Low Level Design

Customer-Personality-Analysis

Written By	GURURAJ S
Version	0.1
Date	20 – MAY - 2023

Document Control

Change Record:

Version	Date	Author	Comments

Reviews:

Version	Date	Reviewer	Comments

Approval Status:

Version	Review Date	Reviewed By	Approved By	Comments

Contents

1. Introduction

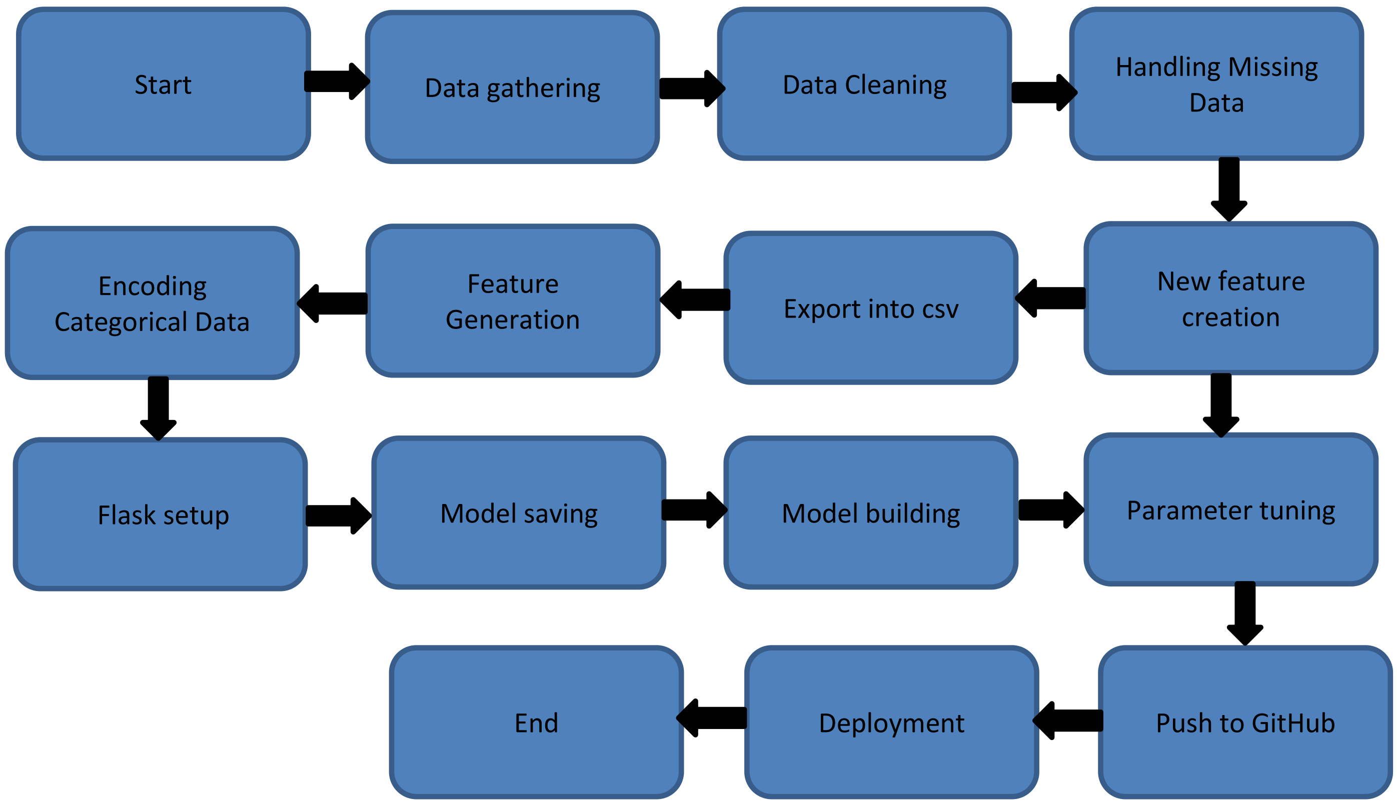
1.1. What is Low-Level design document?

