

High Level Design(HLD)

Credit Card Fraud Transaction Detection

Version : 1.0

Document Version Control

Date Issued	Version	Description	Author
9/14/2021	1	Initial HLD: V-1.0	Sandeep Kashyap

Contents

- 1. Introduction.....5
 - 1.1. Purpose of the Document..... 5
 - 1.2. Objective of HLD..... 5
 - 1.3. Scope of HLD.....5
- 2. System Overview.....6
 - 2.1. Product Perspective.....6
 - 2.2. Problem Statement.....6
 - 2.3. Proposed Solution.....6
 - 2.4. Technical Requirements.....6
 - 2.5. Data Requirements7
 - 2.6. Tools used.....8
 - 2.7. Constraints.....9
 - 2.8. Assumptions.....9
- 3. Design Flow.....10
 - 3.1. Process Flow.....10
 - 3.1.1. Model Training and Evaluation.....10
 - 3.1.2. Deployment Process.....11
 - 3.2. Event Log.....11
- 4. Performance.....12
 - 4.1. Re-usability.....12
 - 4.2. Application Compatibility.....12
 - 4.3. Resource Utilization.....13
 - 4.4. Deployment.....13
 - 4.5. User Interface.....14
 - 4.6. Conclusion.....15

Abstract

In today's world, we are on the express train to a cashless society. According to the World Payments Report, in 2016 total non-cash transactions increased by 10.1% from 2015 for a total of 4826 billion transactions! That's huge ! Also, it's expected that in future years there will be a steady growth of non-cash transactions. Now, while this might be exciting news, on the flip-side fraudulent transactions are on the rise as well. Even with EMV smart chips being implemented, we still have a very high amount of money lost from credit card fraud. This is now becoming a serious problem since most of the time, a person who has become a victim of this fraud doesn't have any idea about what has happened until the very end. So in this project, what we have tried is to create a Web App for the detection of such types of frauds with the help of Machine Learning

1. Introduction

1.1 Purpose of the Document

The purpose of this plan is to

- Identify different design approaches.
- Identify core modules/subsystems of the system and subsystem.
- Identify the best suitable technology for various sub-systems boundary.
- Identify areas that need R&D.
- Identify third party components required in the system.
- Describes the performance requirements
- Identify various usage scenarios.

1.1 Objective of HLD

1. To provide an overview of the entire system.
2. To provide a module-wise breakup of the entire system.
3. To provide introduction and high level working of every module involved.

1.3 Scope of HLD

This HLD covers all areas of the system..

2. System Overview

2.1 Product Perspective

The Credit Card Fraud Detection using classification-based Machine Learning Algorithms.

2.2 Problem statement

Fraud detection is a set of activities undertaken to prevent money or property from being obtained through false pretenses. Fraud detection is applied to many industries such as banking or insurance. In banking, fraud may include forging checks or using stolen credit cards. Other forms of fraud may involve exaggerating losses or causing an accident with the sole intent for the pay-out. With an unlimited and rising number of ways someone can commit fraud, detection can be difficult to accomplish. Fraud detection is a critical issue for retailers determined to prevent losses and preserve customer trust.

2.3 Proposed Solution

The solution here is a Classification based Machine Learning model. It can be implemented by different classification algorithms (like Logistic Regression, Random Forest, Decision Tree, XGBoost and so on. After testing some classification based algorithms we found that Logistic Regression fits best for the data. Here first we are performing a Data preprocessing step, in which Data Profiling, feature engineering, feature selection, feature scaling, then we are going to build our model.

2.4 Technical Requirements

This document addresses the requirements for detecting the anomalies in the transaction for a given credit card details. For that, in this project we are going to use different technologies. Here are some requirements for this project.

- Model should be exposed through API or User Interface, so that anyone can test model
- Model should be deployed on cloud (Azure, AWS, GCP).

- Cassandra Database should be integrated in this project for any kind of user input.

2.5 Data Requirements

Data Requirement completely depends on our problem.

- For training and testing the model, we are using Credit Card Fraud dataset from Kaggle.
- From the user we are taking the following input.

