Low Level Design
Amazon Food Sales Data Analysis

Revision Number: 1.0
Last date of revision: 1/03/2023

Amit Patil

iNeuron

Document Version Control

Date Issued	Version	Description	Author
1 March 2023	1.0	First Version of Complete LLD	Amit Patil

Contents

1. Introduction

1.1 What is Low-Level design document?

The goal of the LDD or Low-level design document (LLDD) is to give the internal logic design of the actual program code for the House Price Prediction dashboard. LDD describes the class diagrams with the methods and relations between classes and programs 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. The 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

The architecture of the project consists of the following components:

- Data Description
- Data Transformation
- Visualization

3. ARCHITECTURE DESCRIPTION

3.1 Data Description

1. Region: The region where the sales took place, represented as a string or object data type.
2.Country: The country where the sales took place, represented as a string or object data type.
3.Item Type: The type of item sold, represented as a string or object data type.
4.Sales Channel: The channel through which the sales were made, represented as a string or object data type.
5.Order Priority: The priority of the order, represented as a string or object data type.
6.Order Date: The date the order was made, represented as a string or object data type.
7.Order ID: A unique identifier for each order, represented as an integer data type.
8.Ship Date: The date the order was shipped, represented as a string or object data type.

9.Units Sold: The number of units sold, represented as an integer data type.

10.Unit Price: The price of each unit, represented as a float data type.

11.Unit Cost: The cost of producing each unit, represented as a float data type.

12.Total Revenue: The total revenue generated by the sale, represented as a float data type.

13.Total Cost: The total cost of producing and shipping the units, represented as a float data type.

14.Total Profit: The total profit generated by the sale, represented as a float data type.

3.2 Data Loading

The data will be cleaned, transformed, and normalized to prepare it for visualization. The data transformation process will involve removing duplicates, handling missing values, and converting data types where necessary

3.3 Data Visualization

The transformed data will be visualized using Matplotlib, a Python plotting library. The following charts and graphs will be created:

1. Bar chart for sales revenue, cost, and profit
2. Stacked bar chart for sales by region
3.Pie chart for sales by sales channel
4.Line chart for sales trends over time
5.Horizontal bar chart for top-selling products and regions
6.Bubble chart for customer demographics and behavior
7.Each chart will be labeled with appropriate titles and axes, and legend will be added where necessary.

4. UNIT TEST CASES

The following unit test cases will be used to ensure the quality of the code:

- Data loading test: This test will ensure that the provided datasets is loaded correctly.
- Data transformation test: This test will ensure that the data is transformed correctly and all required fields are present.
- Chart generation test: This test will ensure that the charts and graphs are generated correctly and display the data accurately.

That's it! I hope this format helps you create a low-level design document for your project. Let me know if you have any further questions or if you need any more assistance.