Architecture
Credit Card Fraud Detection

Written By	Sandeep Kashyap
Document Version	1.1
Last Revised Date	9/27/2021

Document Control

Change Record

Version	Date	Author	Comments
1.0	9/19/2021	Sandeep Kashyap	Introduction And Architecture Defined
1.1	9/27/2021	Sandeep Kashyap	Added used based architecture

Approval Status:

Version	Review Date	Reviewed By	Approved By	Comments
1.0				

Contents :

Document Control **1**

1.Introduction **3**

1.1 Why this Architecture Design Document? 3

2.User Based Architecture **4**

3.Architecture Description **5**

3.1 Data Description 6

3.2 Data Cleaning 6

3.3 Exploratory Data Analysis 6

3.4 Event Log 7

3.5 Data Insertion into Database 7

3.6 Export Data from Database 7

3.7 Data Preprocessing 7

3.8 Model Creation/ Model Building 8

3.9 Model Dump 8

3.10 Data from User 8

3.11 Data Validation 8

3.12 Model Call for specific input 8

3.13 User Interface 9

3.14 Deployment 9

1.Introduction

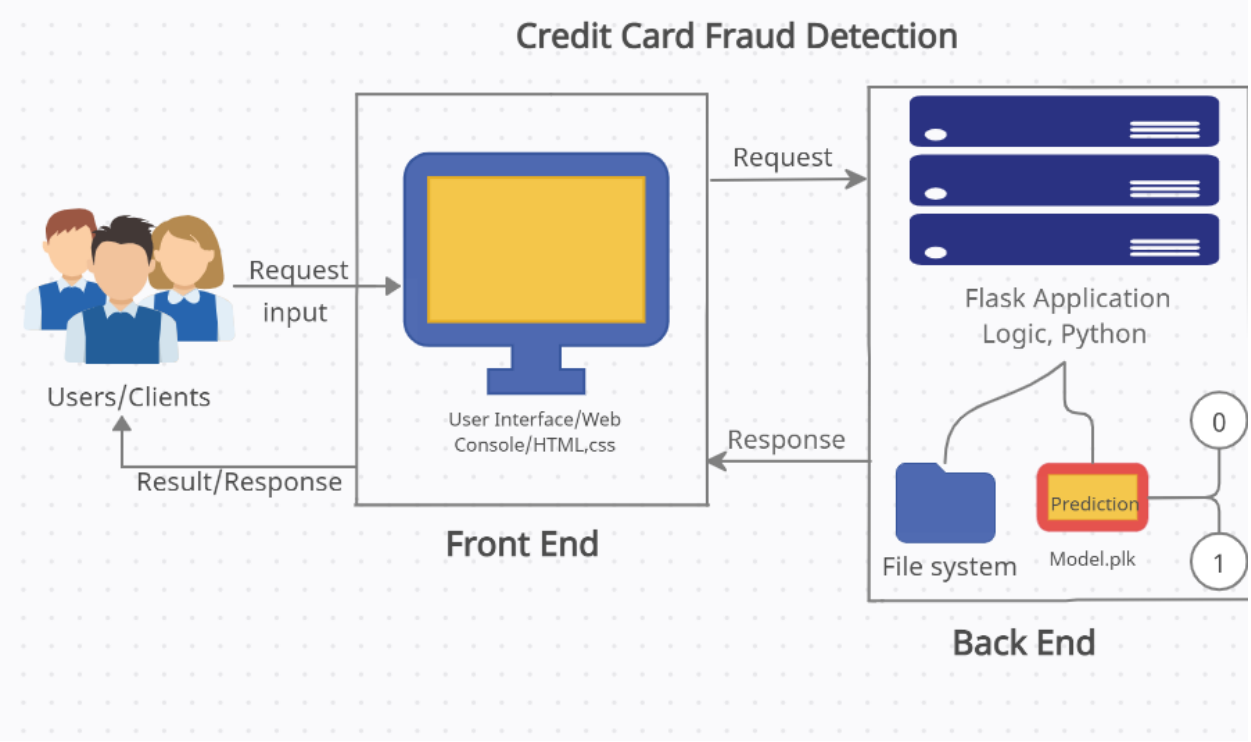
1.1 Why this Architecture Design Document?

The purpose of this document is to provide a detailed architecture design of the Back Order Prediction Project by focusing of four key quality attributes:

