**High Level Design (HLD)**

*Credit Card Fraud Transaction Detection*

*Created By-:*

*Vishvajeet Singh*

*Akanshu Rana*

# 

# 

# 

# Document Version Control

| **Date Issued** | **Version** | **Description** | **Author** |
| --- | --- | --- | --- |
| 26/08/2022 | 1 | Initial HLD — V1.0 | Vishvajeet Singh  Akanshu Rana |

## **Contents**

[Document Version Control. 2](#_heading=h.gjdgxs)

[Abstract. 4](#_heading=h.30j0zll)

1. [Introduction 5](#_heading=h.1fob9te)
   1. [Why this High-Level Design Document?. 5](#_heading=h.3znysh7)
   2. [Scope. 5](#_heading=h.2et92p0)
   3. [Definitions 5](#_heading=h.tyjcwt)
2. [General Description. 6](#_heading=h.1t3h5sf)
   1. [Product Perspective 6](#_heading=h.4d34og8)
   2. [Problem statement 6](#_heading=h.2s8eyo1)
   3. [PROPOSED SOLUTION 6](#_heading=h.17dp8vu)
   4. [FURTHER IMPROVEMENTS 6](#_heading=h.4i7ojhp)
   5. [Technical Requirements. 6](#_heading=h.3rdcrjn)
   6. [Data Requirements 7](#_heading=h.26in1rg)
   7. [Constraints](#_heading=h.2xcytpi) 8
   8. [Assumptions.](#_heading=h.35nkun2) 8
3. [Design Details](#_heading=h.1ksv4uv) 9
   1. [Process Flow.](#_heading=h.44sinio) 9
      1. [Model Training and Evaluation](#_heading=h.2jxsxqh) 9
      2. [Deployment Process](#_heading=h.z337ya) 10
   2. [Event log](#_heading=h.3j2qqm3)
   3. [Error Handling 1](#_heading=h.1y810tw)0
   4. Performance 11
   5. Reusability. 12
   6. Application Compatibility 12
   7. Resource Utilization 12
   8. Deployment. 12
4. Conclusion 12

**Abstract**

We are living in an internet era in this 21st century. Nowadays online transactions have become a prominent mode of payment and credit card is one the most popular methods. But there is an apprehension be Victim of fraud as well. Recently we have seen so many fraudulent transactions through credit cards.

According to the dataset we have found 492 fraud Transactions out of 284,807 transactions in European Countries within two days. It is important that credit card companies must recognize fraudulent credit card transactions so that customers could not be charged for those items that they had not purchased.

To resolve this grievous problem we have tried to create a **Web Application** for the detection these types of frauds with the help of machine learning algorithm.

1. **Introduction**

###### Why this High-Level Design Document?

The purpose of this High-Level Design (HLD) Document is to add the necessary detail to the current project description to represent a suitable model for coding.

This document is also intended to help detect contradictions prior to coding, and can be used as a reference manual for how the modules interact at a high level.

The HLD will:

* + - Present all of the design aspects and define them in detail
    - Describe the user interface being implemented
    - Describe the hardware and software interfaces
    - Describe the performance requirements
    - Include design features and the architecture of the project

##### **Scope**

The HLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology architecture.

The HLD uses non-technical to mildly-technical terms which should be understandable to the administrators of the system.

* 1. **Definitions**



***Term*** 

*Database*

*IDE*

*Heroku*

***Description***

Collection of all the information monitored by this system

Integrated-Development Environment

Heroku Web Services

### General Description

#### Product Perspective

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

* 1. 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.

.

* 1. **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, XG Boost and so on.

After testing some classification based algorithms we found that Random Forest 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.

#### Technical Requirements

This document addresses the requirements for detecting the anomalies in the transaction for a given credit card details. We are using different technologies to detect fraudulent transactions.

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, Heroku).
* Cassandra Database should be integrated in this project for any kind of user input

#### Data Requirements

Data requirement completely depends on our problem statement.

