High Level Design (HLD)

Analysing world's best wine reviews

Revision Number: 1.0
Last date of revision: 09/09/2024

Saurav Kashyap

Document Version Control

Date Issued	Version	Description	Author
9th sept. 2024	1.0	First Version of Complete HLD	Saurav Kashyap

Contents

Abstract

The findings of this study will involve scrutinizing the best wine reviews from around the world in a bid to identifying the aspects that define wine quality as well as consumer preferences.

In this paper, the use of data analysis and data mining methods important attributes of wines including the region origin, alcohol percentage for wines, their rating and the price are analysed to compare and establish trends.

The knowledge provided by our research will be beneficial for the wine consumers, producers, retailers, and collectors to make better choices and find the best wines in the crowded and challenging wine market.

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

From a product perspective, this wine dataset can be used to:

Develop a wine recommendation system: By understanding the relationships between wine attributes (varietal, country, appellation, etc.) and ratings, a system can recommend wines that match a user's preferences.

Optimize wine pricing: Analyze the relationship between wine attributes and price to determine optimal pricing strategies.

Identify market trends: Analyze trends in wine consumption, preferences, and regional variations.
Improve wine production: Use the data to identify factors that contribute to high-quality wines and guide winemakers in their practices.

Problem Statement

Based on this dataset, potential problem statements could be:

Predicting Wine Quality:

- Given the attributes of a wine, can we accurately predict its quality rating?
- Which features have the most significant impact on wine quality?

Understanding Consumer Preferences:

- What factors influence consumer preferences for wine (e.g., varietal, country, price)?
- How do consumer preferences vary across different regions or demographics?

Optimizing Wine Pricing:

- What is the optimal pricing strategy for wines based on their attributes?
- How can we maximize revenue while maintaining customer satisfaction?

Identifying Emerging Trends:

- What are the latest trends in wine consumption and preferences?
- Which wine regions or varietals are gaining popularity?

2.2 Tools used

Data Manipulation and Analysis:

- **NumPy (np):** Provides efficient numerical computations for arrays.
- **Pandas (pd):** Enables data manipulation and analysis through Data Frames and Series.
- **Matplotlib.pyplot (plt):** Offers a versatile toolkit for creating static visualizations.
- **Seaborn (sns):** Builds on Matplotlib for creating higher-level statistical graphics.

Statistical Modelling:

- **Stats models:** Provides a library for various statistical models.

Text Analysis (if applicable):

- **NLTK:** Facilitates natural language processing tasks (potentially for analysing wine descriptions, if available).

Machine Learning:

- **Scikit-learn (sklearn):** The go-to library for machine learning algorithms and tools.
  - **PCA (from sklearn.decomposition):** Performs Principal Component Analysis for dimensionality reduction.
  - **train_test_split (from sklearn.model_selection):** Splits data into training and testing sets.

- **cross_val_score (from sklearn.model_selection):** Performs cross-validation for model evaluation.
- **RandomForestClassifier,AdaBoostClassifier,SVC(from sklearn.ensemble and sklearn.svm):** Implement popular classification algorithms (Random Forest, AdaBoost, Support Vector Machines).
- **Pipeline (from sklearn.pipeline):** Creates a pipeline to chain data preprocessing and modeling steps.
- **StandardScaler (from sklearn.preprocessing):** Standardizes features for better model performance.
- **f1_score (from sklearn.metrics):** Calculates the F1 score (harmonic mean of precision and recall) for classification evaluation.
- **confusion_matrix (from sklearn.metrics):** Generates a confusion matrix for evaluating model performance.

Interactive Data Visualization :

- **Plotly.express (ex):** Creates concise interactive visualizations.
- **Plotly.graph_objs (go):** Offers building blocks for interactive figures.
- **Plotly.figure_factory (ff):** Provides specialized visualization functions.
- **Plotly.subplots (from plotly.subplots):** Allows for creating subplots in a figure.
- **Plotly.offline (pyo):** Enables offline rendering of interactive plots in Jupyter Notebook.

Imbalanced Class Handling :

- **Imbalanced-learn:** Provides tools for addressing imbalanced class distributions (useful if some wine quality ratings are rare).
    - **SMOTE (from imblearn.over_sampling):** Oversamples data points from minority classes to balance the dataset.

