

High Level Design (HLD)

Customer Lifetime Value Prediction

Revision Number: 1.0

Last date of revision: 20/08/2021

Soumit Kar

Contents

Document Version Control	2
Abstract	3
1 Introduction	5
1.1 Why this High-Level Design Document?	5
1.2 Scope	5
2 General Description	6
2.1 Product Perspective & Problem Statement	
2.2 Tools used	6
3 Design Details	7
3.1 Functional Architecture and Model Building	7
3.2 Optimization	8
4 KPIs	9
4.1 KPIs (Key Performance Indicators)	9
5 Deployment	

[illegible]

Abstract

Predict the customer life-time value for an Auto Insurance Company. Its revenue will increase if the amount claimed by customers who have insured in the company is low. Bookish definition: CLV is the total revenue the client will derive from their entire relationship with a customer. CLV gives the marketers an intuition of market structures, financial planning and other future consequences. It helps marketers to find optimum marketing spending to acquire and retain customers.

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.

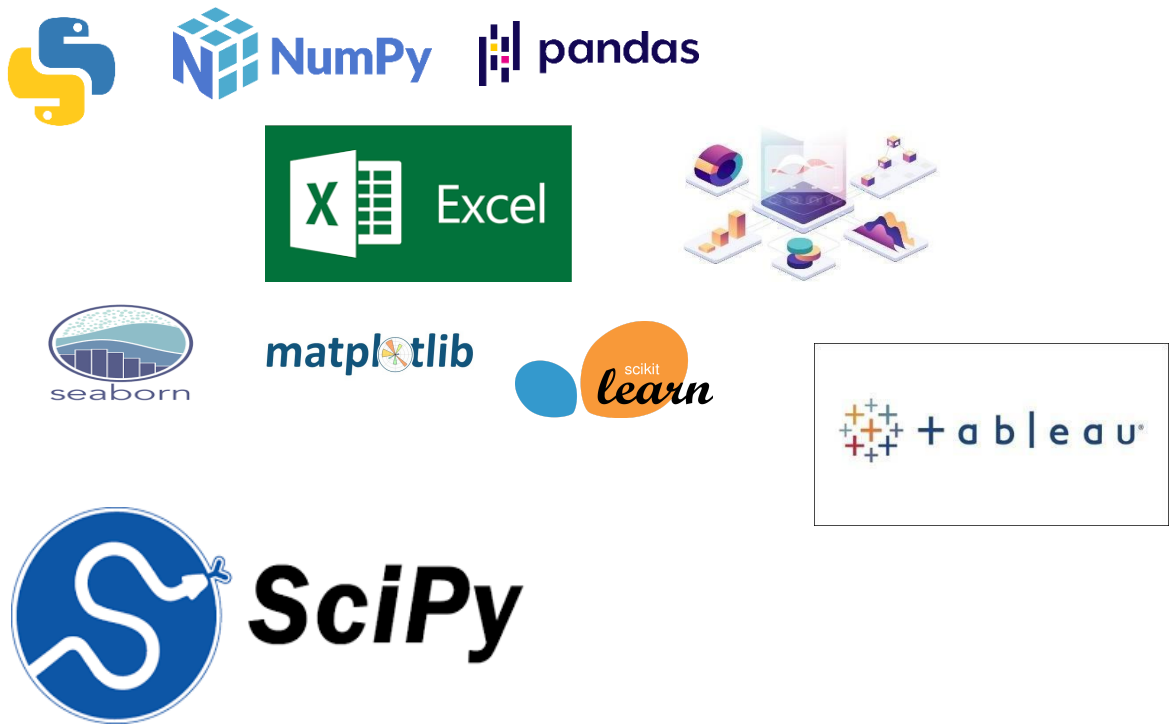
2 General Description

2.1 Product Perspective & Problem Statement

The Customer Lifetime Value Prediction is a machine learning based regression model which will help us to predict the Auto insurance CLV and take the necessary action. The objective of the project is to perform data visualization techniques to understand the insight of the data. This project aims to apply Business Intelligence tool Tableau to get a visual understanding of the data.

2.2 Tools used

Business Intelligence tools and libraries such as Numpy, Pandas, Seaborn, Matplotlib, Scikit-Learn, Excel, Tableau are used to build the whole framework.



