High Level Design (HLD)

Analysing Amazon Sales data

Revision Number: 1.0
Last date of revision: 08/01/2023

Abhilash Amrutkar

Document Version Control

Date Issued	Version	Description	Author
8/1/2023	1.0	First Version of Complete HLD	Abhilash Amrutkar

Contents

Abstract

This project aims to identify trends and patterns in sales data in order to inform business strategy and improve performance. The analysis will be conducted using a combination of statistical and machine learning techniques, and will include the creation of interactive dashboards to visualize the results. Key findings will be presented to stakeholders in the form of reports and recommendations. By better understanding the drivers of sales and identifying opportunities for growth, we hope to increase revenue and improve the overall effectiveness of our sales efforts.

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 Product Perspective & Problem Statement

Sales management has gained importance to meet increasing competition and the need for improved methods of distribution to reduce cost and to increase profits. Sales management today is the most important function in a commercial and business enterprise. 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. Python, R, Tableau, Power BI or you can use any tools and techniques as per your convenience. We would appreciate your valid imagination in finding solutions

2.2 Tools used

Business Intelligence tools and libraries works such as Numpy, Pandas, Matplotlib, Excel, Power BI are used to build the whole framework.

3 Design Details

3.1 Functional Architecture

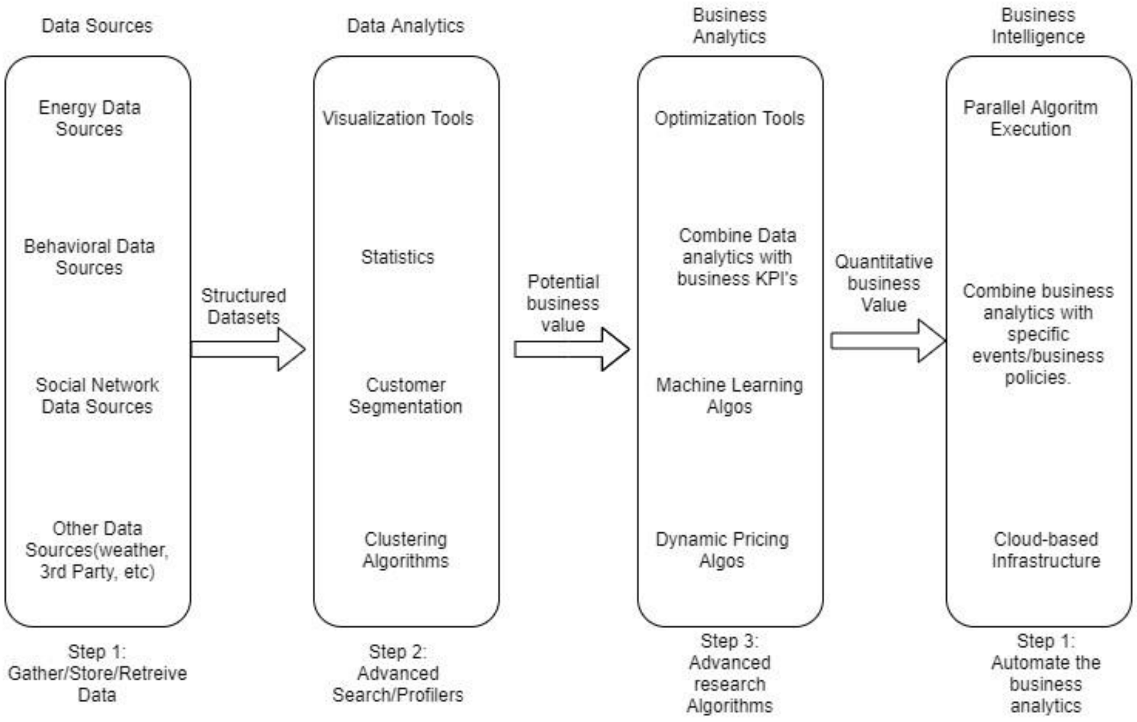
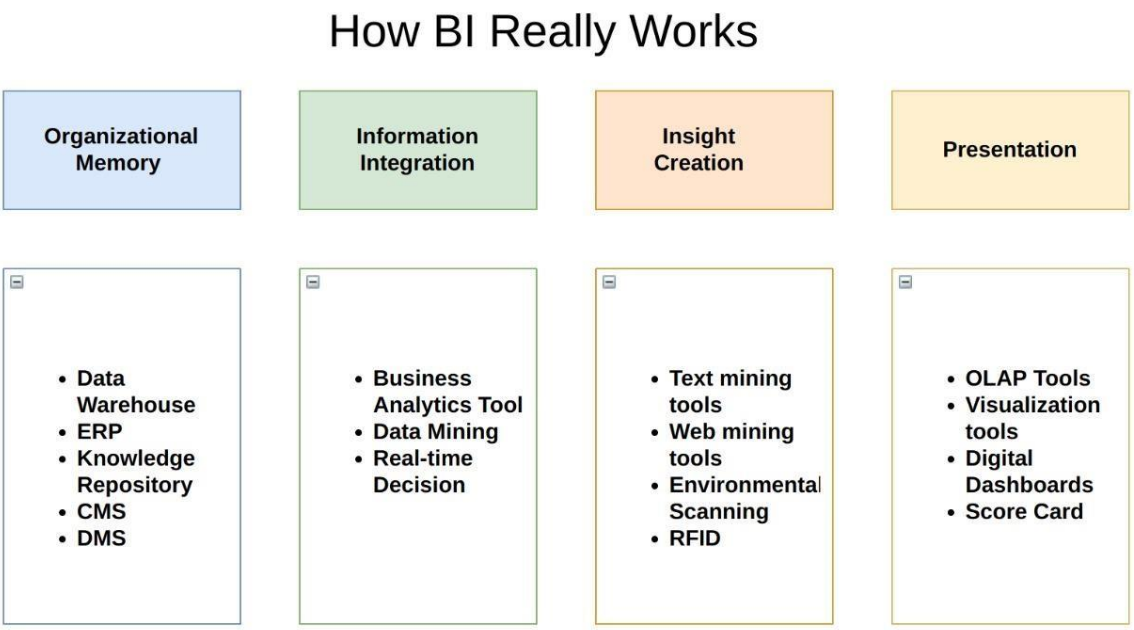

