Architecture

Architecture

Credit Card Default Prediction

Revision Number – 1.0

Last Date of Revision – 16-05-2023

**ARYA DIXIT**

1

Architecture



**Liabaries Requirment**

Flask

flask\_cors

pandas

numpy

scikit-learn

ARCHITECTURE

**Contents**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **[Abstract](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit" \l "heading=h.gjdgxs)** | | | | | | | | | | | | | | | [4](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit" \l "heading=h.gjdgxs) | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **[Introduction](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit" \l "heading=h.30j0zll)** | | |  | | | | | | | | | | | | [4](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit" \l "heading=h.30j0zll) | | | |
|  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | **[Why this Architecture Design documentation?](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit" \l "heading=h.1fob9te)** | | | | | | | | | | | | | |  | | [4](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit" \l "heading=h.1fob9te) |  |
| **[1 Architecture](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit" \l "heading=h.3znysh7)** | | | |  | | | | | | | | | | | [4](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit" \l "heading=h.3znysh7) | | | |
|  | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **[2 Architecture design](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit" \l "heading=h.3dy6vkm)** | | | | | | | |  | | | | | | | [5](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit" \l "heading=h.3dy6vkm) | | |  |
|  |  | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | **[2.1 Data gathering from main source](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit" \l "heading=h.1t3h5sf)** | | | | | | | | | | | | | | [5](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit" \l "heading=h.1t3h5sf) | | |  |
|  |  | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | **[2.2 Data description](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit" \l "heading=h.4d34og8)** | | | | | | | | | | | | |  | [5](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit" \l "heading=h.4d34og8) | | |  |
|  |  | | | |  |  |  |  |  |  |  |  |  | |  |  |  |  |
|  | **[2.3 Upload data into Cassandra](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit" \l "heading=h.2s8eyo1)** | | | | | | | | | | | |  | | [5](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit" \l "heading=h.2s8eyo1) | | | |
|  |  | | | |  |  |  |  | |  |  |  |  | |  |  |  |  |
|  | **[2.4 Export data from database](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit" \l "heading=h.17dp8vu)** | | | | | | | | | | |  |  | | [5](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit" \l "heading=h.17dp8vu) | | |  |
|  |  | | | |  |  |  |  | |  |  |  | | |  |  |  |  |
|  | **[2.5 Data pre-processing](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit" \l "heading=h.3rdcrjn)** | | | | | | | | | | |  | | |  | [5](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit" \l "heading=h.3rdcrjn) | |  |
|  |  | | | |  |  |  |  | |  |  | | | |  | |  |  |
|  | **[2.6 Modelling](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit" \l "heading=h.26in1rg)** | | | |  | | | | |  | | | | |  | | [6](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit" \l "heading=h.26in1rg) | |
|  |  | | | |  |  |  |  | | |  | | | |  | |  |  |
|  | **[2.7 UI integration](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit" \l "heading=h.lnxbz9)** | | | | |  | | | | | | | | | [6](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit" \l "heading=h.lnxbz9) | | |  |
|  |  | | | | |  |  |  | | |  | | | |  | |  |  |
|  | **[2.8 Data from user](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit" \l "heading=h.35nkun2)** | | | | | | | | | | | | | | [6](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit" \l "heading=h.35nkun2) | | |  |
|  |  | | | | |  |  |  | | |  | | | |  | |  |  |
|  | **[2.9 Data validation](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit" \l "heading=h.1ksv4uv)** | | | | | | |  | | | | | | | [6](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit" \l "heading=h.1ksv4uv) | | |  |
|  |  | | | | |  | |  | | |  | | | |  | |  |  |
|  | **[2.10 Rendering the results](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit" \l "heading=h.44sinio)** | | | | | | | | | |  | | | | [6](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit" \l "heading=h.44sinio) | | |  |
|  | **[2.11 Deployment](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit" \l "heading=h.2jxsxqh)** | | | | | | | | | |  | | | |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

3

ARCHITECTURE

Abstract

Our aim is to develop a Machine learning model and testing the model by using the data in relating to previous 6 months payment behaviour which is behavioural data and personal information which is demographic data as input of a client is used for this study.

The research study is conducted using Random Forest Algorithm , Decision tree , Logistic Regression. Our aim is to identify that credit card customer is likely to default in the coming month.

Credit risk plays a major role in the banking industry. Banking's main activities include granting loans, credit cards,investments, mortgages, etc. Credit cards are one of the fastest growing financial services offered by banks in recent years.

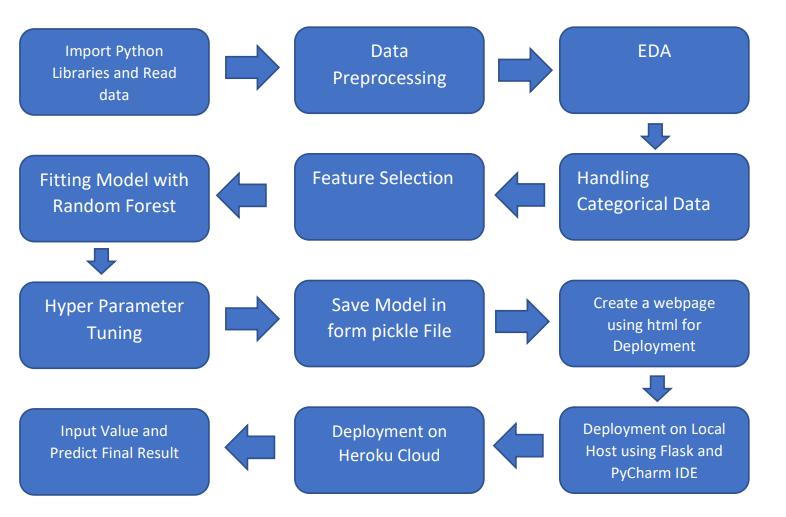
However, as the number of credit card users increases, banks are facing rising credit card failure rates. Therefore,data analytics can provide solutions to address current phenomena and manage credit risk.

Introduction

Why this Architecture Design documentation?

The main objective of the Architecture design documentation is to provide the internal logic understanding of the flight fare prediction 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.

1 Architecture



4

ARCHITECTURE

2 Architecture design

architecture is design in way that is start from importing the python libraries and load the data then process the data then perform the exploratory analysis .then feature scaling then train the model for prediction and predict the output.

2.1 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>

2.2 Data description

This dataset contains information on default payments, demographic factors, credit data, history of payment, and bill statements of credit card clients in Taiwan from April 2005 to September 2005

2.3 Upload data into Cassandra

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 defaulter\_creditcard.
* Cqlsh command is written for creating the data table with required parameters.
* And finally, a cqlsh command is written for uploading the dataset into data table by bulk insertion.

2.4 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.

2.5 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 as there are few null values, those rows are dropped.
* Converted all the required column 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.

2.6 Modelling

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 modelling 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.

2.7 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. Note I did not make the CSS and HTML File .

2.8 Data from user

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

2.9 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.

5

2.10 Rendering the results

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

2.11 Deployment

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

6

**ScreenShot of the App Interface which I will deploy**