Time

V1

V2

V3

V4

V5

V6

V7

V8

V9

V10

V11

V12

V14

V16

V17

V18

V21

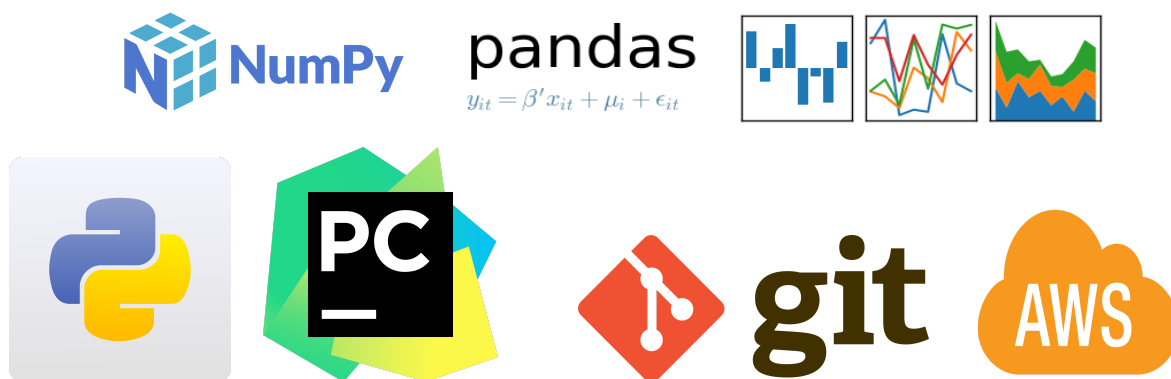
V27

V28

Amount

2.6 Tools Used

- PyCharm is used as an IDE.
- For visualization of the plots, Matplotlib, Seaborn are used.
- Aws is used for deployment of the model.
- Cassandra is used to retrieve, insert, delete, and update the database.
- Front end development is done using HTML/CSS.
- Flask is used for backend development and for API development.
- GitHub is used as a version control system.



2.7 Constraints

The Credit Card Fraud Detection System must be user friendly, errors free and users should not be required to know any of the back end working.

2.8 Assumptions

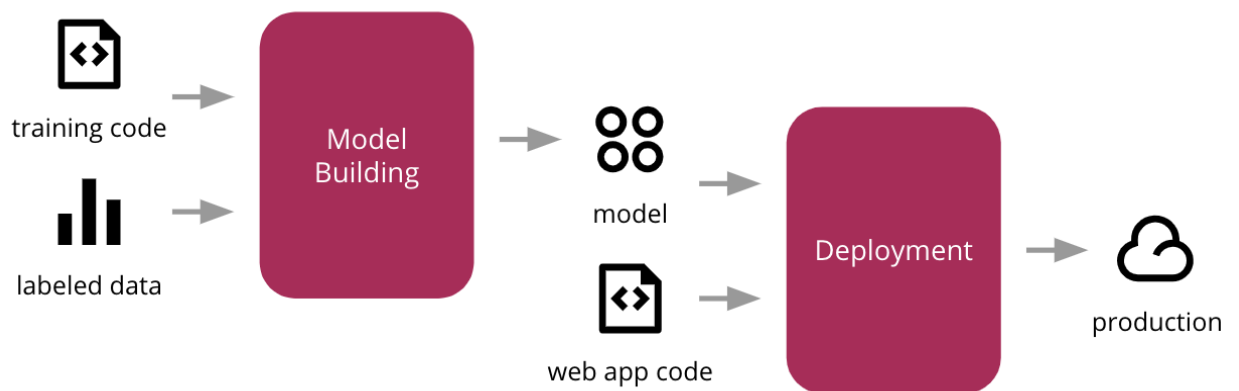
The main objective of the project is to implement the use cases as previously mentioned(2.2 Problem Statement) for new dataset that comes through any credit card transaction. Machine Learning based model is used for detecting anomaly in such use cases based on the input data. It is also assumed that all aspects of this project have the ability to work together in the way the designer is expecting.