3 Design Details

3.1 Functional Architecture and Model Building

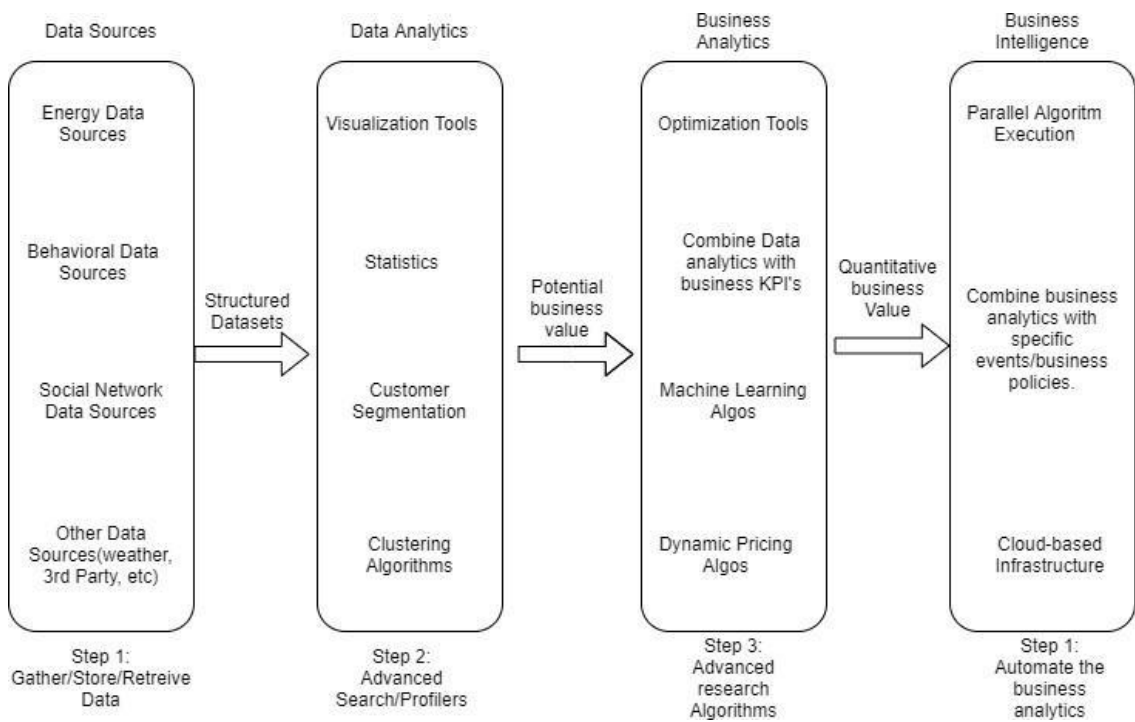
