High Level Design (HLD)

Amazon Sales Data

Analysis

Revision Number: 1.0
Last date of revision: 01/01/2025

Utkarsh Sharma

Document Version Control

Date Issued	Version	Description	Author
01st January 2025	1.0	First Version of Complete HLD	Utkarsh Sharma

Contents

Abstract

This project involves a comprehensive analysis of Amazon's sales data. The primary objective is to gain insights into sales trends, customer preferences, and overall performance of different products across various categories on Amazon. The analysis will leverage historical sales data to identify patterns, forecast future sales, and provide actionable recommendations for improving sales strategies**.**

1.Introduction

1.1 Why this High-Level Design Document?

The purpose of this High-Level Design (HLD) Document is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding and can be used as a reference manual for how the modules interact at a high level.

The HLD will:

- Present all of the design aspects and define them in detail
- Describe the user interface being implemented
- Describe the hardware and software interfaces
- Describe the performance requirements
- Include design features and the architecture of the project
- List and describe the non-functional attributes like:
  - Security
  - Reliability
  - Maintainability
  - Portability
  - Reusability
  - Application compatibility
  - Resource utilization
  - Serviceability

1.2 Scope

The HLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HLD uses non-technical to mildly technical terms which should be understandable to the administrators of the system.

2.General Description

2.1 Problem Statement

Sales management has gained importance to meet increasing competition and the need for improved methods of distribution to reduce costs and to increase profits. Sales management today is the most important function in a commercial and business enterprise.

ETL process. Do ETL: Extract-Transform-Load some Amazon dataset and find for me Sales-trend-> month wise, year wise, yearly month wise Find key metrics and factors and show the meaningful relationships between attributes. Do your own research and come up with your findings.

2.2 Tools used

Business Intelligence tools such as Power BI and Canva are used to build the whole framework.
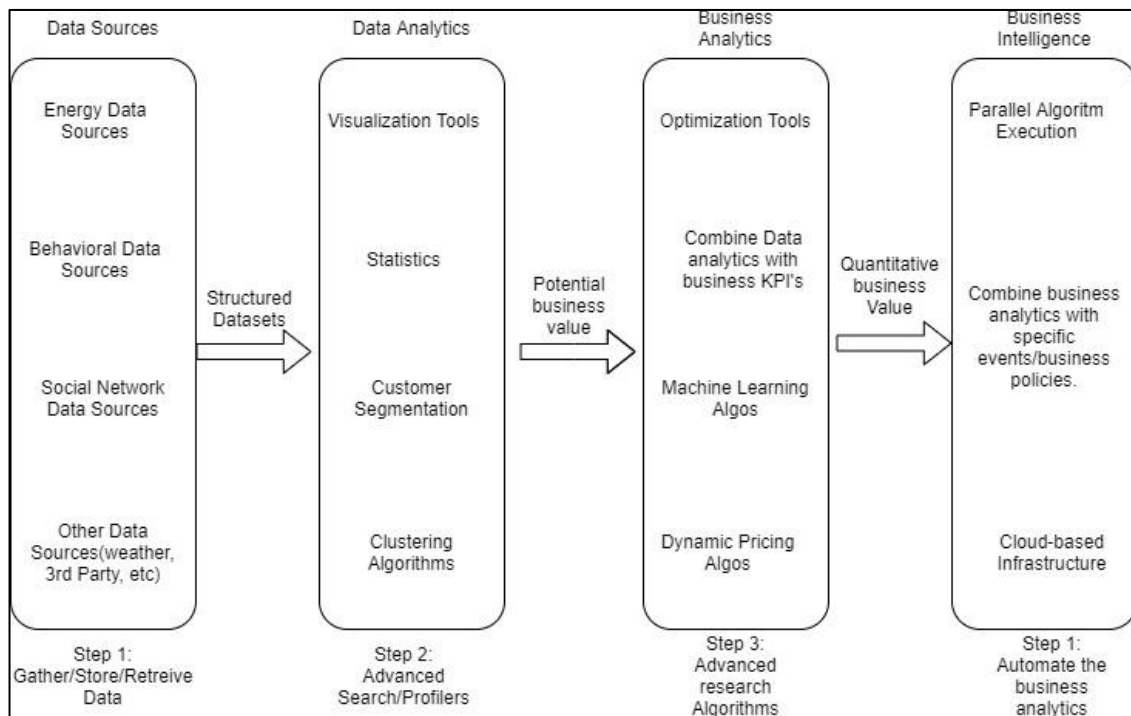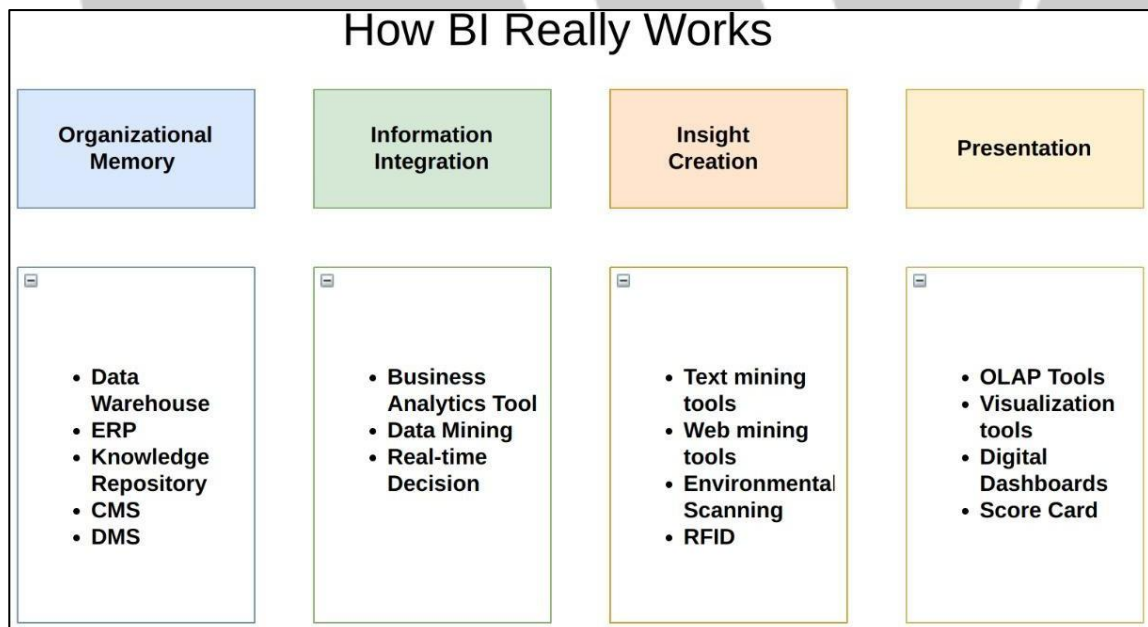
3. Design Details

