finally observed that Logistic regression, decision tree and random forest are the algorithms that gave better results.

5 Model training/validation workflow



6 User I/O workflow



7 Exceptional scenarios

Step	Exception	Mitigation	Module
17th August 2021	1.1	Deploy	Gautam Sharma

8 Test cases

Test case	Steps to perform test case	Module	Pass/Fail

9 Key performance indicators (KPI)

- Time and workload reduction using the EHR model.
- Comparison of accuracy of diff, model prediction. Like SVM ,