3 Design Flow

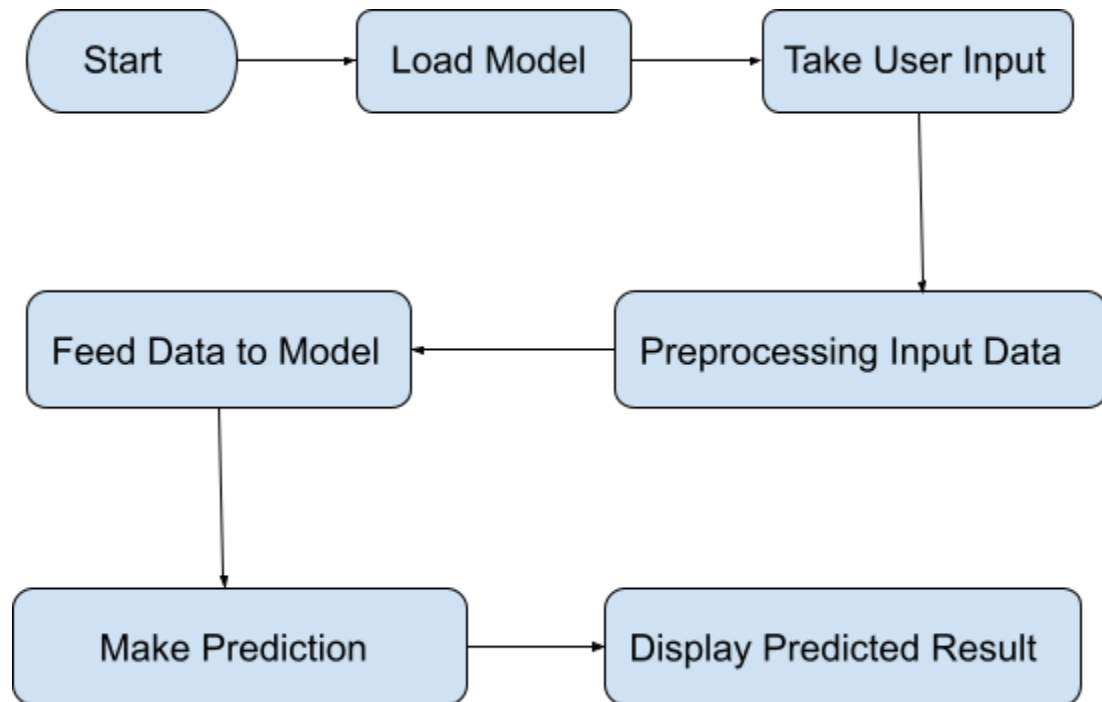
3.1 Process Flow

For identifying the different types of anomalies, we will use a machine learning based model. Below is the process flow diagram.

3.1.1 Model Training and Evaluation



3.1.2 Deployment Process



3.2 Event Log

In this Project we are logging every process so that the user will know what process is running internally.

Step-By-Step Description:

- In this Project we defined logging for every function, class.
- By logging we can monitor every insertion, every flow of data in the database.
- By logging we are monitoring every step which may create problems or every step which is important in the file system.
- We have designed logging in such a way that the system should not hang even after so much logging, so that we can easily debug issues which may arise during process flow.

4 Performance

The Credit Card Fraud Detection Model is used to predict whether a transaction done using a credit card is fraudulent or not. So, it should be as accurate as possible so that it should give as many accurate predictions as possible.

That's why building this model I followed complete process of Machine Learning. Here is a brief summary of the complete process.

1. First, I cleaned our dataset properly by removing all null values and duplicate values present in the dataset.
2. Then I performed Data profiling where I checked no of categorical features and numerical features.
3. After performing some statistical analysis I found out that the dataset is highly unbalanced.
4. I performed under Sampling which made a similar distribution for normal transactions and fraudulent transactions.
5. After some more preprocessing of the dataset I trained my model using some classification machine learning algorithm and found out that Logistic regression suited best for model building with an accuracy of nearly 90%.
6. Since the model was working fine so I performed hyperparameter tuning was not necessary.
7. After that I saved my model in pickle file format for model deployment.
8. Finally I deployed my model on various cloud platforms such as heroku, AWS.

4.1 Re-usability

I have done programming of this project in such a way that it should be reusable. So that anyone can add and contribute without facing any problems.

4.2 Application Compatibility

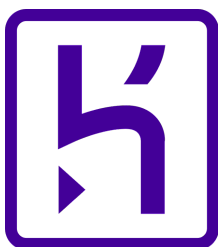
The different module of this project is using Python as an interface between them. Each module has its own job to perform and it is the job of the Python to ensure the proper transfer of information.

4.3 Resource Utilization

In this project, when any task is performed, it is likely that the task will use all the processing power available in that particular system until its job is finished. By keeping this in mind, In this project I have used the concept of multithreading.

