Heart Disease Diagnostic Analysis

HIGH LEVEL DESIGN (HLD)

Author: Shraddha Giradkar

Contents

Abstract:

Heart disease is type of disease that affects the human heart or blood vessels. Heart disease is a term covering any disorder of the heart. Basically Heart disease is a general term that means that the heart is not working normally. Heart diseases have become a major concern nowadays. Studies shows that the number of deaths due to heart failures have increased significantly over the past few decades in India. It has become the leading cause of death in India. A study shows that from 1990 to 2016 the death rate due to heart diseases have increased around 34% from 155.7 to 209.1 deaths per 1 lakh population in India. Thus, preventing heart diseases has become more than necessary. Good data driven systems for prediction of heart diseases can improve the research and prevention process, which makes sure that more people can live healthy lives.

1 Introduction:

1.1Why 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 before 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
- Reusability
-Application compatibility
-Resource utilization
-Serviceability
-Reliability
-Maintainability
-Portability

1.2Scope:

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:

The goal of this project is to analyse and to predict the probability of heart disease occurrence, based on a combination of features that describes the disease. To achieve this goal, we used a data set that is formed by taking into consideration some of the information of 303 individuals. The problem is based on the given information about each individual we have to calculate that whether that individual will suffer from heart disease or not.

2.2 Tools used:

MS-Power BI, NumPy, Pandas, Seaborn, Matplotlib, MS-Excel, Jupyter Notebook and Python Programming Language are used to build the whole framework.

3 Design Details:

3.1 Functional Architecture:

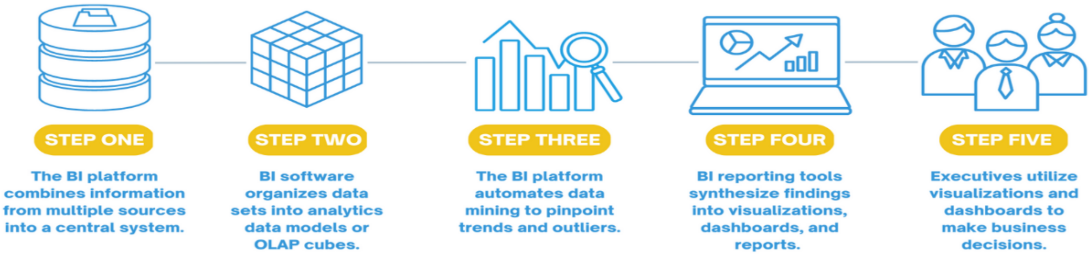
STEP ONE
The BI platform combines information from multiple sources into a central system.
STEP TWO
BI software organizes data sets into analytics data models or OLAP cubes.
STEP THREE
The BI platform automates data mining to pinpoint trends and outliers.
STEP FOUR
BI reporting tools synthesize findings into visualizations, dashboards, and reports.
STEP FIVE
Executives utilize visualizations and dashboards to make business decisions.

Figure 1: Functional Architecture of Business Intelligence

3.2 Optimization:

1. data strategy drives performance

- It minimize the number of fields.
- It minimize the number of records.
- Optimize to speed up future queries by materializing calculations, removing columns and the use of accelerated views

2. Reduces the data points in your view

- Practice guided analytics so there is no need to fit everything you plan to show in a single view.
- Remove dimensions from the detail shelf that are not needed.
- Explore and try to display the data in different types of views.

3. Limit the filters by number and type

- Reduce the number of filters in use because excessive filters on a view will create a more complex query, which takes a long time to return results.
- Use an include filter because exclude filters load the entire domain of a dimension while including filters do not. An include filter runs much faster than an exclude filter, especially for dimensions which has many members.
- Use a continuous date filter because continuous date filters can take the advantage of the indexing properties in your database and are faster than discrete data filters.
- Use Boolean or numeric filters
- Use parameters and action filters as these reduces the query load.

4. Optimize and materialize the calculations

- Perform calculations in the database
- Reduce the number of nested calculations.
- Try to use MIN or MAX functions instead of AVG. AVG requires more processing than MIN or MAX.

4 KPI:

Dashboards will be implemented to display certain KPIs and relevant indicators for the disease. 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.

Power BI

Key indicators:

1. Percentage of People Having Heart Disease and Age Distribution according to Gender

2. Gender Distribution Based on Heart Disease

3. Chest Pain Experienced by People Suffering from Heart Disease

4. Blood Pressure, Cholesterol Level and Maximum Heart Rate of People According to their Age and Heart Disease Patients.

5. ST Depression Experienced by People According to their age and heart disease.