Low Level Design

Market Basket Project on E-Commerce

Brazilian E-Commerce Public Dataset by Olist

Written By	Mohammed Ashfaq ali
Document Version	0.3
Last Revised Date	Nov 2022

Document Control

Change Record:

Version	Date	Author	Comments
0.1	01-Nov-2022	Mohammed Ashfaq Ali	Introduction & Architecture defined
0.2	02– Nov -2022	Mohammed Ashfaq Ali	Architecture & Architecture Description appended and updated
0.3	04 – Nov -2022	Mohammed Ashfaq Ali	Unit Test Cases defined and appended

Reviews:

Version	Date	Reviewer	Comments
0.2	19 – Nov -2022		Document Content , Version Control and Unit Test Cases to be added

Approval Status:

Version	Review Date	Reviewed By	Approved By	Comments

Contents

1. Introduction

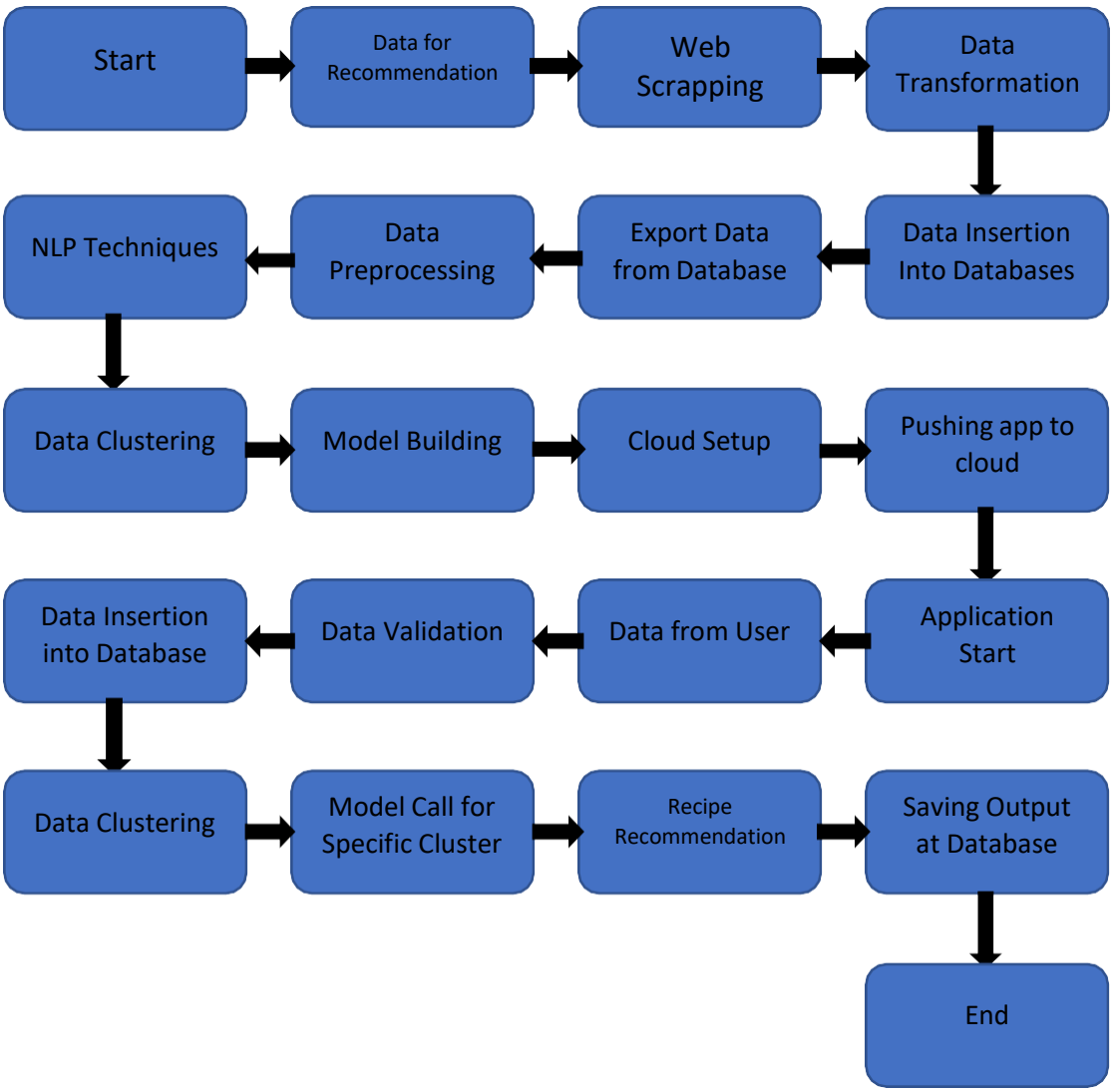
1.1. What is Low-Level design document?

