Architecture Design

# Credit Card Default Prediction

Revision Number – 1.1

## Last Date of Revision : 16-06-2022

Kollipati Vamshi Krishna

Document Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Description | Author |
| 15-06-2022 | 1.0 | Abstract  Introduction  Architecture | Vamshi |
| 15-06-2022 | 1.1 | Architecture Design | Vamshi |

## Contents

Document Version Control ………………………………………2

Abstract …………………………………………………………... 4

1. Introduction ……………………………………………………………………………. 5
   1. Why this Architecture Design Document ? ………………………………….. 5
2. Architecture

….…………………………………………………………………………………… 5

1. Architecture Design

……………………………………………………………………………………… 5

* 1. Data Collection

……….…………………………………………………………………………. 5

* 1. Data Description

…..…………………………………………………………………………… 6

* 1. Importing Data into Database

…………………………………………………………….. 6

* 1. Exporting Data from Database

…………………………………………………………… 6

* 1. Data Preprocessing

………………………………………………………………………..…. 7

* 1. Modelling Process

………………....………………………………………………...…….….. 7

* 1. UI Integration ……………………………………………………………………………..…….. 7
  2. Data from User

………………………………………………………………………………..… 8

* 1. Data Validation

…….……………………………………………………………………….…... 8

* 1. Rendering the Results

…………….…………………….………….………………………. 8 3.11

Deployment

………………………………………….………………………………………...

8

### Abstract

Recent trends are to build tall buildings in big cities as a way out of the current housing overpopulation problem. These new structures unveil problems that if not addressed in time could cause catastrophes of unimaginable impact. Some of those problems is the incidence of a fire threat happening upstairs in one of those buildings, medical emergencies due to any road accidents or mob that may cause threat to the human kind. This work discusses the implementation of the unmanned ground vehicles to spot the real location of the medical emergencies due to road mishap, mob or illegal activities such as hooliganism, snatching, robbery and the fire emergency and accordingly channelize or route them to the concerned helpline for quick mitigation and avoid disaster.

