High Level Design (HLD)
Wine Data Analysis

Revision Number: 1.0
Last date of revision: 10/09/2021

Atul kumar singh

Document Version Control

Date Issued	Version	Description	Author
01th June 2021	1.0	First Version of Complete HLD	Atul kumar singh

Contents

Abstract

Wine Data Analysis are an important reflection of the economy, and wine ranges are of great interest for both buyers and sellers. Ask a home buyer to describe their dream house, and they probably won't begin with the height of the basement ceiling or the proximity to an country. But this playground competition's data-set proves that much more influences price negotiations than the number of bedrooms or a white-picket fence.

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

Housing prices are an important reflection of the economy, and housing price ranges are of great interest for both buyers and sellers. In this project, house prices will be predicted given explanatory variables that cover many aspects of residential houses. The objective of the project is to perform data visualization techniques to understand the insight of the data. This project aims apply various Business Intelligence tools such as Tableau or Power BI to get a visual understanding of the data.

2.2 Tools used

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

3. Design Details

3.1 Functional Architecture

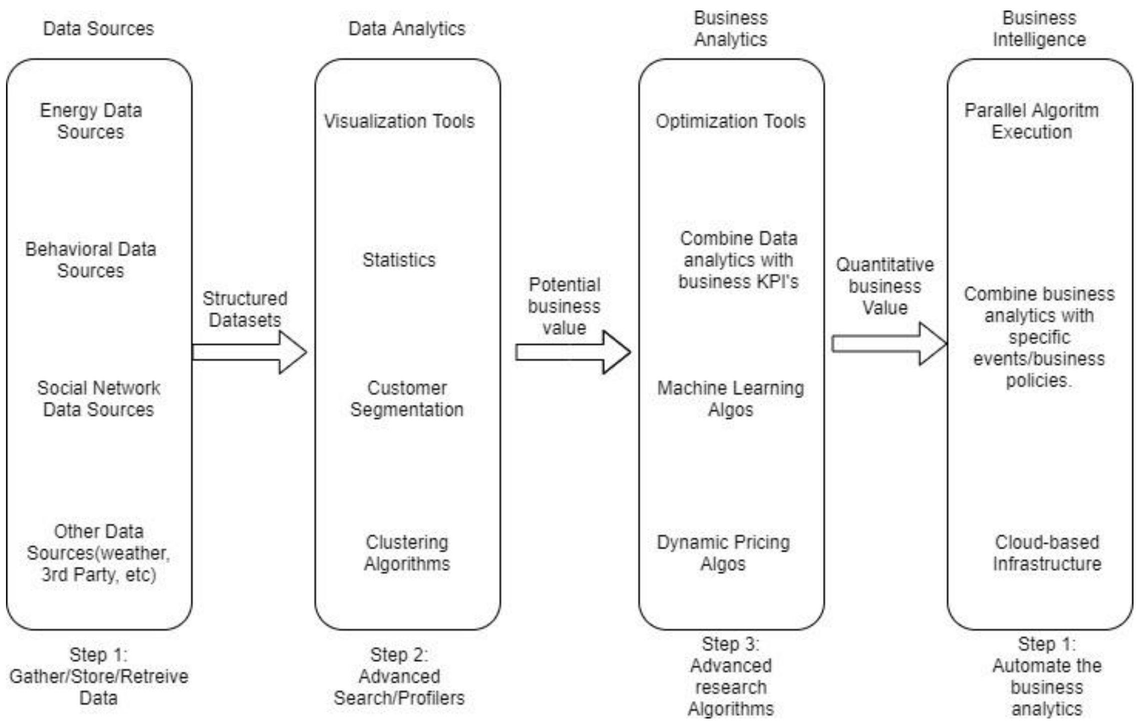
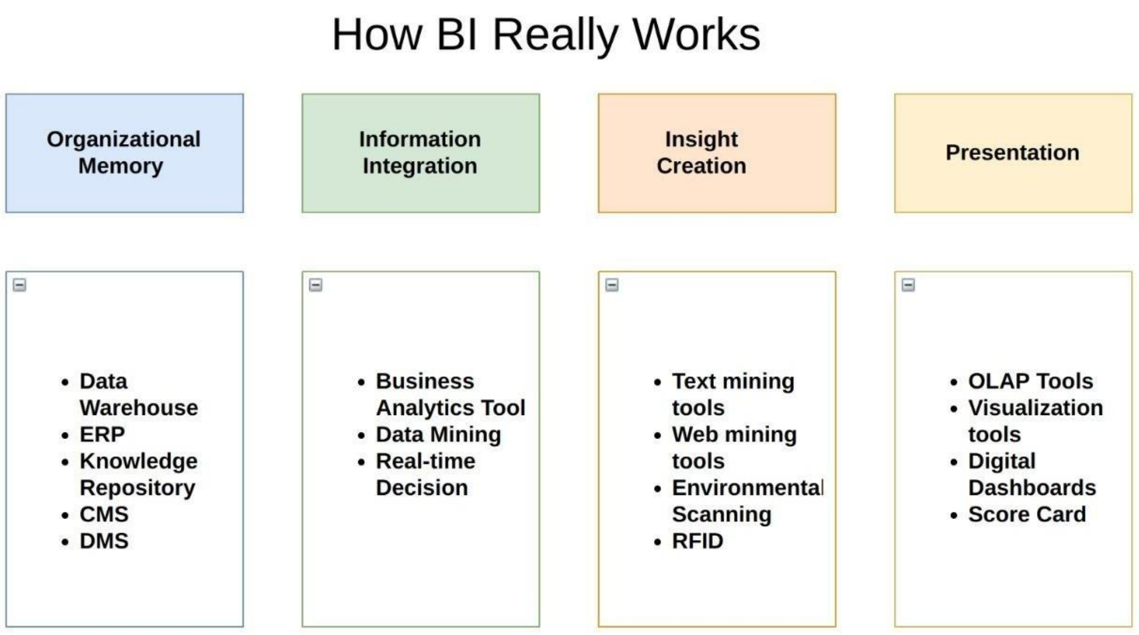

Data Sources
Energy Data Sources
Behavioral Data Sources
Social Network Data Sources
Other Data Sources(weather, 3rd Party, etc)
Step 1: Gather/Store/Retreive Data
Structured Datasets
Data Analytics
Visualization Tools
Statistics
Customer Segmentation
Clustering Algorithms
Step 2: Advanced Search/Profilers
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.

Deployment

Prioritizing data and analytics couldn’t come at a better time. Your company, no matter what size, is already collecting data and most likely analyzing just a portion of it to solve business problems, gain competitive advantages, and drive enterprise transformation. With the explosive growth of enterprise data, database technologies, and the high demand for analytical skills, today’s most effective IT organizations have shifted their focus to enabling self-service by deploying and operating python at scale, as well as organizing, orchestrating, and unifying disparate sources of data for business users and experts alike to author and consume content.

python prioritizes choice in flexibility to fit, rather than dictate, your enterprise architecture. Plot leverage your existing technology investments and integrate into your IT infrastructure to provide a self-service, modern analytics platform for your users.