4.4 Deployment

I have deployed this on Heroku and AWS



HEROKU

4.5 User Interface

Home page:

Credit Card Fraud Detection

Please enter the given feature values only in correct order

Time

V1

V2

V3

V4

V5

V6

V7

V8

V9

V10

V11

V12

V14

V16

V17

V18

V21

V27

V28

Amount

Submit

Credit Card Fraud Detection

Please enter the given feature values only in correct order

9.0

0.338262

1.119593

1.044367

0.222187

0.499361

0.246761

0.651583

0.069539

0.736727

0.366846

1.017614

0.836390

0.443523

0.739453

0.540980

0.476677

-0.246914

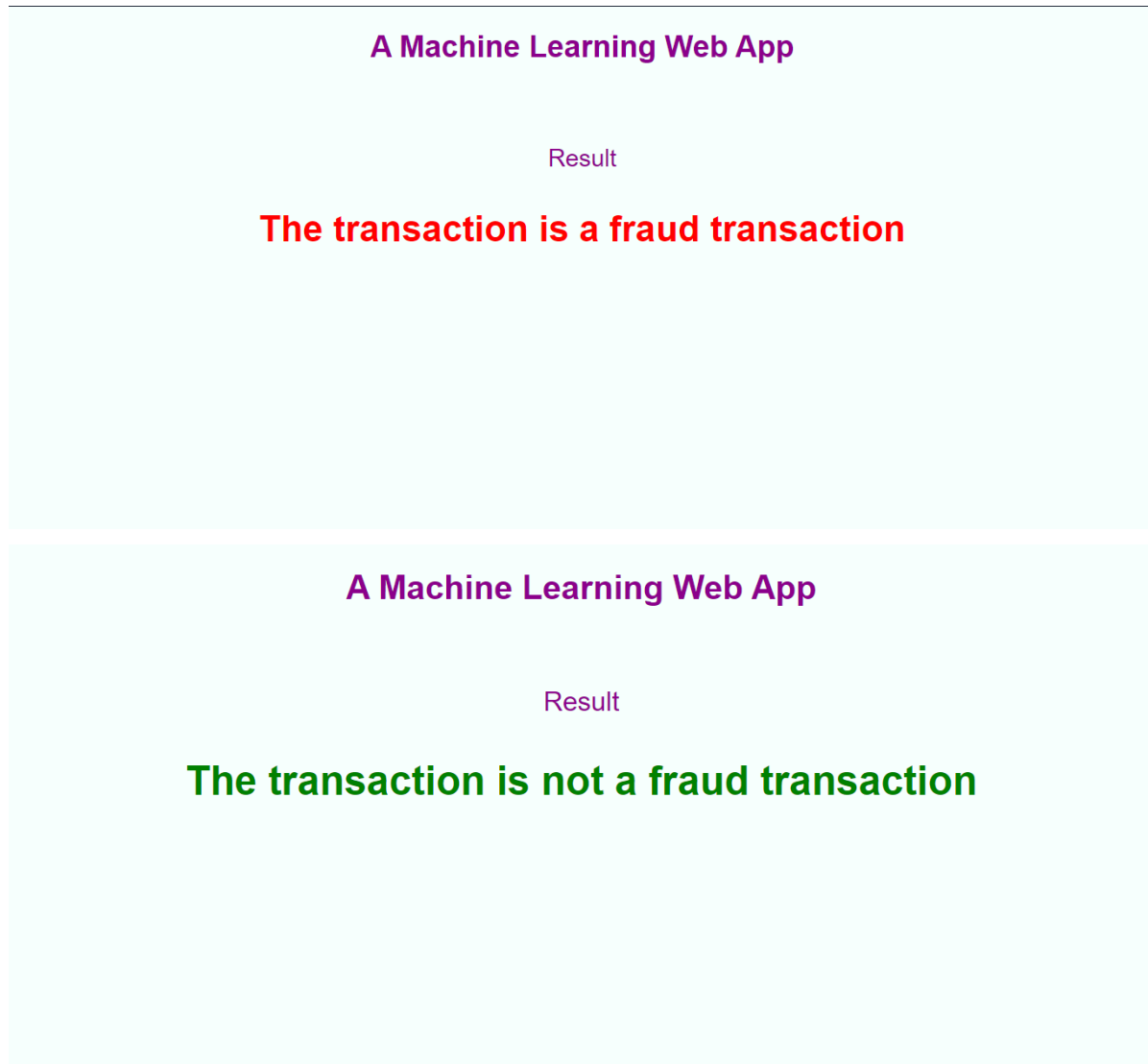
0.246219

0.083076

3.68

Submit

Prediction Page:



Conclusion:

Credit Card Fraud Detection Web app will predict whether a given transaction is a fraud transaction or not based on its given details. This will enhance the security and save customers from various kinds of fraud done through credit card.