3 Design Details

3.1 Functional Architecture

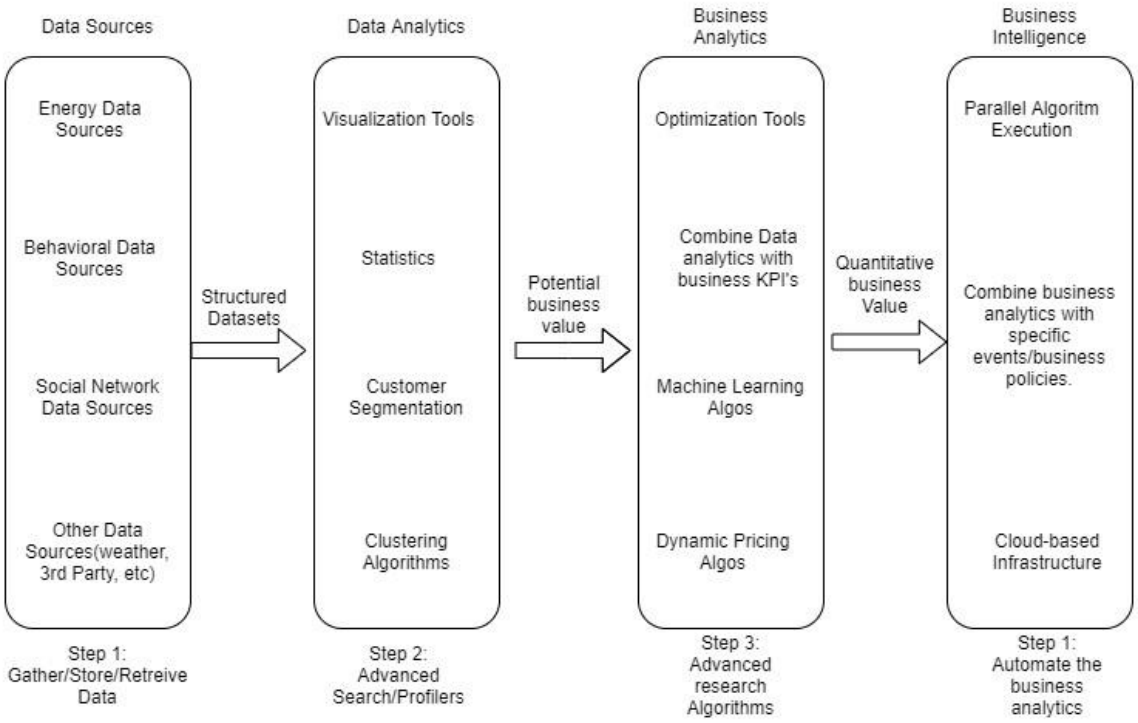
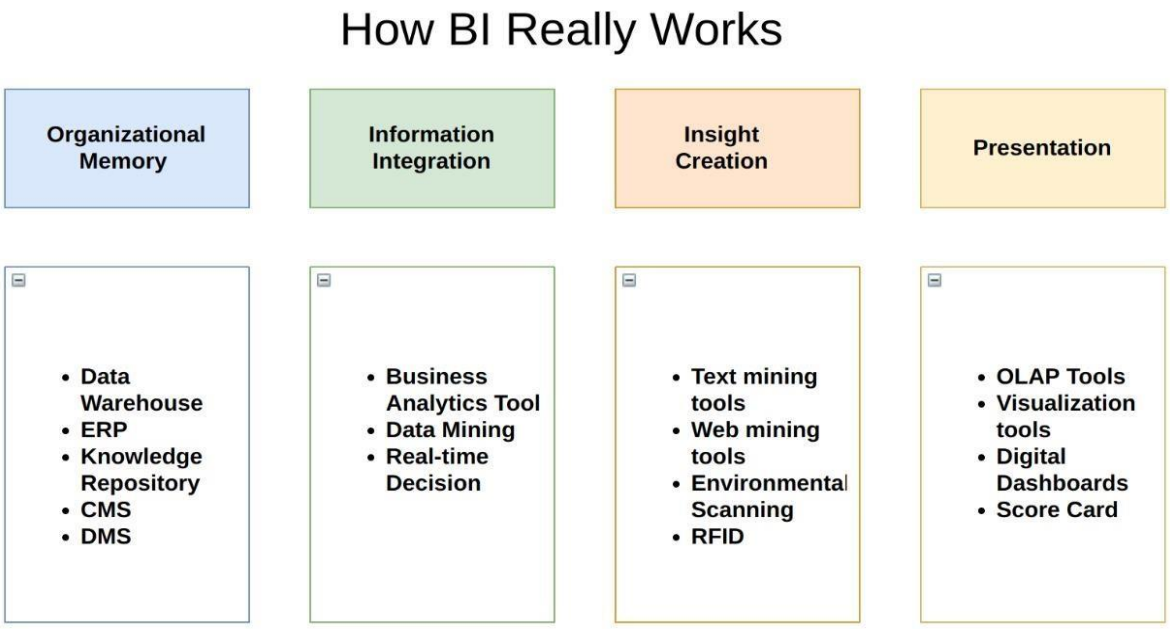


Figure 1: Functional Architecture of Business Intelligence



3.2 Optimization

Feature Engineering:

- **Interaction Terms:** Create new features by combining existing ones (e.g., alcohol_per_price).
- **One-Hot Encoding:** Convert categorical variables (like country, category) into numerical representations.
- **Normalization or Standardization:** Scale numerical features to a common range for better model convergence.

