https://lh6.googleusercontent.com/yOGQaiPJO2d4m_1cZJqtubj4nehnEfBZGO7PvkXWc1kEtV5PT71rQ15Eyi3iJV5FFCZGgREJgdJTHrEj4h5WdBQnh4mQGe4BlwfVJq7FyQdU5agG21Bvri21hScJVn7TN_cC7XeFUYyFQjGArmbCKUs

LOW LEVEL DOCUMENT **Travel Package Purchase Prediction**

***September 2, 2022***

***Written BY OTHMANE ZOUBAIRI***

1 | P a g e

FOOD RECOMMENDATION LLD ii

iii LOW LEVEL DESIGN (LLD)

Contents

**1. Introduction**.............................................................................................1

**1.1. What is Low-Level design document?** .......................................................2

**1.2.Scope**.......................................................................................................3

**2. Architecture**...................................................................................................4

**Architecture Description**...................................................................................4

**Data Description**..............................................................................................5

**data ingestion………………………………………………………………………………………5**

**Data validation…………………………………………………………………………………………6**

**Data Transformation**.......................................................................................6

**Model Trainer**...............................................................................................7

**Model Evaluation**............................................................................................7

**Model Pusher**...................................................................................................8

**Deployment**......................................................................................................9.

**Interface …………………………………………………………………………………………………10**

**Test Cases** .........................................................................................................

**DOCUMENT VERSION CONTROL**

Change Record

|  |  |  |  |
| --- | --- | --- | --- |
| Date Date | Version Version | Comments Comments | Author  Author |
| 12/07/2022 12/07/2022 | 0.1 0.1 | Introduction and  architecture defined | othmane zoubairi |
| 28/07/2022 28/07/2022 | 0.2 0.2 | Architecture updated  Architecture updated  and unit test case  and unit test case  defined  defined | othmane zoubairi |

Review

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Reviewer | Comments |
|  |  |  |  |

Approval Status

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Reviewer | Approved By | Comments |
|  |  |  |  |

ABSTRACT

The Travel company wants to enable and establish a viable business model to expand the customer base. One of the ways to expand the customer base is to introduce a new offering of packages.

The goal is to predict the probability of Package Purchase based on the travel's characteristics and  Travel history for the Client.

With the help of Data Science and Machine learning technology,  I developed an application, which allows a banker to determine  the probability Of Travel Purchase in just a few seconds.

1 Introduction

1.1 What is Low-Level design document?

The goal of LLD or a Low-level design document is to give an  internal logical design of the actual program code for the Credit  Card Default Probability Prediction. LLD describes the class  diagrams with the methods and relations between classes and

the program specs. It describes the modules so that the  programmer can directly code the program from the document.

1.2 Scope

**Low-level design (LLD) is a component level design process  that follows a step-by-step refinement process. This process  can be used for designing data structures, required software  architecture, source code and ultimately, performance  algorithms. Overall, the data organization may be defined during  requirement analysis and then defined during data design work.**

2 Architecture



3 Architecture Description

3.1 Data Collection

For training and testing the model, I used the public set  available in Ineuron intership :

URL – https://raw.githubusercontent.com/manotti22/Travel-Package-Prediction-Projekt/main/TourismData.csv

3.2 Data Dictionary

Customer details:[¶](https://www.kaggle.com/code/dscodingp19/travel-package-purchase-prediction" \l "Customer-details:" \t "_self)

1. CustomerID: Unique customer ID
2. Age: Age of customer
3. TypeofContact: How customer was contacted (Company Invited or Self Inquiry)
4. CityTier: City tier depends on the development of a city, population, facilities, and living standards.
5. Occupation: Occupation of customer
6. Gender: Gender of customer
7. NumberOfPersonVisiting: Total number of persons planning to take the trip with the customer
8. PreferredPropertyStar: Preferred hotel property rating by customer
9. MaritalStatus: Marital status of customer
10. NumberOfTrips: Average number of trips in a year by customer
11. Passport: The customer has a passport or not (0: No, 1: Yes)
12. OwnCar: Whether the customers own a car or not (0: No, 1: Yes)
13. NumberOfChildrenVisiting: Total number of children with age less than 5 planning to take the trip with the customer
14. Designation: Designation of the customer in the current organization
15. MonthlyIncome: Gross monthly income of the customer

Label Column ;

1. ProdTaken: Whether the customer has purchased a package or not (0: No, 1: Yes)

3.3 Variable Information

This is a classification problem. The target variable is  Prodtaken . The aim of the project is to predict  the probability of Purchase Travel Package given various attributes of the  customer given below.