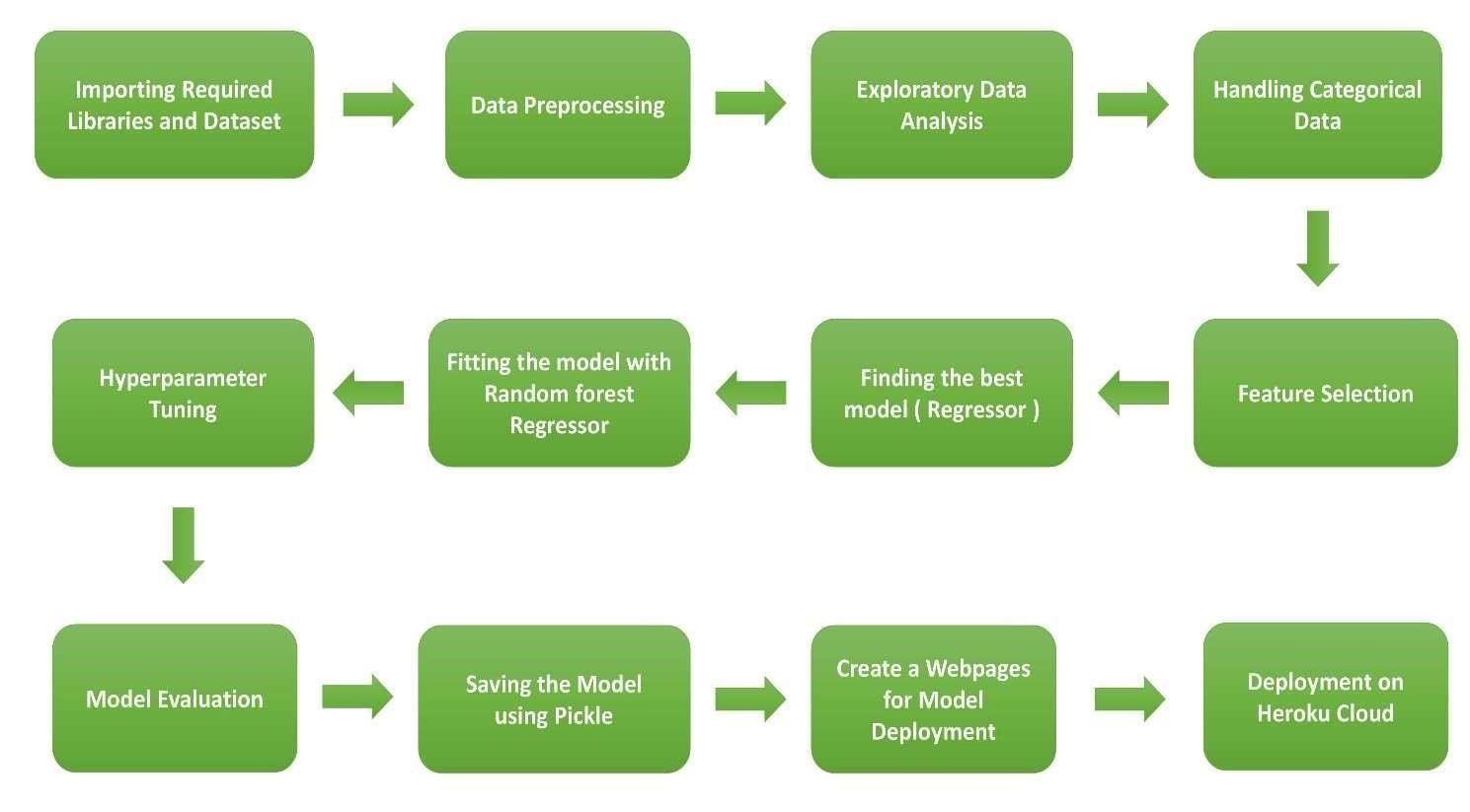
### 1. Introduction

1.1 Why this Architecture Design Document ?

The main objective of the Architecture design documentation is to provide the internal logic understanding of the Credit card defaults code. The Architecture design documentation is designed in such a way that the programmer can directly code after reading each module description in the documentation.

