Low-Level Design

Market Customer Segmentation

|  |  |
| --- | --- |
| Written By | Parekh Kathan Chiragkumar |
| Version | 1.0 |
| Date | 05-06-2003 |

**Document Change Control Record**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Comments** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

### Reviews:

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

### Approval Status:

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

**Index**

|  |  |
| --- | --- |
| **Content** | **Page No** |
| 1. Introduction | 2 |
| 1.1 What is Low-Level Design Document | 2 |
| 1.2 Scope | 2 |
| Architecture | 3 |
| 2. Architecture Description | 4 |
| 2.1 Data Description | 4 |
| 2.2 Data Gathering | 4 |
| 2.3 Raw Data Validation | 4 |
| 2.4 Model Building | 5 |
| 2.5 Model Saving | 5 |
| 2.6 Github | 5 |
| 3. Unit Test Cases | 6 |

**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 **‘Stores Sales Prediction’**. 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](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 :**

**2. Architecture Description**

**2.1 Data Description**

Given is the variable name, variable type, the measurement unit, and a brief description. The concrete compressive strength is the regression problem. The order of this listing corresponds to the order of numerals along the rows of the database.

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Measurement |
| Item\_Identifier | String | Unique product ID |
| Item\_Weight | Float | Weight of product |
| Item\_Fat\_Content | String | Whether the product is low fat or not |
| Item\_Visibility | Float | The % of a total display area of all products in a store allocated to the particular product |
| Item\_Type | String | The category to which the product belongs |
| Item\_MRP | Float | Maximum Retail Price (list price) of the product |
| Outlet\_Identifier | String | Unique store ID |
| Outlet\_Establishment\_Year | Integer | The year in which the store was established |
| Outlet\_Size | String | The size of the store in terms of ground area covered |
| Outlet\_Location\_Type | String | The type of city in which the store is located |
| Outlet\_Type | String | Whether the outlet is just a grocery store or some sort of supermarket |
| Item\_Outlet\_Sales | Float | Sales of the product in the particular store. This is the outcome variable to be predicted. |

**2.2 Data Gathering**

Data source: <https://www.kaggle.com/datasets/vjchoudhary7/customer-segmentation-tutorial-in-python>

Train and Test data are stored in .csv format.

**2.3 Raw Data Validation**

Raw data validation is a crucial step in customer segmentation using machine learning. It involves assessing the quality, accuracy, completeness, and consistency of the dataset. This validation ensures that the data is reliable, representative, and free from errors or inconsistencies. Key considerations include data quality, accuracy, completeness, consistency, formatting, privacy, security, and sample size. By conducting thorough validation, potential issues can be identified and addressed, leading to improved reliability and accuracy in the customer segmentation analysis.

**2.4 Model Building**

1. Preprocess the dataset by handling missing values, removing outliers, and scaling variables.
2. Determine the number of clusters, K, using techniques like the elbow method or silhouette analysis.
3. Initialize K cluster centroids either randomly or using the K-means++ method.
4. Assign each data point to the nearest centroid based on the Euclidean distance.
5. Recalculate the centroids by taking the mean of the data points within each cluster.
6. Repeat steps 4 and 5 until convergence or a maximum number of iterations is reached.
7. Evaluate the clustering results using metrics like the within-cluster sum of squares (WCSS) or silhouette coefficient.
8. Optionally, perform post-processing steps such as interpreting and labeling the clusters based on the characteristics of the data points within each cluster.
9. Use the trained K-means model to predict the cluster assignments of new data points.
10. Validate and refine the model by iteratively adjusting the number of clusters or exploring alternative clustering algorithms if needed.

**2.5 Model Saving**

Model is saved using joblib

extracted by the model to predict the prediction of sales, this is performed in this stage.

**2.6 GitHub**

The whole project directory will be pushed into the GitHub repository.

**3. Unit Test Cases.**

| **Test Case** | **Description** | **Input** | **Expected Output** |
| --- | --- | --- | --- |
| 1 | Empty Dataset | Empty dataset | Error or warning message indicating empty dataset |
| 2 | Minimum Required Dataset | Dataset with minimum required fields | Segmented customer groups based on available attributes |
| 3 | Dataset with Missing Values | Dataset with missing values | Cleaned dataset with missing values handled appropriately for segmentation |
| 4 | Large Dataset | Large dataset with a significant number of records | Accurate segmentation results within a reasonable time |
| 5 | Known Segment Characteristics | Dataset with pre-labeled customer segments | Correct assignment of customers to corresponding segments |
| 6 | Varying Number of Segments | Dataset with different numbers of desired segments | Segmentation into specified number of distinct and meaningful groups |
| 7 | Random Data Shuffling | Shuffled version of the dataset | Consistent segmentation results regardless of data order |
| 8 | Performance Metrics Evaluation | Segmented customer groups and ground truth labels | Calculation of relevant performance metrics (e.g., WCSS, silhouette coefficient) to evaluate segmentation quality |