3.1 Functional Architecture

Figure 1: Functional Architecture of Business Intelligence

3.2 Optimization

Your data strategy drives performance

- Minimize the number of fields
- Minimize the number of records
- Optimize extracts to speed up future queries by materializing calculations, removing columns and the use of accelerated views

Reduce the marks (data points) in your view

- Practice guided analytics. There's no need to fit everything you plan to show in a single view. Compile related views and connect them with action filters to travel from overview to highly granular views at the speed of thought.
- Remove unneeded dimensions from the detail shelf.
- Explore. Try displaying your data in different types of views.

Limit your filters by number and type

- Reduce the number of filters in use. Excessive filters on a view will create a more complex query, which takes longer to return results. Double-check your filters and remove any that aren't necessary.

- Use an include filter. Exclude filters load the entire domain of a dimension, while include filters do not. An include filter runs much faster than an exclude filter, especially for dimensions with many members.

- Use a continuous date filter. Continuous date filters (relative and range-of-date filters) can take advantage of the indexing properties in your database and are faster than discrete date filters.

- Use Boolean or numeric filters. Computers process integers and Booleans (t/f) much faster than strings.

- Use parameters and action filters. These reduce the query load (and work across data sources).

Optimize and materialize your calculations

- Perform calculations in the database
- Reduce the number of nested calculations.

- Reduce the granularity of LOD or table calculations in the view. The more granular the calculation, the longer it takes.

  - LODs - Look at the number of unique dimension members in the calculation.

  - Table Calculations - the more marks in the view, the longer it will take to calculate.

- Where possible, use MIN or MAX instead of AVG. AVG requires more processing than MIN or MAX. Often rows will be duplicated and display the same result with MIN, MAX, or AVG.

- Make groups with calculations. Like include filters, calculated groups load only named members of the domain, whereas Tableau's group function loads the entire domain.

- Use Booleans or numeric calculations instead of string calculations. Computers can process integers and Booleans (t/f) much faster than strings. Boolean>Int>Float>Date>DateTime>String

4. KPIs

Dashboards will be implemented to display and indicate certain KPIs and relevant indicators for the disease.

As and when, the system starts to capture the historical/periodic data for a user, the dashboards will be included to display charts over time with progress on various indicators or factors

4.1 KPIs (Key Performance Indicators)

Key indicators displaying a summary of the Housing Price and its relationship with different metrics

1. Total Revenue generated by the company
2. Total units sold by the company
3. Monthly revenue generated by the company
4. Yearly revenue generated by the company
5. Comparison between the online and offline market
6. Region wise revenue generated

5. Deployment

1. Data Integration
- **Data Sources**: Connect Power BI to various data sources, including Amazon sales data stored in databases, Excel files, or cloud storage platforms.
- **Data Refresh**: Set up automatic data refresh schedules to ensure the dashboards display the most up-to-date information.

2. Data Modelling
- **Data Transformation**: Use Power Query Editor to clean, transform, and structure the data for analysis.
- **Relationships**: Define relationships between different data tables to enable comprehensive analysis and accurate reporting.

3. Dashboard Creation
- **Visualizations**: Create a variety of visualizations such as bar charts, line graphs, pie charts, and heat maps to represent sales data effectively.
- **Interactive Elements**: Incorporate slicers, filters, and drill-through options to allow users to interact with the data and gain deeper insights.

4. Deployment and Sharing
- **Publishing**: Publish the Power BI report to the Power BI Service, making it accessible to authorized users.
- **Workspaces**: Organize dashboards within workspaces for better collaboration and management.
- **Access Control**: Set up role-based access to ensure that only authorized personnel can view and interact with the dashboards.

5. Monitoring and Maintenance
- **Usage Monitoring**: Monitor usage metrics to understand how stakeholders are interacting with the dashboards and identify areas for improvement.
- **Maintenance**: Regularly update the data models, visualizations, and reports to keep them aligned with the evolving business needs.

6. User Training and Support
- **Training Sessions**: Conduct training sessions for end-users to familiarize them with the functionalities of the Power BI dashboards.
- **Documentation**: Provide comprehensive documentation and user guides to assist users in navigating and utilizing the dashboards effectively.
- **Support**: Establish a support system to address any technical issues or user queries.

This deployment plan ensures that the Power BI dashboards for Amazon Sales Analysis are effectively integrated, managed, and utilized to provide valuable insights and support data-driven decision-making.