* We loaded the data and did preprocessing, featuring engineering, modeling and testing the data.
* 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**

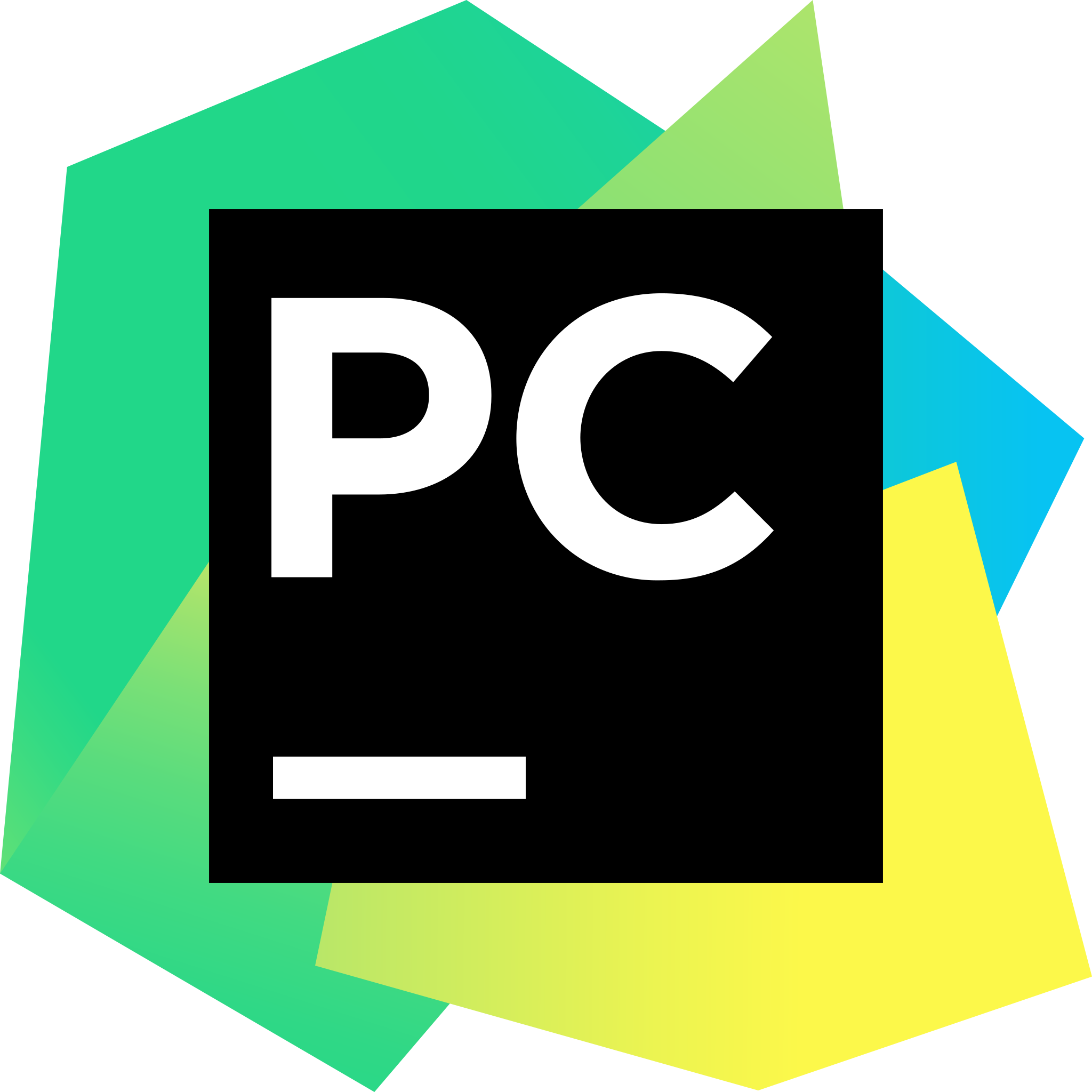
**- V1 to V27**

**- Amount**

* 1. **Tools used**

Python programming language and frameworks such as NumPy, Pandas, seaborn, Scikit-learn, are used to build the whole model.

* PyCharm is used as an IDE.
* Heroku 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 models are used for detecting anomalies in such use cases based on the input data.