Data Sources
Energy Data Sources
Behavioral Data Sources
Social Network Data Sources
Other Data Sources(weather, 3rd Party, etc)
Step 1: Gather/Store/Retreive Data
Structured Datasets
Data Analytics
Visualization Tools
Statistics
Customer Segmentation
Clustering Algorithms
Step 2: Advanced Search/Profilers
Potential business value
Business Analytics
Optimization Tools
Combine Data analytics with business KPI's
Machine Learning Algos
Dynamic Pricing Algos
Step 3: Advanced research Algorithms
Quantitative business Value
Business Intelligence
Parallel Algoritm Execution
Combine business analytics with specific events/business policies.
Cloud-based Infrastructure
Step 1: Automate the business analytics

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)

- Revenue: The total amount of money earned through sales activities.
- Profitability: The amount of profit generated by the business, calculated as revenue minus expenses.
- Customer satisfaction: A measure of how satisfied customers are with the products or services they have received.
- Market share: The percentage of the total market that is being served by the business.
- Return on investment (ROI): The amount of profit generated relative to the amount of money invested in the business.
- Efficiency: A measure of how well the business is using its resources to produce goods or services.

5 Deployment

To publish a Power BI report or dashboard to the Power BI service, follow these steps:

1. Open the Power BI Desktop application and sign in with your Power BI account.
2. Load and prepare the data for your report or dashboard.
3. Create the report or dashboard using the Power BI visualizations and tools.
4. Save the report or dashboard to the Power BI service by clicking the "Publish" button in the Home tab.
5. Select the destination workspace in the Power BI service where you want to publish the report or dashboard.
6. Click "Publish" to upload the report or dashboard to the Power BI service.

Once the report or dashboard has been published, users can access it by logging into the Power BI service or by embedding it into another application or website using the Power BI API or embedding code.