The goal of LLD or a low-level design document (LLDD) is to give the internal logical design of the actual program code for Food Recommendation System. 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

- This is a Brazilian ecommerce public dataset of orders made at Olist Store. The dataset has information of 100k orders from 2016 to 2018 made at multiple marketplaces in Brazil. Its features allows viewing an order from multiple dimensions: from order status, price, payment and freight performance to customer location, product attributes and finally reviews written by customers. We also released a geolocation dataset that relates Brazilian zip codes to lat/lng coordinates.
- This is real commercial data, it has been anonymized, and references to the companies and partners in the review text have been replaced with the names of Game of Thrones great houses.

3.2. Web Scrapping

- In order to create a more complete recipe collection we will need some more datasets.

3.3. Data Transformation

- In the Transformation Process, we will convert our original dataset which is in JSON format to CSVFormat. And will merge it with the Scrapped dataset.

3.4. Data Insertion into Database

a. Database Creation and connection - Create a database with name passed. If the database is already created, open the connection to the database.
b. Table creation in the database.
c. Insertion of files in the table

3.5. Export Data from Database

- Data Export from Database - The data in a stored database is exported as a CSV file to be used forData Pre-processing and Model Training.

3.6. Data Pre-processing

- Data Pre-processing steps we could use are Null value handling, stop words removal, punctuation removal, Tokenization, Lemmatization, TFIDF, Imbalanced data set handling, Handling columns withstandard deviation zero or below a threshold, etc.

3.7. Data Clustering

- K-Means algorithm will be used to create clusters in the pre-processed data. The optimum number of clusters is selected by plotting the elbow plot. The idea behind clustering is to implement differentalgorithms to train data in different clusters. The K-means model is trained over preprocessed data and the model is saved for further use in prediction

3.10. Model Building

- After clusters are created, we will find the best model for each cluster. For each cluster, algorithms will be passed with the best parameters derived from Grid-Search. We will calculate the AUC scoresfor models and select the model with the best score. Similarly, the models will be selected for each cluster. All the models for every cluster will be saved for use in Recommendation.

3.11. Data from User

- Here we will collect physiological data from user such as user id, product id, purchased date, user name etc.

3.12. Data Validation

- Here Data Validation will be done, given by the user

3.13. User Data Inserting into Database

- Collecting the data from the user and storing it into the database. The database can be either MySQL or Mongo DB.

3.14. Data Clustering

- The model created during training will be loaded, and clusters for the user data will be predicted.

3.15. Model Call for Specific Cluster

- Based on the cluster number, the respective model will be loaded and will be used topredict/Recommend the data for that cluster.

3.16. Recipe Recommendation & Saving Output in Database

- After calling model Recipe/Output will be recommended, this output will be saved in Database and itwill be used to show the same Output if other users provide the same data.

3.17. Deployment

- We will be deploying the model to AWS or Heroku.
- This is a workflow diagram for the Recipe Recommendation.

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 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 the User is able to sign up in the application	1. Application is accessible	The User should be able to sign up in the application
Verify whether user is able to successfully login to the application	1. Application is accessible 2. User is signed up to the application	User should be able to successfully login to the application
Verify whether user is able to see input fields on logging in	1. Application is accessible 2. User is signed up to the application 3. User is logged in to the application	User should be able to see input fields on logging in
Verify whether user is able to edit all input fields	1. Application is accessible 2. User is signed up to the application 3. User is logged in to the application	User should be able to edit all input fields
Verify whether user gets Submit button to submit the inputs	1. Application is accessible 2. User is signed up to the application 3. User is logged in to the application	User should get Submit button to submit the inputs
Verify whether user is presented with recommended results on clicking submit	1. Application is accessible 2. User is signed up to the application 3. User is logged in to the application	User should be presented with recommended results on clicking submit
Verify whether the recommended results are in accordance to the selections user made	1. Application is accessible 2. User is signed up to the application 3. User is logged in to the application	The recommended results should be in accordance to the selections user made
Verify whether user has options to filter the recommended results as well	1. Application is accessible 2. User is signed up	User should have options to filter the recommended results as well

	to the application 3. User is logged in to the application	
Verify whether KPIs modify as per the user inputs for the user's health	1. Application is accessible 2. User is signed up to the application 3. User is logged in to the application	KPIs should modify as per the user inputs for the user's health
Verify whether the KPIs indicate details of the suggested recipe	1. Application is accessible 2. User is signed up to the application 3. User is logged in to the application	The KPIs should indicate details of the suggested recipe