**Low Level Design**

Google Analytics Customer Revenue Prediction

|  |  |
| --- | --- |
| Written By | Teena Jain |
| Document Version | 1.0 |
| Last Revised Date | 25-March-2022 |

|  |  |  |  |
| --- | --- | --- | --- |
| Date Issued | Version | Description | Author |
| 25-March-2022 | 1 | Initial LLD -V 1.0 | Teena Jain |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Contents

[1. Introduction 4](#_Toc99008080)

[1.1. What is Low-Level design document? 4](#_Toc99008081)

[1.2. Scope 4](#_Toc99008082)

[2. Architecture 4](#_Toc99008083)

[3. Architecture Description 4](#_Toc99008084)

[3.1 Data Description 4](#_Toc99008085)

[3.2 Data Transformation 5](#_Toc99008086)

[3.3. Data Insertion into Database 5](#_Toc99008087)

[3.4. Export Data from Database 5](#_Toc99008088)

[3.5. Data Pre-processing 6](#_Toc99008089)

[3.6. Model Building 6](#_Toc99008090)

[3.7. Data Validation 6](#_Toc99008091)

[3.8. User Data Inserting into Database 6](#_Toc99008092)

[3.9. Recipe Recommendation & Saving Output in Database 6](#_Toc99008093)

[3.10. Deployment 6](#_Toc99008094)

# Introduction

## What is Low-Level design document?

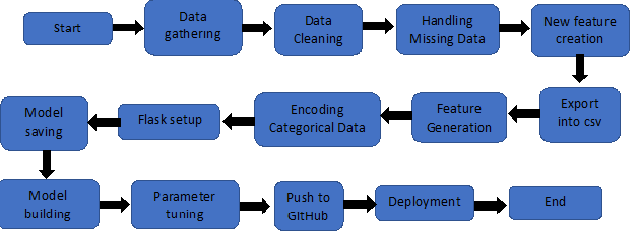
The goal of LLD or a low-level design document (LLDD) is to give the internal logical design of the actual program code for Store Sales Prediction. LLD describes the class diagrams with the methods and relations between classes and program specs. It describes the modules so that the programmer can directly code the program from the document.

## Scope

Low-level design (LLD) is a component-level design process that follows a step-by-

step refinement process. This process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work

# Architecture



# 3. Architecture Description

## 3.1 Data Description

I got 35 GB+ Data set in CSV format and than stored in Cassandra database. Some information is saved in (JSON) format. This Data Contains 1708337 rows and 13 columns in training set.

|  |  |
| --- | --- |
| Variable | Description |
| fullVisitorId. | A unique identifier for each user of the Google Merchandise Store |
| channelGrouping | The channel via which the user came to the Store. |
| date | The date on which the user visited the Store. |
| device | The specifications for the device used to access the Store. |
| geoNetwork | This section contains information about the geography of the user. |
| sessionId | A unique identifier for this visit to the store. |
| socialEngagementType | Engagement type, either “Socially Engaged” or “Not Socially Engaged”. |
| totals | This section contains aggregate values across the session. |
| trafficSource | This section contains information about the Traffic Source from which the session originated. |
| visitId | An identifier for this session. This is part of the value usually stored as the \_utmb cookie. This is only unique to the user. For a completely unique ID, you should use a combination of fullVisitorId and visitId. |
| visitNumber | The session number for this user. If this is the first session, then this is set to 1. |
| visitStartTime | The timestamp (expressed as POSIX time). |

## 3.2 Data Transformation

In the Transformation Process, we will convert our original dataset in which some columns are in JSON format to CSV Format. And will merge it with the dataset.

## 3.3. Data Insertion into Database

a. Database Creation and connection - Created a database with name passed and than open the connection to the database.

b. Table creation in the database.

c. Insertion of files in the table

## 3.4. Export Data from Database

Data Export from Database - The data in a stored Cassandra database is exported as a CSV file to be used for Data Pre-processing and Model Training.

## 3.5. Data Pre-processing

Data Pre-processing steps: It includes Null values handling, Imbalanced data set handling, Removal of unwanted columns for model training, feature selection, Handling columns below a threshold, encoding categorical data etc.

## 3.6. Model Building

For model building, algorithms will be passed with the best parameters derived from Random. Search CV for classification and Regression model. Classification model determines whether customer would return during test time window. for Regression model, picking only the customers who returned during test time window. Now, we run LightGBM model for 10 iteration and took average of those.

## 3.7. Data Validation

Here Data Validation will be done, given by the user.

## 3.8. User Data Inserting into Database

Collecting the data from the user and storing it into the Cassandra database.

## 3.9. Recipe Recommendation & Saving Output in Database

After calling model Recipe/Output will be recommended, this output will be saved in Database, and it will be used to show the same Output.

## 3.10. Deployment

We will be store the predicted file into client database.

This is a workflow diagram.

|  |  |  |
| --- | --- | --- |
| **Test Case Description** | **Pre-Requisite** | **Expected Result** |
| Verify whether the Application URL is  accessible to the user | 1. Application URL  should be defined | Application URL should be  accessible to the user |
| Verify whether the Application loads completely for the user when the URL is accessed | 1. Application URL is accessible 2. Application is deployed | The Application should load completely for the user when the URL is accessed |
| Verify whether a user is able to see input fields while opening the application | 1. Application is accessible 2. The user is able to see the input fields | Users should be able to see input fields on logging in |
| Verify whether a user is able to enter the input values. | 1. Application is accessible 2. The user is able to see the input fields | The user should be able to fill the input field |
| Verify whether a user gets predict button to submit the inputs | 1. Application is accessible 2. The user is able to see the input fields | Users should get Submit button to submit the inputs |
| Verify whether a user is presented with recommended results on clicking submit | 1. Application is   accessible   1. The user is able to see the input fields. 2. The user is able to see the submit button | Users should be presented with recommended results on clicking submit |
| Verify whether a result is in accordance with the input that the user has entered | 1. Application is accessible 2. The user is able to see the input fields. 3. The user is able to see the submit button | The result is accordance with the input that the user has entered |