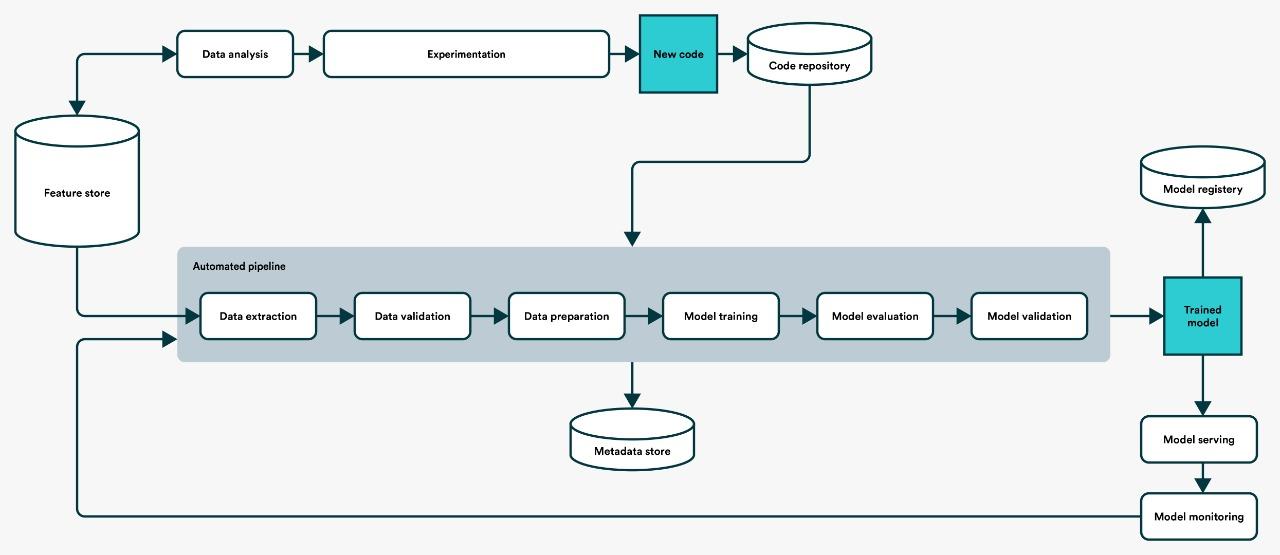
It is also assumed that all aspects of this project have the ability to work together in the way the designer is expecting.

