High Level Design (HLD)

Analyzing Amazon Sales Data

Revision Number: 1.0
Last date of revision: 14/04/2023

Mohammad
Aadil

Document Version Control

Date Issued	Version	Description	Author
14th Apr 2023	1.0	First Version of Complete HLD	Mohammad Aadil

Contents

Abstract

Sales management is a critical function in today's business world, as it helps organizations meet increasing competition, reduce costs, and improve profits. In this project, we aim to perform Extract-Transform-Load (ETL) on an Amazon dataset and analyze sales trends month-wise, year-wise, and yearly-month-wise. We will also identify key metrics and factors that affect sales and explore the relationships between attributes to gain insights into how businesses can improve their sales performance.

Our project will involve accessing the Amazon dataset and using ETL techniques to extract, transform, and load the data into a data warehousing system. We will then perform sales trend analysis to identify trends based on different time periods, including month-wise, year-wise, and yearly-month-wise. This analysis will help us identify seasonal variations, forecast future sales, and gain insights into how to improve sales performance.

We will also identify key metrics and factors that influence sales, including customer demographics, product categories, pricing, and promotions. Through data analysis, we will explore the relationships between these factors and gain insights into how they impact sales. This analysis will help businesses make informed decisions and develop strategies to improve their sales performance.

Our findings will provide valuable insights into sales trends and factors that affect sales, enabling businesses to make data-driven decisions and improve their sales performance. By leveraging our insights, businesses can develop effective sales strategies, increase profitability, and maintain their competitive edge in today's business environment.

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.

2.2 Tools used

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

2.3 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

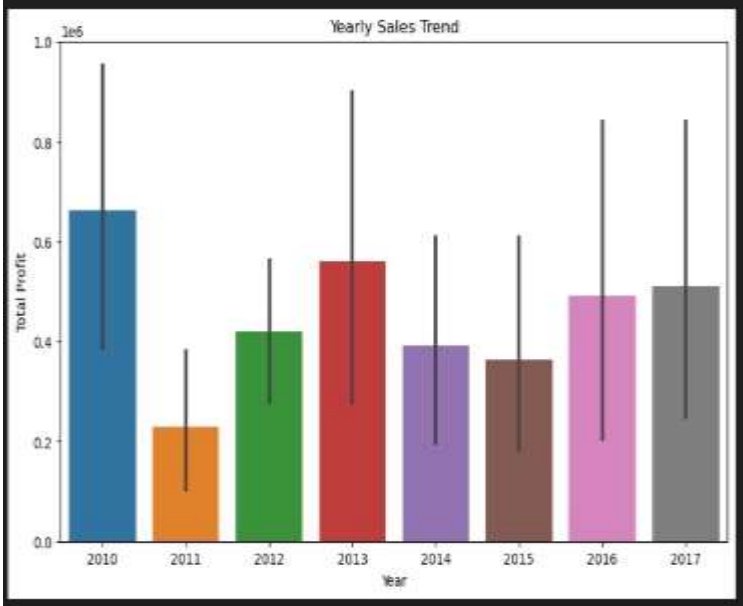
- Perform calculations in the database
- Reduce the number of nested calculations.
- Reduce the granularity of LOD or table calculations in the view. The more granular the calculation, the longer it takes.
    - LODs - Look at the number of unique dimension members in the calculation.
    - Table Calculations - 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

```
# Yearly sales plot

plt.figure(figsize=(10, 6))
sns.barplot(data=sales_data, x='Year', y='Total Profit')
plt.title('Yearly Sales Trend')
plt.xlabel('Year')
plt.ylabel('Total Profit')
```

```
Text(0, 0.5, 'Total Profit')
```



```
# Monthly sales plot
plt.figure(figsize=(10, 6))
sns.lineplot(data=sales_data, x='Month', y='Total Profit', hue='Year')
plt.title('Monthly Sales Trend')
plt.xlabel('Month')
plt.ylabel('Total Profit')
```

```
Text(0, 0.5, 'Total Profit')
```