High Level Design (HLD)

Analyzing Amazon Sales Data Analysis

Revision Number: 1.0
Last date of revision: 13/07/2023

Karan Singh

Document Version Control

Date issued	Version	Description	Author
13th Jul 2023	1.0	Full version of HLD Document	Karan Singh

Abstract

In the face of rising competition and the desire for improved distribution methods to reduce costs and increase profits, sales management has become a crucial function in commercial and business enterprises. To address this, the project focuses on performing ETL (Extract-Transform-Load) on an Amazon dataset to analyze sales trends. The objective is to uncover sales trends on a month-wise, year-wise, and yearly_month-wise basis, while also identifying key metrics and factors that contribute to these trends. Through independent research and analysis, meaningful relationships between attributes will be established, leading to insightful findings and recommendations for enhancing sales management practices and driving business success.

Contents

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

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

Product Perspective:

Our product aims to provide a comprehensive sales trend analysis solution by performing ETL (Extract-Transform-Load) on an Amazon dataset. By extracting sales data, transforming it into a suitable format, and loading it into a data analysis tool, we enable users to gain insights into sales trends on a month-wise, year-wise, and yearly_month-wise basis. The product focuses on identifying key metrics and factors while showcasing meaningful relationships between attributes for effective sales management.

Problem Statement:

Sales management has become increasingly crucial due to rising competition and the need for improved distribution methods to reduce costs and increase profitability. In today's commercial and business enterprises, sales management stands as the most important function. However, there is a lack of a comprehensive solution that analyzes sales trends in a granular manner. This poses challenges for businesses in understanding the month-wise, year-wise, and yearly_month-wise sales trends, and their relationships with key metrics and factors. Therefore, there is a need for a product that performs ETL on an Amazon dataset, discovers the meaningful relationships between attributes, and presents insightful findings to enhance sales management strategies and decision-making processes.