## Design Details

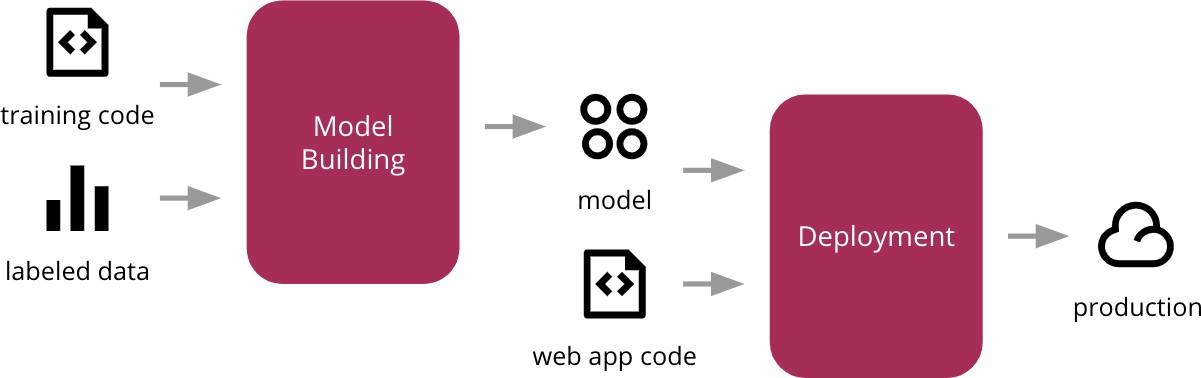
##### Process Flow

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

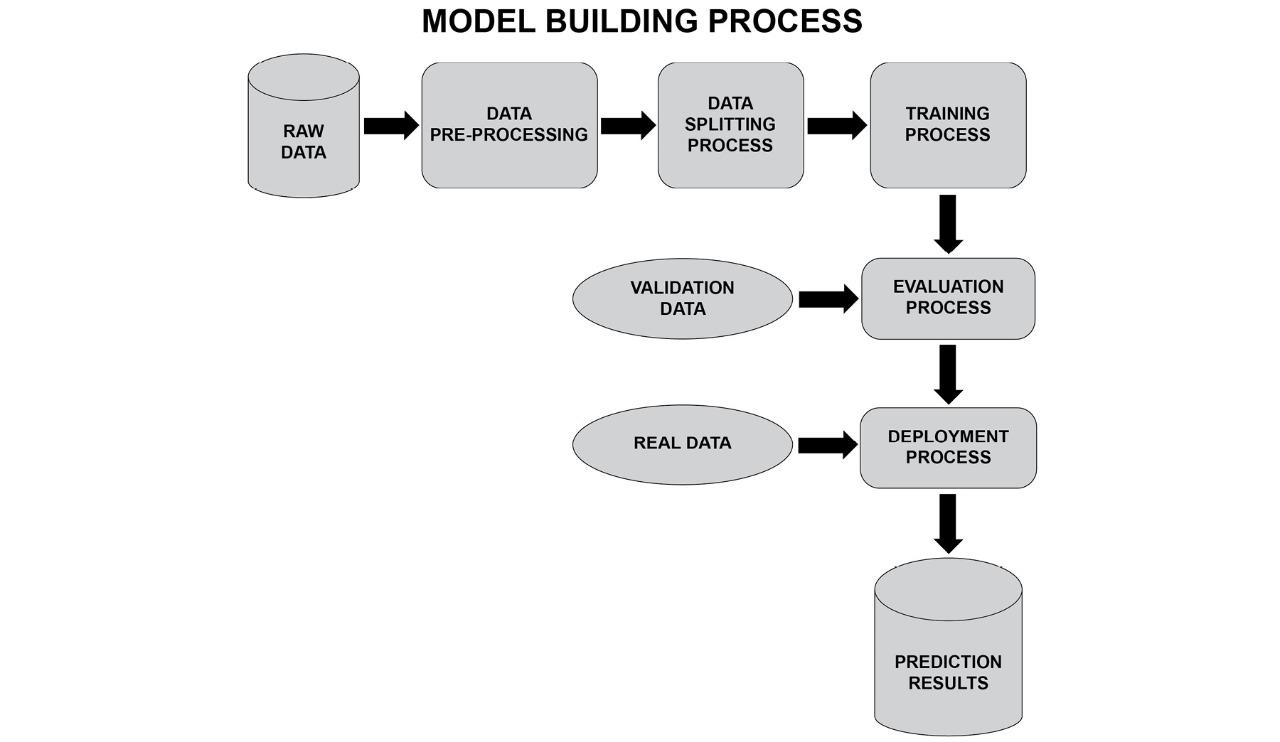
##### Proposed methodology



##### Model Training and Evaluation



##### Deployment Process



* 1. Event log

The system should log every event so that the user will know what process is running internally.

**Initial Step-By-Step Description:**

1. The System identifies at what step logging required
2. The System should be able to log each and every system flow.
3. Developers can choose logging methods. You can choose database logging/ File logging as well.
4. System should not hang even after using so many loggings. Logging just because we can easily debug issues so logging is mandatory to do.
   1. **Error Handling**

Should errors be encountered, an explanation will be displayed as to what went wrong? An error will be defined as anything that falls outside the normal and intended usage.

3.4 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 the 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 Random forest suited best for model building with an accuracy of nearly 93%.
6. I performed hyperparameter tuning then the accuracy increased and it became 96%.
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.

#### 3.5 Reusability

The code written and the components used should have the ability to be reused with no problems.

#### 3.6 Application Compatibility

The different components for this project will be using Python as an interface between them. Each component will have its own task to perform, and it is the job of the Python to ensure proper transfer of information.

#### 3.7 Resource Utilization

When any task is performed, it will likely use all the processing power available until that function is finished.

**3.8 Deployment**

## Conclusion

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