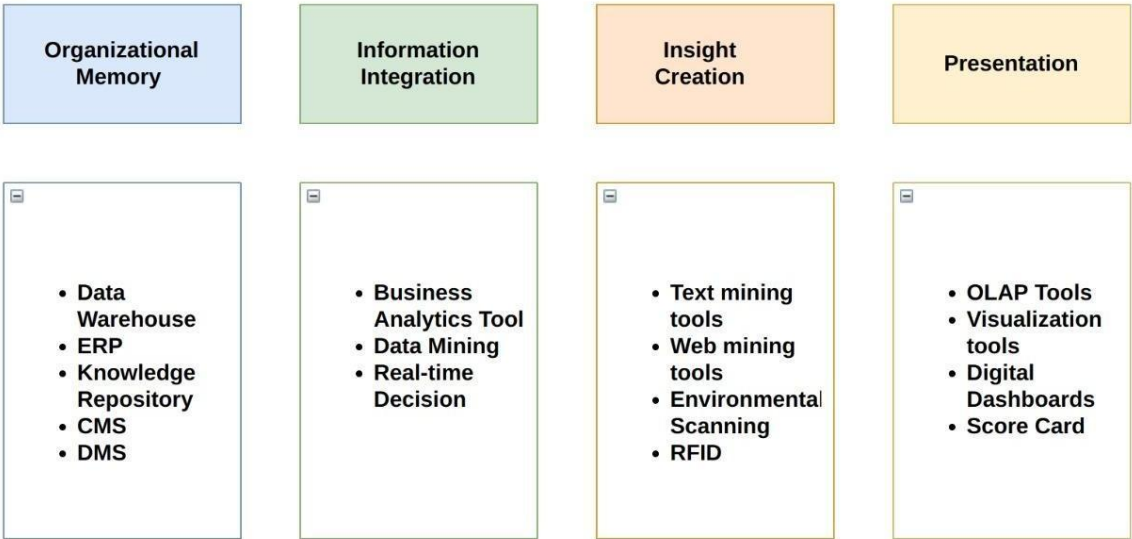
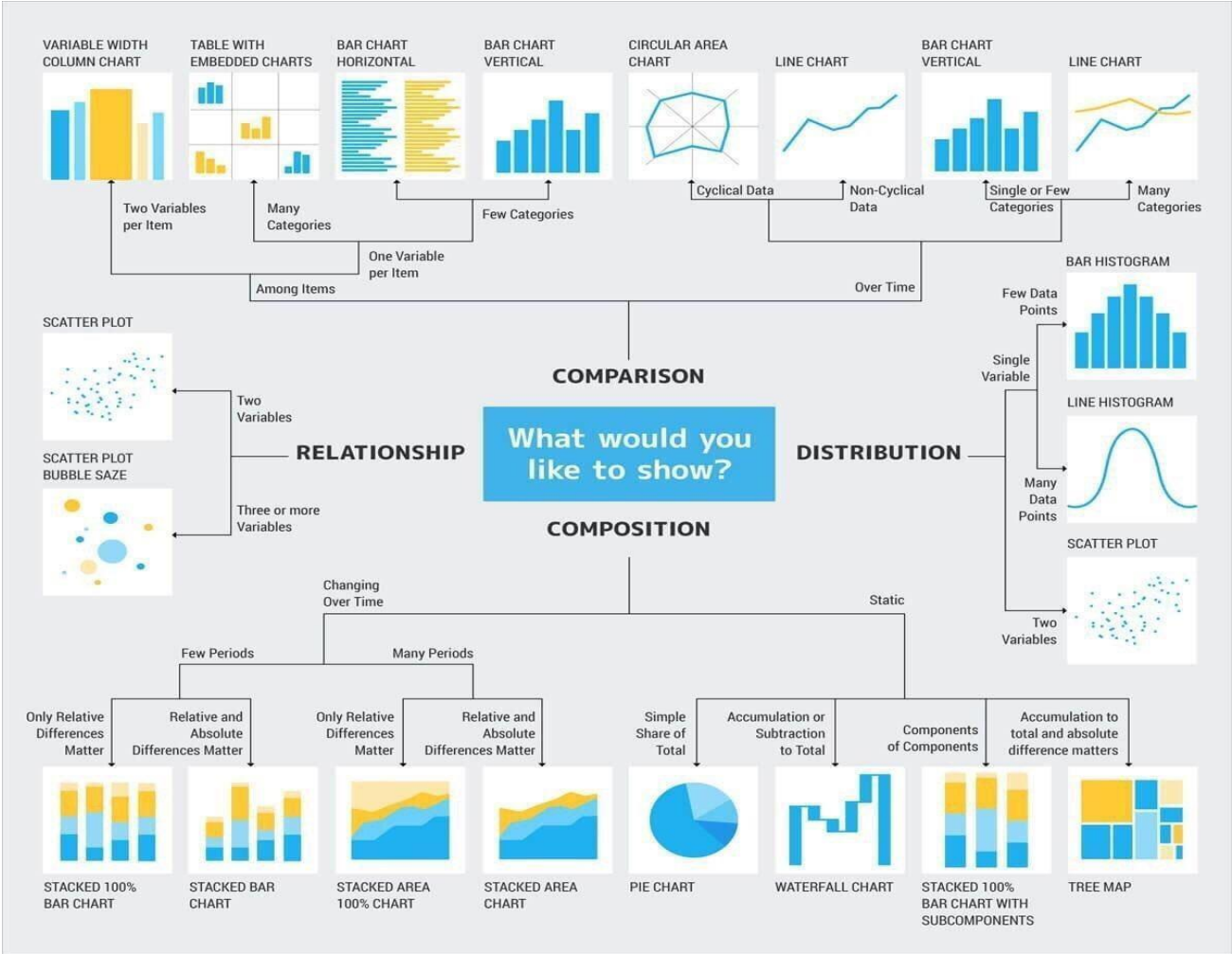