4. Data Ingestion

* data in data ingestion folder
* Split data in train and test data
* Saving data in data ingestion folder

4.1 Data Validation

* data in data validation folder
* data drift and EDA
* Report in evidently to show and analysis Data Distribution
* Saving report in data validation Folder

4.2 Data Transformation

* data in data Transformation folder
* Using Pipeline and columns transformer to processing Data
* Splitting data in array by Standard Scalar and different technique
* Putting train array data and test array data in transformation folder and creating prepossessing Object file.
* Saving Processing pkl in data transformation folder for future Transformation

1. **Model Trainer**

* We have built various models like Descision Tree Classifier, , Gradient Boosting and Bagging Classifier, etc. • Each of the above models was built taking their default parameters.
* We have use Gridsearch CV to have best model in comparison all models
* We save best model file in Model Trainer folder

5.1 Data Evaluation

* we used Metric for Classification : Recall Score, F1 Score and Model accuracy In train and test model to evaluate our best model
* we save the evaluation model in evaluation Folder
* Bagging Classifier was the best Model
* Reason For Choosing This Model :
* Apart From a good training and test score, the reasons for  choosing this model are as follows :
* • Can handle missing values.
* • Can work well even on imbalanced data.

**5.2 Model Pusher**

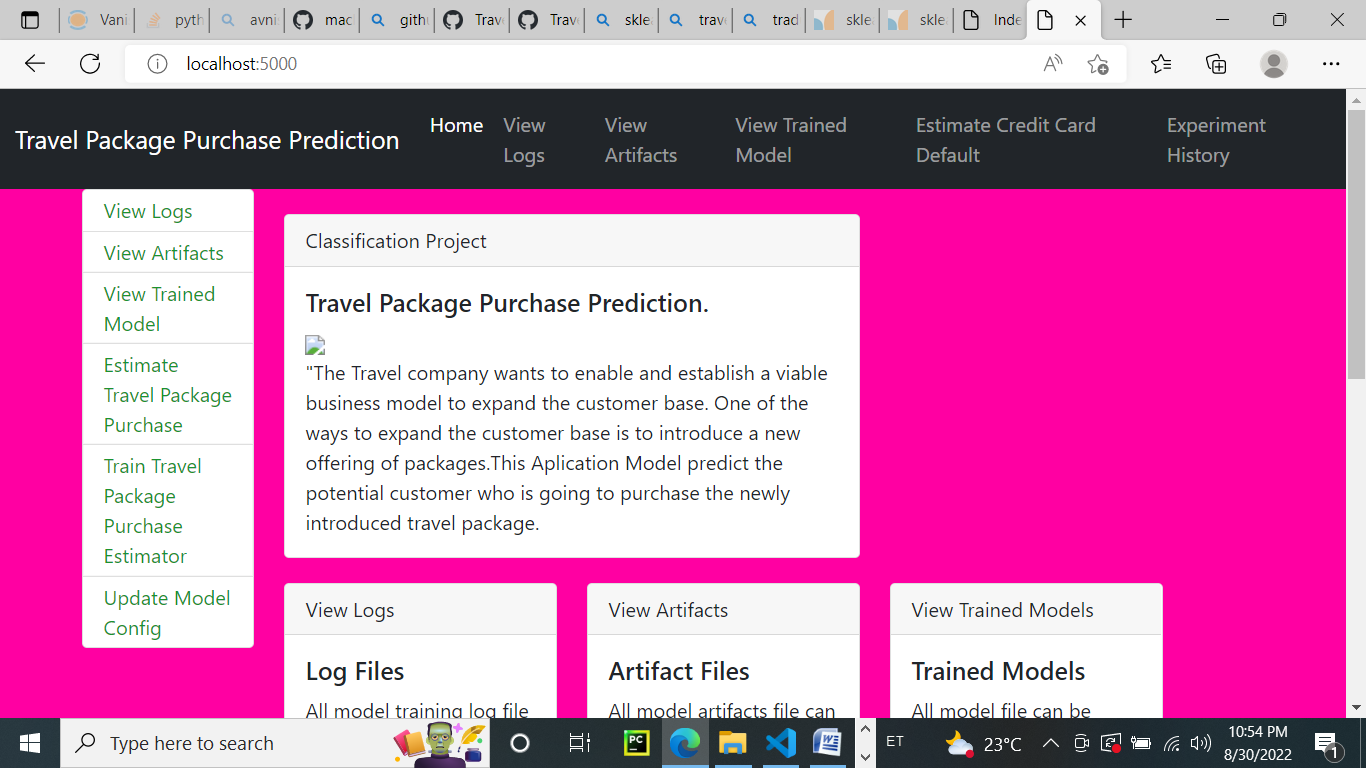
* **The best model was Bagging Classifier using GridSearchCV.**
* **The model has been saved in ‘Model.pkl’**
* **If we have an updated better model by new data ,**