Model Selection and Tuning:

- **Experiment with Different Models:** Try various regression models (e.g., linear regression, random forest, gradient boosting) to find the best fit.
- **Hyperparameter Tuning:** Optimize model parameters using techniques like grid search, random search, or Bayesian optimization.
- **Ensemble Methods:** Combine multiple models (e.g., stacking, bagging) to improve performance.

Data Preprocessing:

- **Handle Missing Values:** Impute missing values using appropriate methods (e.g., mean, median, mode, imputation algorithms).
- **Outlier Detection and Removal:** Identify and handle outliers that can skew the model's performance.

Evaluation Metrics:

- **Choose Appropriate Metrics:** Select metrics relevant to the task (e.g., R-squared, mean squared error, mean absolute error for regression).
- **Cross-Validation:** Evaluate model performance on multiple folds of the data to avoid overfitting.

Feature Importance:

- **Identify Key Factors:** Determine which features have the most significant impact on wine quality.

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)

Overall Performance:

- **R-squared:** Measures the proportion of variance in the target variable (rating) explained by the model.
- **Mean Squared Error (MSE):** Quantifies the average squared difference between predicted and actual ratings.
- **Mean Absolute Error (MAE):** Measures the average absolute difference between predicted and actual ratings.

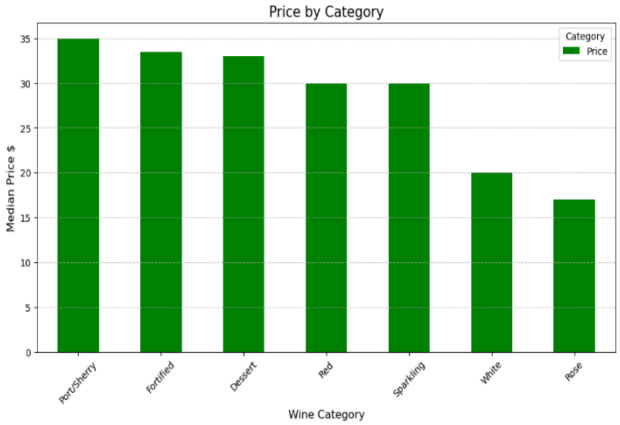
Specific Metrics:

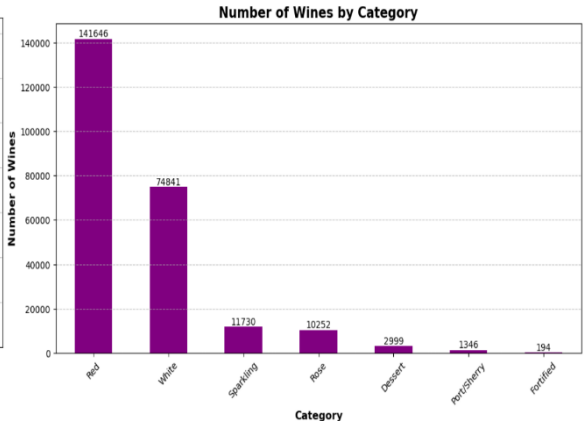
- **Accuracy:** For classification tasks (if applicable), measures the overall correctness of predictions.
- **Precision:** Measures the proportion of positive predictions that are actually positive.
- **Recall:** Measures the proportion of actual positive cases that are correctly predicted.
- **F1-score:** The harmonic mean of precision and recall, providing a balanced metric.
- **Confusion Matrix:** Visualizes the classification performance, including true positives, true negatives, false positives, and false negatives.

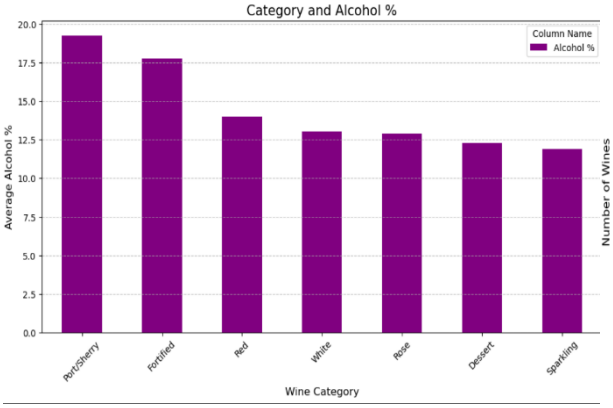
Feature Importance:

