Detail Project Report

# Credit Card Default Prediction

Revision Number – 1.2

## Last Date of Revision : 30-10-2022

Adil Anwar

Document Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Description | Author |
| 25 - 10 - 2022 | 1.0 | Abstract  Introduction  General Description | Adil Anwar |
| 28 -10 - 2022 | 1.1 | Technical Requirements  Data Requirements  Data Pre-processing Design Flow | Adil Anwar |
| 30 -10 - 2022 | 1.2 | Data from User and its validation  Rendering the Results  Deployment  Conclusion | Adil Anwar |

## Contents

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

Abstract ……………………………………………………………………... 4

1. Introduction

……………………………………………………………………………. 5

* 1. Why this DPR Document ?

………………………………………………………………….. 5

1. General Description

……………………………………………………………………………………… 5

* 1. Problem Perspective

…………………………………………………………………………. 5

* 1. Problem Statement

…………………………………………………………………………… 5

* 1. Proposed Solution …………………………………………………………………………….. 5
  2. Further Improvements

……………………………………………………………………… 6

1. Technical Requirements

………………………………………………………………………………. 6

* 1. Tools Used

………….……………………………………………………………………………..

6

1. Data Requirements

……………………………………………………………………………………… 6

* 1. Data Collection

……….…………………………………………………………………………. 6

* 1. Data Description

…..…………………………………………………………………………… 7

* 1. Importing Data into Database

…………………………………………………………….. 7

* 1. Exporting Data from Database

…………………………………………………………… 7

1. Data Preprocessing

………………………………………………………………………………………. 8

1. Design Flow

………………………………………………………………………………………..………

… 8

* 1. Model Creation and Evaluation

…………………………………………………..………. 8

* 1. UI Integration

……………………………………………………………………………..……... 8

* 1. Deployment Process

……………………………………………………………………..……. 9

* 1. Logging

………………………………………………………………………………………

…..…. 9

1. Data from User

………………………………………………………………………………………………

9

1. Data Validation

……………………………………………………………………………………….……..

9 9 Rendering the Results

…………….…………………….……………………………………………….... 9

1. Deployment

……………………………………………………………………………………………….

. 9

1. Conclusion

………………………………………………………………………………………………

…. 10

1. FAQs

………………………………………………………………………………………………

…………… 10

### 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 DPR Document ?

The main purpose of this DPR documentation is to add the necessary details of the project and provide the description of the machine learning model and the written code. This also provides the detailed description on how the entire project has been designed end-to-end.

Key points :

* Describes the design flow
* Implementations
* Software requirements Architecture of the project Non-functional attributes like:
* Reusability
* Portability
* Resource utilization

### 2. General Description

2.1 Problem Perspective

The Credit card default Prediction system is a machine learning-based default prediction model which will help us to know the possibility to default credit card payment by a customer.

2.2 Problem Statement

Financial threats are displaying a trend about the credit risk of commercial banks as the

incredible improvement in the financial industry has arisen. In this way, one of the

biggest threats faces by commercial banks is the risk prediction of credit clients. The

goal is to predict the probability of credit default based on credit card owner's

characteristics and payment history.

.

2.3 Proposed Solution

The solution proposed here is a Credit card default Prediction model can be implemented to perform above mention use case. ln this case, we have to enter the last six months bill amounts , paid amounts, payment status, credit limit, personal details. Based on the above details model predicts 0 or 1. 0 means the person won’t default ,1 means default

2.4 Further Improvements

We also analysis the data used for training the ML model by considering different occasions such as Weekday, Season or any Social reasons, considering different angle of business. If we method such information and predict the discounted , it will bring some loss to the airline companies but user can get benefit from that. If we develop these using Business perspective, this technique isn’t thought - about.

### 3. Technical Requirements

As technical requirements, we doesn’t need any specialized hardware for virtualization of the application. The user should have the device that has the access to the web and the fundamental understanding of providing the input.

3.1 Tools Used

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

* VScode is used as IDE.
* For visualization of the plots, Matplotlib, Seaborn and Plotly are used.
* 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.
* Python is used for backend development.
* GitHub is used as version control system.

### 4. Data Requirements

The Data requirements is totally supported the matter statement and also the dataset is accessible on the Kaggle within the file format of (.xlsx).

4.3 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 Credit Card Default
* 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.

4.4 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.

