Low Level Design

Travel Package Purchase Prediction

|  |  |
| --- | --- |
| Written By | Sonali Dhamanekar |
| Document Version | 0.3 |
| Last Revised Date | 13-05-2023 |

**Document Control**

### Change Record:

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Comments** |
| 0.1 | 13 – May -  2023 | Sonali D. | Introduction & Architecture defined |
| 0.2 | 13 – May -  2023 | Sonali D. | Architecture & Architecture Description appended and updated |
| 0.3 | 20 – May -  2023 | Sonali D. | Test Cases defined and appended |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

### Reviews:

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Reviewer** | **Comments** |
| 0.2 | 20 – May -  2023 | Sonali D. | Document Content, Version Control and Test Cases to be added |

### Approval Status:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Review**  **Date** | **Reviewed By** | **Approved By** | **Comments** |
|  |  |  |  |  |

Contents

1. [Introduction 1](#_bookmark0)
   1. [What is Low-Level design document? 1](#_bookmark1)
   2. [Scope 1](#_bookmark2)
2. [Architecture 2](#_bookmark3)
3. [Architecture Description 3](#_bookmark4)
   1. [Data Description 3](#_bookmark5)
   2. [Web Scrapping 3](#_bookmark6)
   3. [Data Transformation 3](#_bookmark7)
   4. [Data Insertion into Database 3](#_bookmark8)
   5. [Export Data from Database 3](#_bookmark9)
   6. [Data Pre-processing 3](#_bookmark10)
   7. [Data Clustering 3](#_bookmark11)
   8. [Model Building 4](#_bookmark12)
   9. [Data from User 4](#_bookmark13)
   10. [Data Validation 4](#_bookmark14)
   11. [User Data Inserting into Database 4](#_bookmark15)
   12. [Data Clustering 4](#_bookmark16)
   13. [Model Call for Specific Cluster 4](#_bookmark17)
   14. [Recipe Recommendation & Saving Output in Database 4](#_bookmark18)
4. [Test Cases 5](#_bookmark20)

# Introduction

## 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.

## Scope

Low-level design (LLD) is a component-level design process that follows a step-by-

step [refinement](https://en.wikipedia.org/wiki/Refinement_(computing)) 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

# Architecture

# Architecture Description

## Data Description

This dataset contains 4888 recipes including 20 different ingredients.

## Data Transformation

In the Transformation Process, we will convert our original dataset which is in JSON format to CSV Format.

## Data Insertion into Database

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

## Export Data from Database

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

## 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 with standard deviation zero or below a threshold, etc.

## 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 different algorithms 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

## 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 scores for 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.

## Data from User

Here we will collect physiological data from user such as user age, occupation, gender, product pitched, preferred property star, marital status, passport, own car, designation, monthly income as well as information directly provided by the user such as daily food intake

## Data Validation

Here Data Validation will be done, given by the user

## 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.

## Data Clustering

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

## Model Call for Specific Cluster

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

## Recipe Recommendation & Saving Output in Database

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

# Test Cases

|  |
| --- |
| **Test Case Description** |
| The model predicted output shape is proper or not |
| Test dataset leakage i.e. checking whether the data in training and testing datasets have no duplication |
| Temporal data leakage which involves checking whether the dependencies between training and test data do not lead to unrealistic situations in the time domain like training on a future data point and testing on a past data point |
| Check for the output ranges. In the cases where we are predicting outputs in a certain range (for example when predicting probabilities), we need to ensure the final prediction is not outside the expected range of values. |
| Ensuring a gradient step training on a batch of data leads to a decrease in the loss |
| Invariance tests which involve testing the model by tweaking only one feature in a data point and checking for consistency in model predictions. For example, if we are working with a travel prediction dataset then change in marital status should not affect an individual’s eligibility for the travel. |
| Directional expectations wherein we test for a direct relation between feature values and predictions. For example, in the case of a travel prediction problem, having a higher salary should definitely increase a person’s eligibility for a travel. |