The goal of LLD or a low-level design document (LLDD) is to give the internal logical design of the actual program code for Customer-Personality-Analysis LLD describes the class diagrams with the methods and relations between classes and 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 refined during data design work

2. Architecture



3. Architecture Description

3.1. Data Description

People:

- ID: Customer's unique identifier
- Year_Birth: Customer's birth year
- Education: Customer's education level
- Marital_Status: Customer's marital status
- Income: Customer's yearly household income
- Kidhome: Number of children in customer's household
- Teenhome: Number of teenagers in customer's household
- Dt_Customer: Date of customer's enrollment with the company
- Recency: Number of days since customer's last purchase
- Complain: 1 if the customer complained in the last 2 years, 0 otherwise

Products:

- MntWines: Amount spent on wine in last 2 years
- MntFruits: Amount spent on fruits in last 2 years
- MntMeatProducts: Amount spent on meat in last 2 years
- MntFishProducts: Amount spent on fish in last 2 years
- MntSweetProducts: Amount spent on sweets in last 2 years
- MntGoldProds: Amount spent on gold in last 2 years

Promotion:

- NumDealsPurchases: Number of purchases made with a discount
- AcceptedCmp1: 1 if customer accepted the offer in the 1st campaign, 0 otherwise
- AcceptedCmp2: 1 if customer accepted the offer in the 2nd campaign, 0 otherwise
- AcceptedCmp3: 1 if customer accepted the offer in the 3rd campaign, 0 otherwise
- AcceptedCmp4: 1 if customer accepted the offer in the 4th campaign, 0 otherwise
- AcceptedCmp5: 1 if customer accepted the offer in the 5th campaign, 0 otherwise
- Response: 1 if customer accepted the offer in the last campaign, 0 otherwise

Place:

- NumWebPurchases: Number of purchases made through the company’s website
- NumCatalogPurchases: Number of purchases made using a catalogue
- NumStorePurchases: Number of purchases made directly in stores
- NumWebVisitsMonth: Number of visits to company’s website in the last month

3.2. Data Gathering

Dataset link :- Link
We got csv files . then we start data preparation on given Dataset purpose and EDA.

3.3. Data Cleaning

We observed some inconsistency in DataFrame I,e we change in date which is present in string to convert it int o date and time and year convert to present age of customer also we check other columns data type and other info

3.4. Handling Missing Data

There was missing values in one columns i.e.,. Income so we drop nan values from the given dataset

3.5. Feature Generation

For Item_Identifier feature, we have around 0 unique values. This column would not be important in dataset w e would drop we change in date which is present in string to convert it into date and time and year convert to p resent age of customer also we check other columns data type and other info

3.6. Feature Selection

We included all the features for model training except for Outlet_Identifier and of course Outlet_Establishment_Year and Item_Identifier

3.7. Encoding Categorical Data

Label Encoding was used for ordinal columns were yes:1 and no:0
Gender male:1 and female:0

3.8. Model Selection

For model selection we had used Pycaret library that will perform multiple operations and give algorithm best for data. sklearn had given decision Tree as best for our data.

3.9 Parameter Tuning

Parameters are tuned using Grid searchCV. The parameters are tuned on Gradient Boost model.

3.10 Model Building

After doing all kinds of pre-processing operations mention above and hyperparameter tuning data is passed to Gradient Regressor model It was found that it performs best with the accuracy i.e. 93%

3.11 Model Saving

Model is saved using pickle library in `.pkl` format.

3.12 Flask Setup

After saving the model, the API building process. Web application creation was created using. Whatever the data user will enter and then that data will be extracted by the model to predict the prediction of cluster.

3.13 Github

The whole project directory will be pushed into the GitHub repository

3.14 Deployment

The cloud environment was set up and the project was deployed from GitHub into the render.com cloud platform.

.

4. 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 URLis accessed	1.Application URLis accessible 2.Application is deployed	The Application should load completely for the user when theURL is accessed
Verify whether a user is able to see inputfields while opening the application	1.Application is accessible 2.The user is able to see the input fields	Users should be able to see inputfields on logging in
Verify whether a user is able to enter the input values.	1.Application is accessible 2. The user is able to see the input fields	The user should be able to fill the input field
Verify whether a user gets predict button to submit the inputs	1.Application is accessible 2.The user is able to see the input fields	Users should get Submit button to submit the inputs

Verify whether a user is presented with recommended results on clicking submit	1. Application is accessible 2. The user is able to see the input fields. 3. The user is able to see the submit button	Users should be presented with recommended results on clicking submit
Verify whether a result is in accordance with the input that the user has entered	1. Application is accessible 2. The user is able to see the input fields. 3. The user is able to see the submit button	The result should be in accordance with the input that the user has entered