DPR
Credit Card Default Prediction Model

Revision Number – 1.2

Last Date of Revision – 25/10/2022

Document Version Control

Date	Version	Description	Author
24-10-2022	1.0	Abstract, Introduction	
25-10-2022	1.1	Deployment and Process	
25-10-2022	1.2	Q and A	

Contents

Abstract

In present day majority of people are using credit card for online shopping, EMI payments. Moreover banks are encouraging customers to use it by approving them freely without any initial charges. Because credit card interest income is the major share of profit for all commercial banks. Even if customers are not willing to buy, bank marketing team reaches out through phone call to sell it. At first the interest rate are normal to grope the customer into the practice of using it, later they will levy charges on everything to loot money from the customers. Due to this reason many customers are facing difficulty in repaying the credit amount on time. If they missed the deadline the interest rate will get multiplied leading to defaulting the account if left unpaid even for a short span of time. The model we are going to build will consider all inputs from user to predict whether the account will be defaulted or not based on various parameters.

INTRODUCTION

Why this DPR Documentation?

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

1 Description

1.1 Problem Perspective

The credit card default prediction is a machine learning model which helps us to predict whether the account will be defaulted or not..

1.2 Problem Statement

To create machine learning model to detect anomalies in credit repayment transaction and predict whether the account will be defaulted or not.

1.3 Proposed Solution

The proposed model will be capable of studying customers based on various parameters like gender, education qualification, marital status, previous repayment history and current outstanding balance. Based on that it can predict the account default status in advance and notify the respective bank official for further action.

1.4 Solution Improvements

Further we can enhance our model with additional information like customer expenditure pattern, CIBIL score, loan EMI or availability of any fixed deposit mapped to the account. These informations will improve the prediction accuracy and can suggest alternative methods to retrieve the pending dues in-case of future default.

2 Technical Requirements

There are no hardware requirements required for using this application, the user must have an interactive device which has access to the internet and must have the basic understanding of providing the input. And for the backend part the server must run all the software that is required for the processing the provided data and to display the results.

2.1 Tools Used

- Python 3.8 is used as the programming language and frame works like numpy, pandas, sklearn and other modules for building the model.
- VSCode is used as IDE.
- For visualizations seaborn and parts of matplotlib are being used.
- For data collection Cassandra database is being used.
- Front end development is done using HTML/CSS.
- Flask is used for both data and backend deployment.
- GitHub is used for version control.
- Heroku is used for deployment.

3 Data Requirements

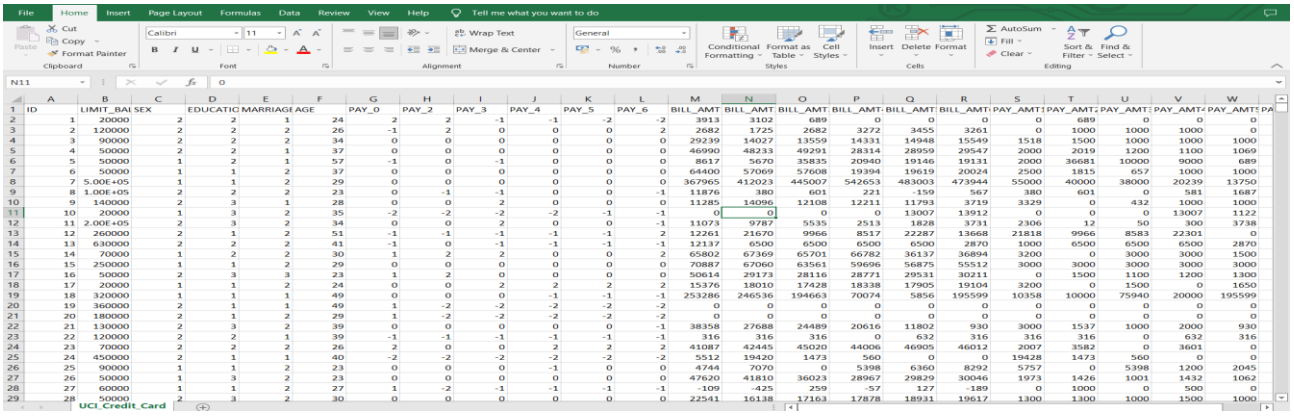
The data requirement is completely based on the problem statement. And the data set is available on the Kaggle in the form of csv(.csv). As the main theme of the project is to get the experience of real time problems, we are again importing the data into the Cassandra data base and exporting it into csv format.