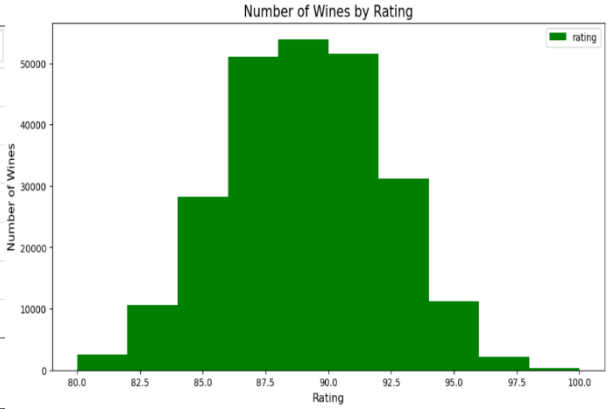
- **Feature Importance:** Identifies the most influential features (e.g., alcohol, price, varietal) in predicting wine quality.

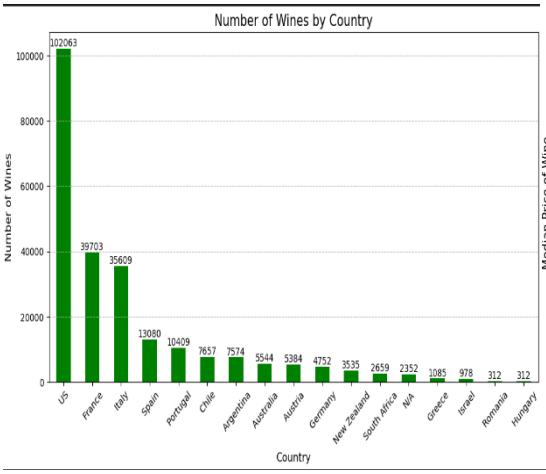
5 Dashboard-


Price by Category
Category
Price
Median Price $
Wine Category
Port/Sherry
Fortified
Dessert
Red
Sparkling
White
Rose

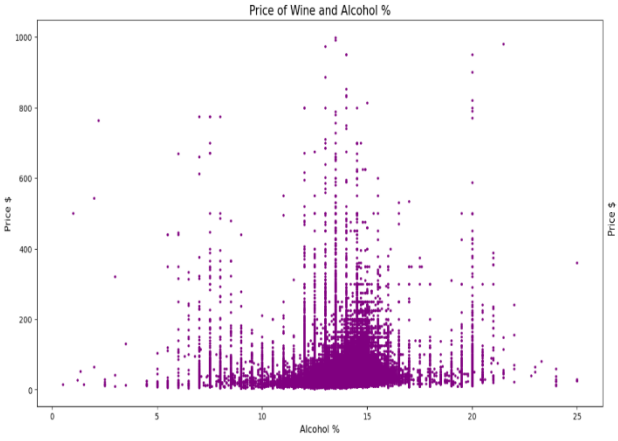

Number of Wines by Category
Number of Wines
Category
141646
74841
11730
10252
2999
1346
194
Red
White
Sparkling
Rose
Dessert
Port/Sherry
Fortified

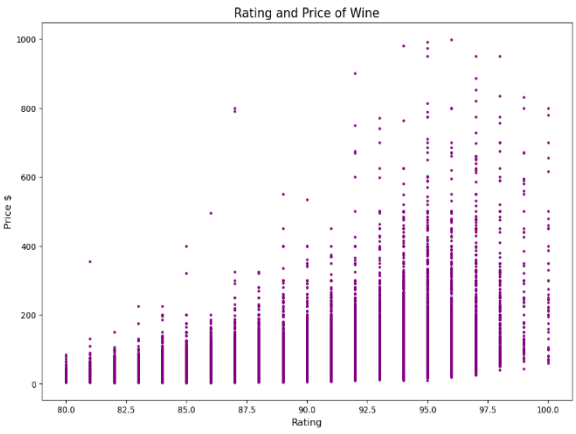

Category and Alcohol %
Column Name
Alcohol %
Average Alcohol %
Wine Category
Port/Sherry
Fortified
Red
White
Rose
Dessert
Sparkling


Number of Wines by Rating
rating
Number of Wines
Rating


Number of Wines by Country
Number of Wines
Country
102063
39703
35609
13080
10409
7657
7574
5544
5384
4752
3535
2659
2352
1085
978
312
312
US
France
Italy
Spain
Portugal
Chile
Argentina
Australia
Austria
Germany
New Zealand
South Africa
N/A
Greece
Israel
Romania
Hungary


Price of Wine by Country
Column Name
Price $
Median Price of Wine
Country
US
Italy
Israel
Austria
France
Hungary
Germany
New Zealand
N/A
Australia
Greece
Portugal
South Africa
Spain
Argentina
Chile
Romania


Price of Wine and Alcohol %
Price $
Alcohol %


Rating and Price of Wine
Price $
Rating