2.2 Tools Used

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

3. Design details

3.1 Functional Architecture

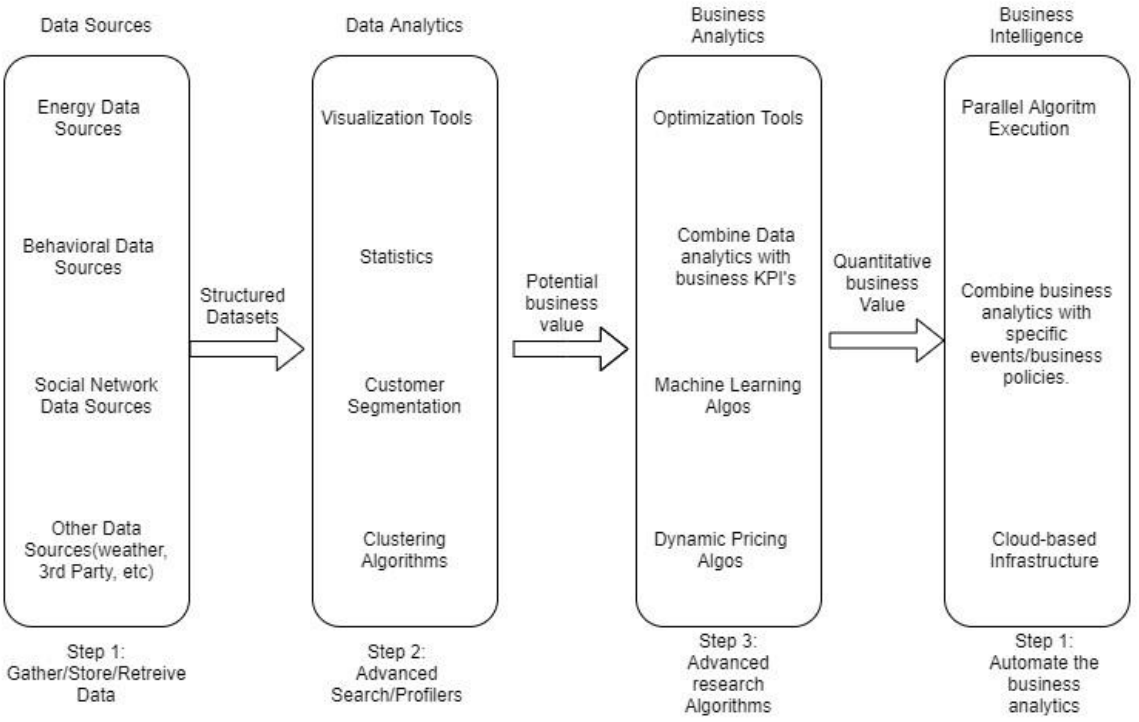

Figure 1: Functional Architecture of Business Intelligence

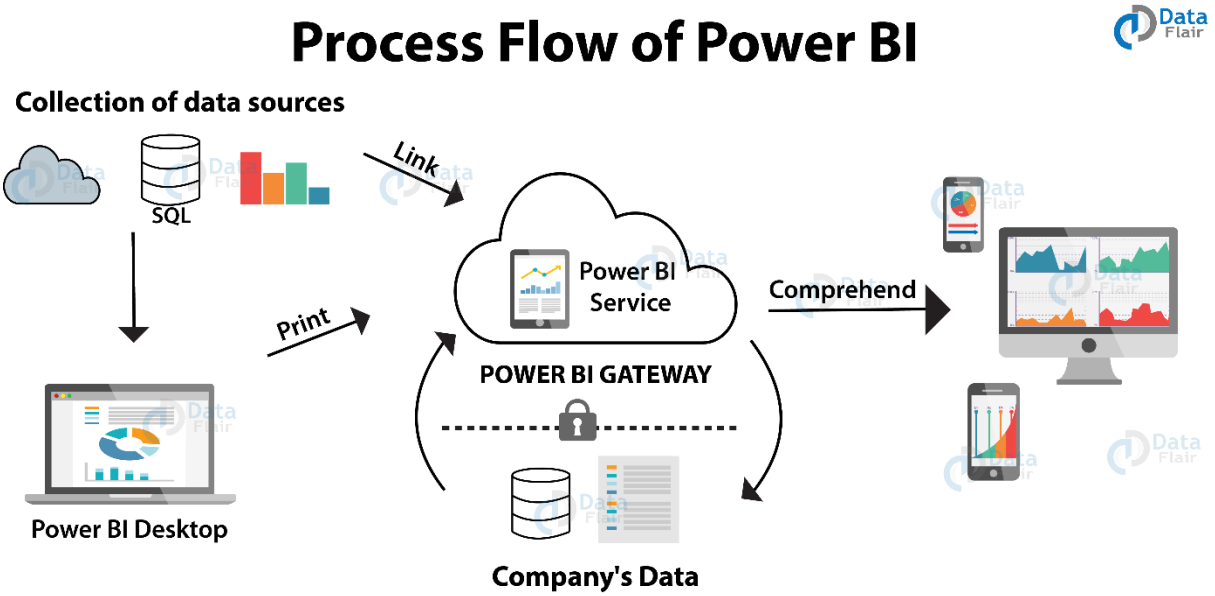

Figure 2: Process Flow Diagram of Power BI

3.2 Optimization

1. **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 view

2. 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.

3. 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).

4. 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.
      - o LODs - Look at the number of unique dimension members in the calculation.
      - o 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.

4. KPIs(Key Performance Indicators)

1. Monthly Sales Revenue: Measure the total sales revenue generated on a monthly basis to track the financial performance of the sales department.
2. Yearly Sales Growth Rate: Calculate the percentage increase in sales revenue from the previous year to assess the overall sales growth and business performance.
3. Monthly Sales Quantity: Monitor the total quantity of products sold on a monthly basis to gauge the sales volume and identify trends in customer demand.
4. Average Order Value: Determine the average value of each sales order to understand customer purchasing patterns and identify opportunities to increase order size.
5. Sales Conversion Rate: Calculate the percentage of leads or prospects that convert into actual sales to evaluate the effectiveness of the sales team in closing deals.
6. Customer Acquisition Cost (CAC): Measure the cost incurred to acquire a new customer, including marketing and sales expenses, to assess the efficiency of customer acquisition efforts.
7. Customer Retention Rate: Determine the percentage of existing customers retained over a specific period to evaluate the effectiveness of customer retention strategies.
8. Sales Pipeline Value: Track the total value of potential sales opportunities in the sales pipeline to assess the potential revenue growth and identify areas for improvement in the sales process.
9. Sales Cycle Length: Measure the average time it takes for a sales opportunity to move through the sales cycle, from initial contact to closed deal, to identify bottlenecks and optimize the sales process.
10. Customer Satisfaction Score (CSAT): Collect feedback from customers to assess their level of satisfaction with the sales process, products, and services provided.