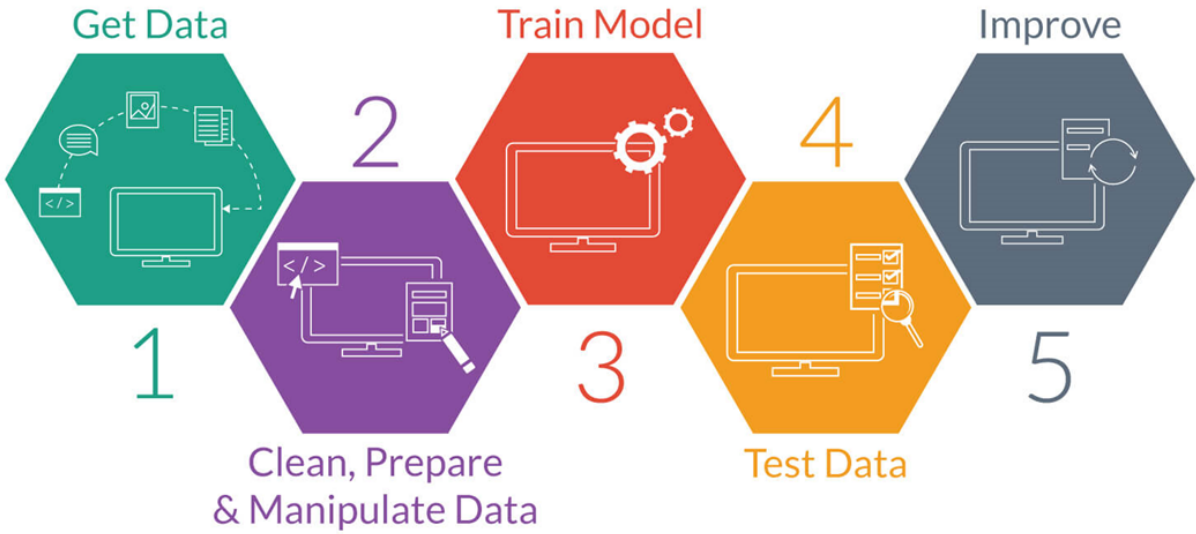
Figure 1: Functional Architecture of Business Intelligence

How BI Really Works





Model Building



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 [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 table calculations in the view. The more granular the calculation, the longer it takes.

- 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

4 KPIs

Dashboards will be implemented to display and indicate certain KPIs and relevant indicators for the customer lifetime value.

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)

Key indicators displaying a summary of the customer lifetime value and its relationship with different metrics

1. Impact of Policy, CLV, No of Complaints on Demographic factors of customers.
2. Impact of CLV on Customer centric factors (income, employment status, vehicle size and class, no of policies)
3. Influence of Business centric parameter on CLV
4. Influence of Customer centric parameter on CLV
5. Influence of Demographic parameter on CLV

5 Deployment

Prioritizing data and analytics couldn't come at a better time. Companies, no matter what size, are 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 Tableau at scale, as well as organizing, orchestrating, and unifying disparate sources of data for business users and experts alike to author and consume content.