### 5. 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.

### 6. Design Flow



6.2 UI Integration

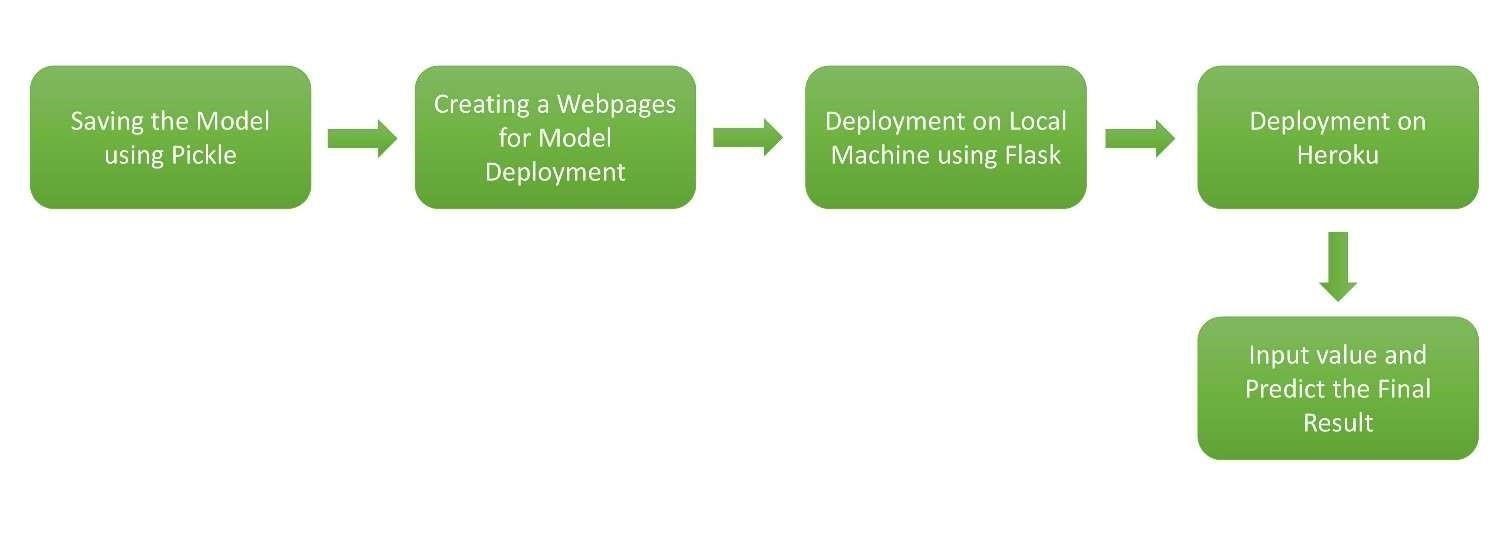
Both CSS and HTML files are being created and are being integrated with the created machine learning model. All the required files are then integrated to the app.py file and tested locally.



6.3



Deployment Process



6.4 Logging

In logging, at each if an error or an exception is occurred, the event is logged into the system log file with reason and timestamp. These helps the developer to debug the system bugs and rectifying the error.

7. Data from User

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

### 8. 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.

9. Rendering the Results

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

### 10. Deployment

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

### 11. Conclusion

The Flight Fare Prediction system will predict the price for helping the customers with the trained knowledge with set of rules. The user can use these system to recognize the approximate value of its flight fare for his or her journey.

### 12. Frequently Asked Questions ( FAQs )

Q1) What’s the source of data ?

The data for training is provided by the client in multiple batches and each batch contain multiple files.

Q2) What was the type of data ?

The data was the combination of numerical and Categorical values.

Q3) What’s the complete flow you followed in this Project ?

Refer Page no 6 for better Understanding.

Q4) After the File validation what you do with incompatible file or files which didn’t pass the validation ?

Files like these are moved to the Achieve Folder and a list of these files has been shared with the client and we removed the bad data folder.

Q5) How logs are managed ?

We are using different logs as per the steps that we follow in validation and modeling like File validation log, Data Insertion, Model Training log, prediction log etc.

Q6) What techniques were you using for data pre-processing ?

* Removing unwanted attributes.
* Visualizing relation of independent variables with each other and output variables.
* Checking and changing Distribution of continuous values.
* Removing outliers
* Cleaning data and imputing if null values are present.
* Converting categorical data into numeric values.

Q7) How training was done or what models were used ?

* Before dividing the data in training and validation set, we performed preprocessing over the data set and made the final dataset.
* As per the dataset training and validation data were divided.
* Algorithms like Linear regression, SVM, Decision Tree, Random Forest, XGBoost were used based on the recall, final model was used on the dataset and we saved that model.

Q8) How Prediction was done ?

The testing files are shared by the client. We Performed the same life cycle on the provided dataset. Then, on the basis of dataset, model is loaded and prediction is performed. In the end we get the accumulated data of predictions.

Q9) What are the different stages of deployment?

* First, the scripts are stored on GitHub as a storage interface.
* The model is first tested in the local environment.
* After successful testing, it is deployed on Heroku.