3.1 Data Gathering from Main Source

The data for the current project is being gathered from Kaggle dataset, the link to the data is: https://www.kaggle.com/datasets/uciml/default-of-credit-card-clients-dataset

3.2 Data Description

There are about 10k+ records of flight information such as airlines, data of journey, source, destination, departure time, arrival time, duration, total stops, additional information, and price. A glance of the dataset is shown below:



3.3 Import Data into Database

Created an api for the upload of the data into the Cassandra database, steps performed are:

- Connection is made with the database.
- Created a database with name flight fare.
- Cqlsh command is written for creating the data table with required parameters.
- And finally, a cqlsh command is written for uploading the dataset into the data table by bulk insertion.

3.4 Export Data from Database

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

4 Data Pre-Processing

Steps performed in pre-processing are:

- First the data types are being checked and found only the price column is of type integer.
- Checked for null values if there are few null values, those rows are dropped.
- Converted all the required columns into the date time format.
- Performed one-hot encoding for the required columns.
- Scaling is performed for required data.

And, the data is ready for passing to the machine learning algorithm.

5 Design Flow

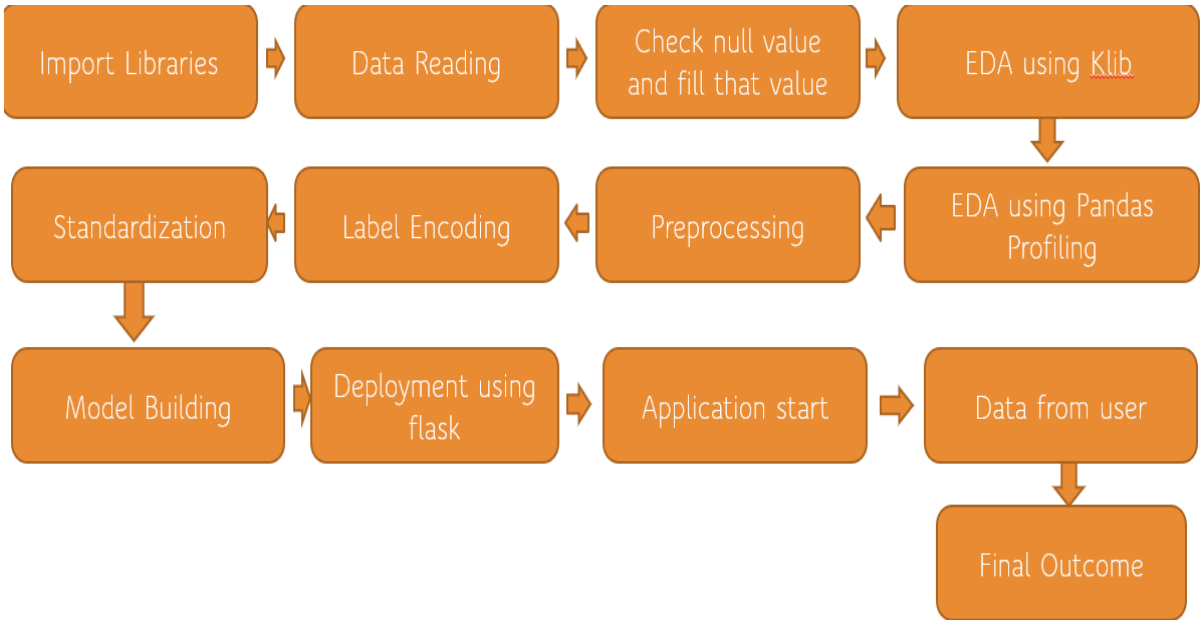
5.1 Modeling

The pre-processed data is then visualized and all the required insights are being drawn. Although from the drawn insights, the data is randomly spread but still modeling is performed with different machine learning algorithms to make sure we cover all the possibilities. And finally, as expected random forest regression performed well and further hyperparameter tuning is done to increase the model's accuracy.

5.2 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.

5.3 Modelling Process&5.4 Deployment Process



6 Data from User

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

7 Data Validation

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

8 Rendering the Results

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

9 Deployment

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

Conclusion

The flight fare prediction can predict the price based on the trained data set in the algorithm. Hence the user can know the approximate cost for their journey.

Q & A:

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.

Q 2) What was the type of data?

The data was the combination of numerical and Categorical values.

Q 3) What's the complete flow you followed in this Project?

Refer Page no 6 for better Understanding.

Q 4) 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.

Q 5) 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.

Q 6) 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.

Q 7) How training was done or what models were used?

- Before dividing the data in the training and validation set, we performed pre-processing 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.

Q 8) 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 the dataset, the model is loaded and prediction is performed. In the end we get the accumulated data of predictions.

Q 9) 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.