. Architecture Design

3



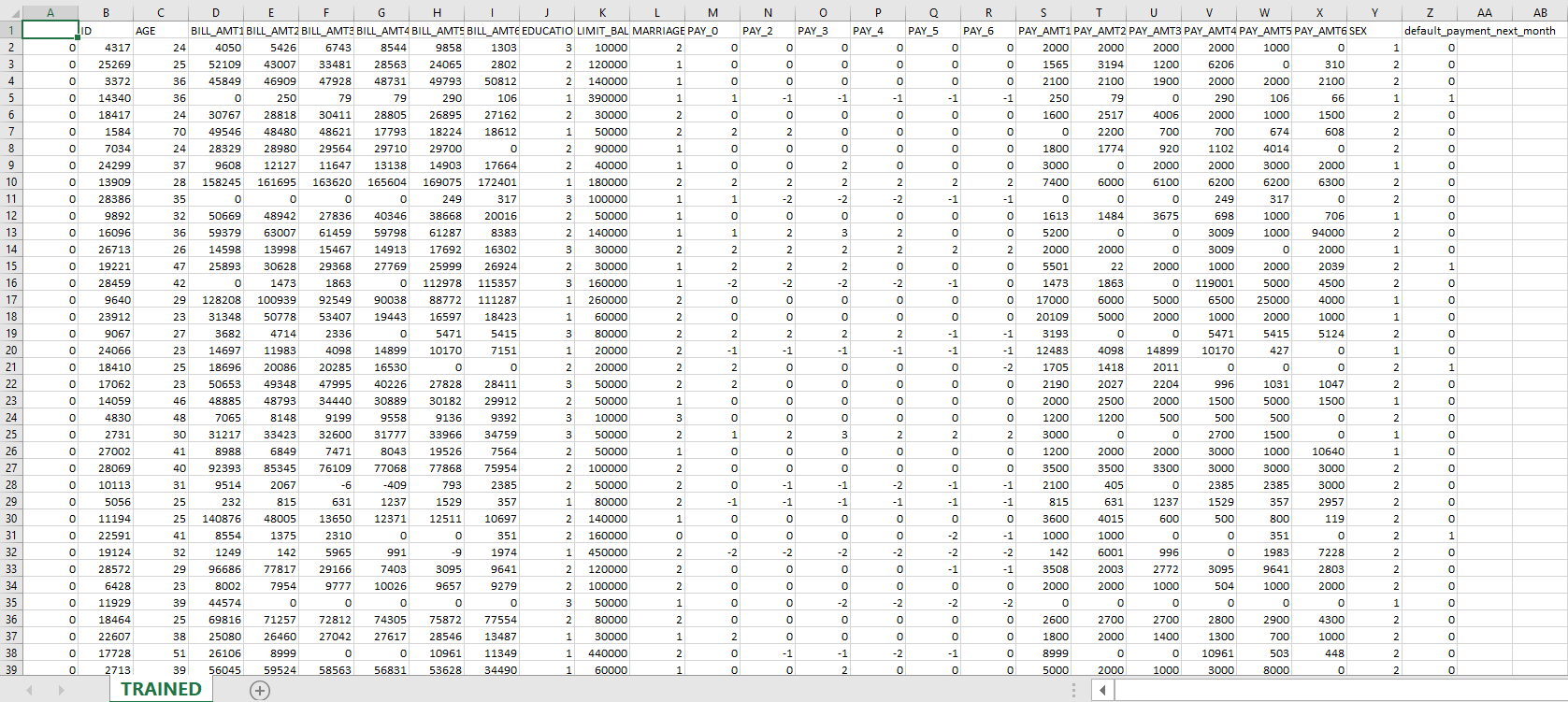
3.1 Data Collection

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.

3.2 Data Description

Credit card default is dataset publicly available on the Kaggle. The information in the dataset is present in two separated excel files named as train.xlsx and test.xlsx. Dataset contains 10683 rows which shows the information such Date of Journey, Source, Destination, Arrival Time, Departure Time, Total stops, Airlines, Additional Info and Price. The glance of the Dataset is :



3.3 Importing data into Database

Created associate API for the transfer of the info into the Cassandra info, steps performed are :

* Connection is created with the info.
* Created a info with name cardInfo.
* cqlsh command is written for making the info table with needed parameters.
* And finally, a cqlsh command is written for uploading the Knowledge Set into data table by bulk insertion.