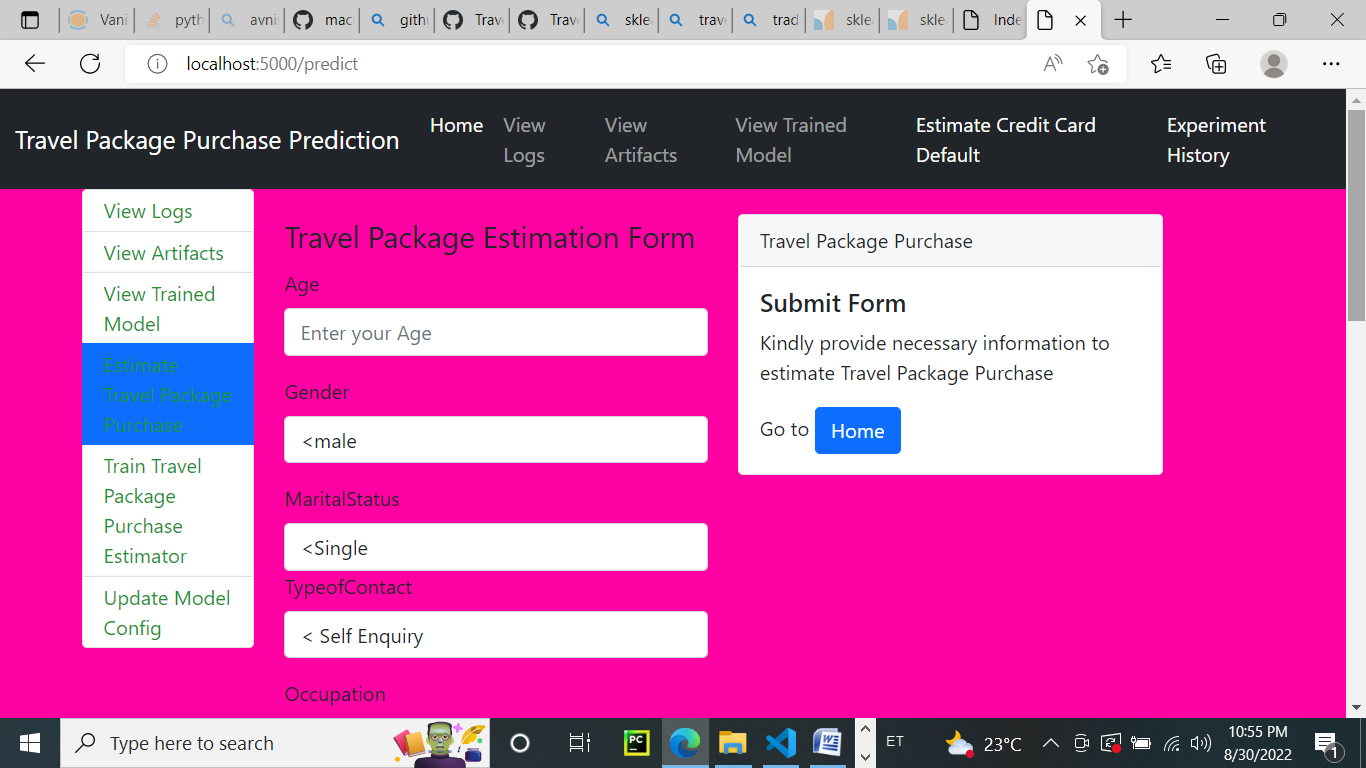
**we can save it in model pusher save models folder**

**5.3 Heroku Deplyement**

* **I deployed the application on the web using Heroku**
* **The deployment part of the code runs in the “app.py” file, connecting with the web page designed using HTML with CSS  styles.The html front end template can be found in templates  folder.**

6 User Interface





  7 Unit Test Cases

|  |  |  |
| --- | --- | --- |
| Test Case Description | Pre-Requisite | Expected-Result |
| Verify whether the  Application URL is  accessible to the user | 1. Application URL  should be defined | Application URL should be accessible to the user. |
| Verify whether the  Application loads completely  for the user when the URL is  accessed. | 1. Application URL is Accessible.  2. Application is  deployed. | The application should load completely for the user  when the URL is accessed. |
| Verify whether user can edit all the input fields | 1. Application URL is Accessible.  2. Application loads completely for the  user.  3. All the input fields Loaded. | User should be able to edit  all the input fields |
| Verify whether user gets “Predict” button to make predictions on the given inputs | 1. Application URL is Accessible.  2. Application loads completely for the  user.  3. All the input fields Loaded. | User should get a “Predict” button to make predictions  on the given inputs. |
| Verify whether user is  Presented with  recommended results on  clicking the “Predict” button | 1. Application URL is Accessible.  2. Application loads completely for the  user.  3. All the input fields Loaded. | Users should be presented with recommended results  on clicking the “Predict”  button. |