Deployment Process

TABLEAU 10 SERVER ARCHITECTURE

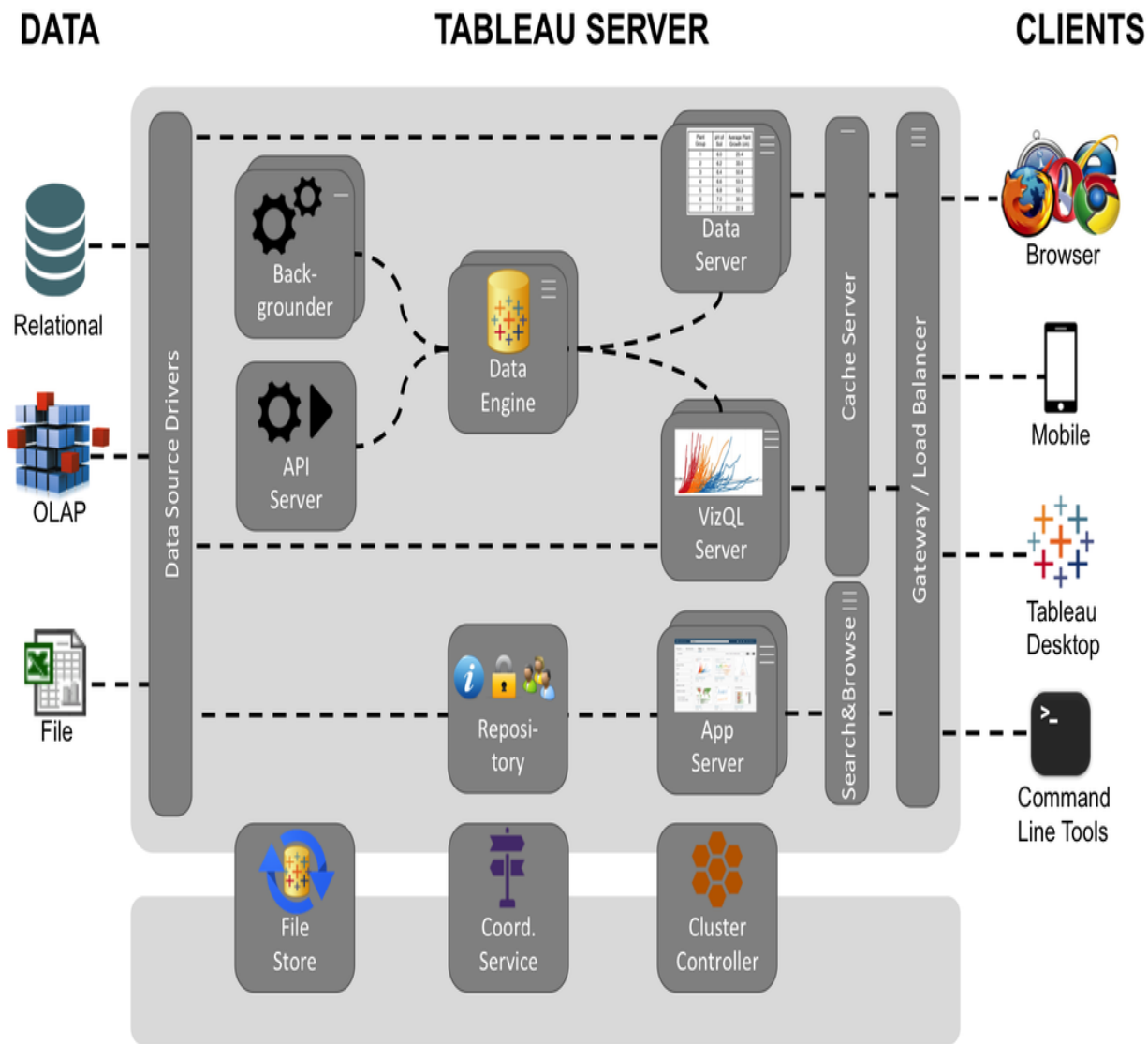


Tableau prioritizes choice in flexibility to fit, rather than dictate, your enterprise architecture. Tableau Server and Tableau Online leverage your existing technology investments and integrate into your IT infrastructure to provide a self-service, modern analytics platform for your users. With on-premises, cloud, and hosted options, there is a version of Tableau to match your requirements. Below is a comparison of the three types:

TYPE PROS CONS

Tableau Server - On Premises

- Full control of hardware and software
- Infrastructure and data remain behind your firewall
- Need dedicated administrators to manage hardware and software
- Additional infrastructure needed to access off-network (mobile, external)

Tableau Server - Public Cloud (IaaS)

- Full control of software on managed hardware
- Puts infrastructure in same place as data (for migration to cloud)
- Flexibility to spin up/down hardware as needed
- Need dedicated administrators to manage software
- Additional infrastructure needed to access off-network (mobile, external)

Tableau Online (SaaS)

- Fully hosted solution (hardware, software upgrades)
- Fast to deploy
- Easy for external audience to access
- Single-site in multi-tenant environment
- Cubes are not supported
- No guest account access