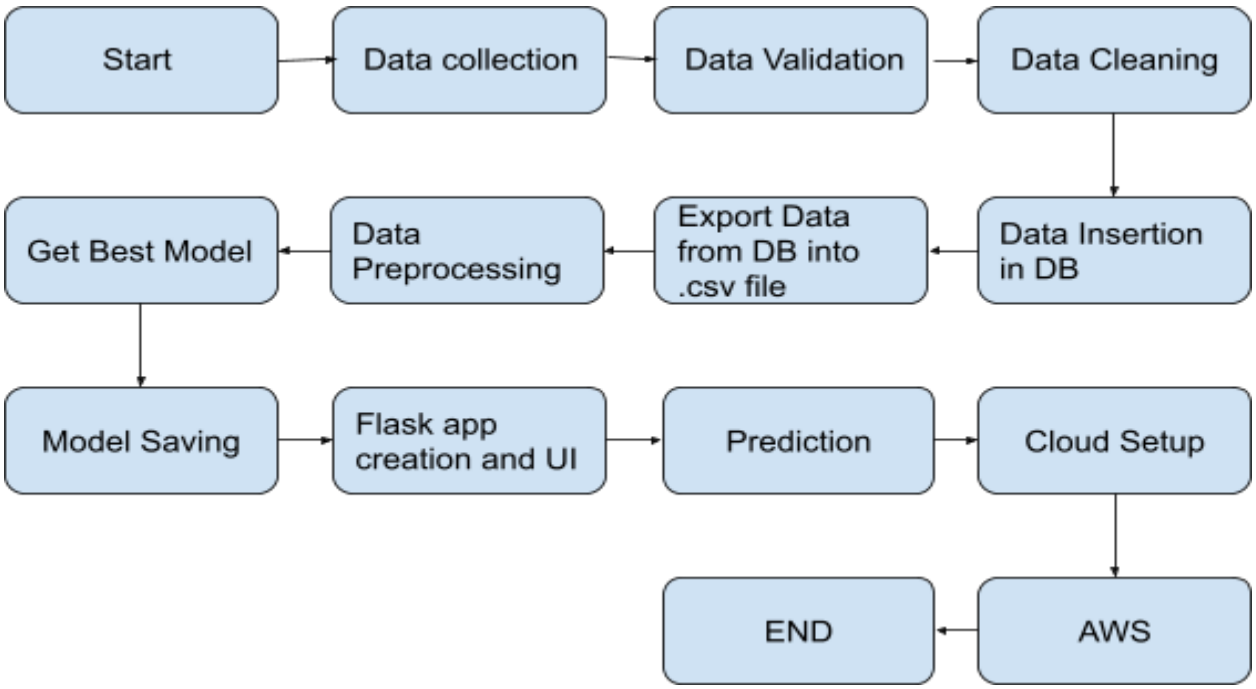
Usability, availability, maintainability, testability.

This document will address the background for this project, and the architecturally significant function requirements.The intention of this document is to help the development team to determine how the system will be structured at the highest level. Finally, the project coach can use this document to validate that the development team is meeting the agreed-upon requirements during the evaluation of the team's efforts.

2. User Based Architecture



3.Architecture Description



Architecture

3.1 Data Description

This data is about fraud detection in credit card transactions. The data was made by credit cards in September 2013 by European cardholders. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions.

It contains only numeric input variables which are the result of a PCA transformation. The original features and more background information about the data is not provided.

The dataset contains 284,807 instances, 492 instances are fraudulent, the remaining 284,315 instances are genuine.

3.2 Data Cleaning

In the Cleaning process, I cleaned up all the data having null values because the percentage of null values in the dataset was very less. So I have dropped all the rows that were containing null values.

3.3 Exploratory Data Analysis

I have done EDA in such a way that every nook and corner of features were clearly justified with the help of correlation, plotting the heat map using seaborn and matplotlib and so on,and found out that the data set is quite good but is highly unbalanced

3.4 Event Log

The system should log every event so that the user will know what process is running internally. Logging is implemented using python's standard logging library.Initial step by step description:-

- The system should be able to log each and every system flow.
- System must be able to handle logging at greater scale because it helps debugging the issue and hence it is mandatory too.

3.5 Data Insertion into Database

- Database Creation and connection - Create a database with name passed. If the database is already created, open the connection to the database.
- Table creation in the database.
- Insertion of files in the table

3.6 Export Data from Database

Data Export from Database - The data in a stored database is exported as a CSV file to be used for Data Pre-Processing and Model Training.

3.7 Data Preprocessing

Data pre-processing steps we could use are Null Value handling. Categorical to Numerical Transformation of columns, Undersampling the unbalanced data , splitting the data into train and test sets. Handling columns with standard deviation zero or below a threshold, etc.

3.8 Model Creation/ Model Building

After cleaning and processing the data, completing feature engineering I have done train test splitting using method build in preprocessing file and implemented various Classification Algorithm and found out that Logistic Regression suits best for the model with an excellent accuracy.

3.9 Model Dump

After comparing all accuracies and finding the best model for the dataset I have created a model and dumped the model in a pickle file format with the help of pickle module.

3.10 Data from User

Here The user will have to enter all the features values in correct order and have to submit it to the model with the help of UI interface.The data will be fed to the model which will predict whether the feature set represents a fraudulent transaction or not.

3.11 Data Validation

Here Data Validation will be done, given by the user.

3.12 Model Call for specific input

Based on the User Input will be thrown to the backend in the variable format then it will be converted into a numpy array which will be fed to our model. The loading of the pickle file will be done and then the model will predict whether the Input is fraudulent or not by sending the result to our html page.

3.13 User Interface

In Frontend creation I have made a user interactive page where users can enter their input values to our application. In their frontend page I have made a form which has beautiful styling with CSS. This HTML user input data is transferred in variable format to the backend. Made these html fully in a decoupled format.

3.14 Deployment

I have deployed my model in heroku and aws cloud platform.