* 1. Exporting Data from Database

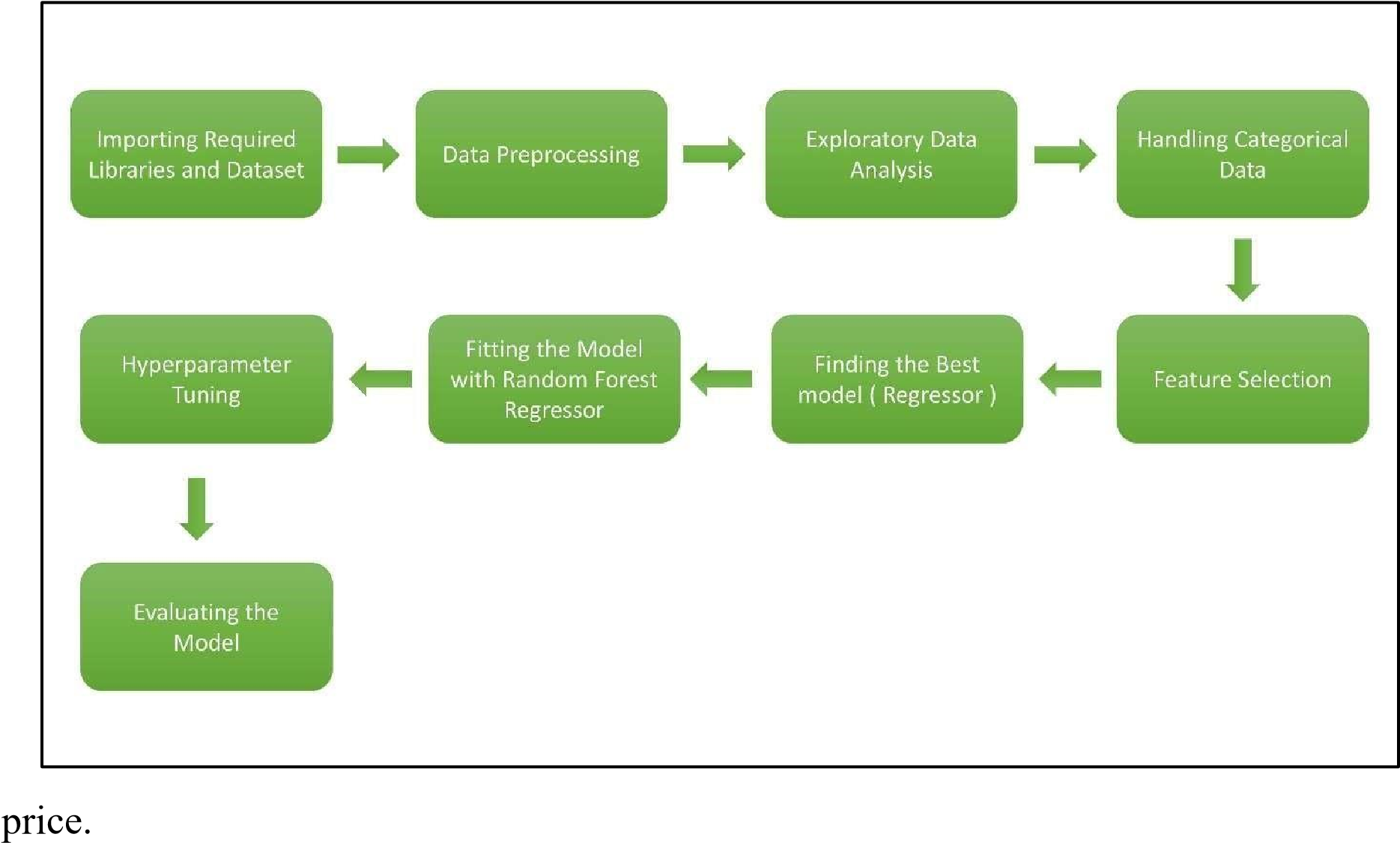
In the above created API, the download URL is also being created, which downloads the data into a csv file format.

* 1. Data Preprocessing
* Checked for info of the Dataset, to verify the correct datatype of the Columns.
* Checked for Null values, because the null values can affect the accuracy of the model.
* Converted all the desired columns into Datetime format.
* Performed One – Hot encoding on the desired columns.
* Checking the distribution of the columns to interpret its importance.

Now, the info is prepared to train a Machine Learning Model.

* 1. Modelling Process

After preprocessing the data, We visualize our data to gain insights and then these insights are randomly spread and split into two parts, train and test data. After splitting the data, we use Random Forest Regressor to model our data to predict the Flight Fare



* 1. UI Integration

Both CSS and HTML files are being created and are being integrated with the created machine learning model. All the required files are then integrated to the app.py file and tested locally.

* 1. Data from User

The data from the user is retrieved from the created HTML web page.

* 1. Data Validation

The data provided by the user is then being processed by app.py file and validated. The validated data is then sent to the prepared model for the prediction.

* 1. Rendering the Results

The data sent for the prediction is then rendered to the web page.

* 1. Deployment

The tested model is then deployed to Heroku. So, users can access the project from any internet devices.