**Architecture 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 | Architecture -V 1.0 | Teena Jain |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Contents

[1 Introduction 4](#_Toc99083295)

[1.1 What is Low-Level design document? 4](#_Toc99083296)

[1.2 Scope 4](#_Toc99083297)

[1.3 Constraints 4](#_Toc99083298)

[1.4 Out of Scope 4](#_Toc99083299)

[2 Technical specifications 4](#_Toc99083300)

[2.1 Google Analytics Customer Revenue Prediction Dataset Overview 4](#_Toc99083301)

[2.1.1 Training database 5](#_Toc99083302)

[2.1.2 Test Database 5](#_Toc99083303)

[2.1.3 Input Schema 5](#_Toc99083304)

[2.2 Predicting Revenue 5](#_Toc99083305)

[2.3 Logging 6](#_Toc99083306)

[2.4 Database 6](#_Toc99083307)

[3 Technology stack 6](#_Toc99083308)

[4 Proposed Solution 6](#_Toc99083309)

[5 Model Training or validation workflow 6](#_Toc99083310)

[6 User I/O workflow 7](#_Toc99083311)

[7 Key Performance Indicators (KPI) 7](#_Toc99083312)

**Abstract:**

Create an automated system for predicting potential future business, finding potential customers based on the various parameters as decided by the machine learning algorithm. The purpose of the document is to explain the High architecture that would be used for developing the Google Store revenue prediction system. we are given with the user's past data and transactions (when they logged into G-store). so, by using this data we need to predict the future revenue will be created by those customers. We will build a predictive model using G-store data set to predict the total revenue per customer that helps in better use of marketing budget, and we will also interpret the most impacting element on the total revenue prediction using different models.

# 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

## Constraints

We will only be selecting a few columns of the dataset.

## Out of Scope

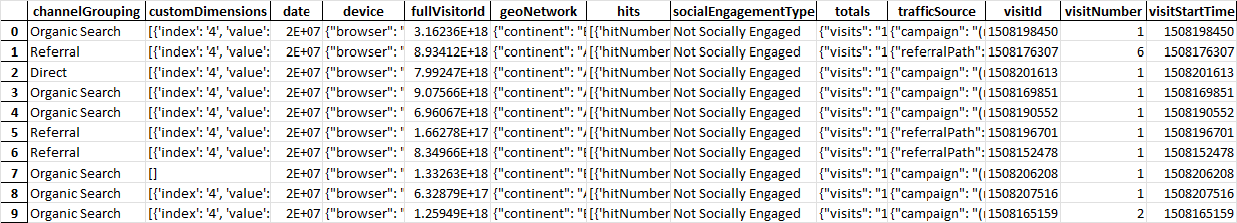
Delineate specific activities, capabilities, and items that are out of scope for the project.

# Technical specifications

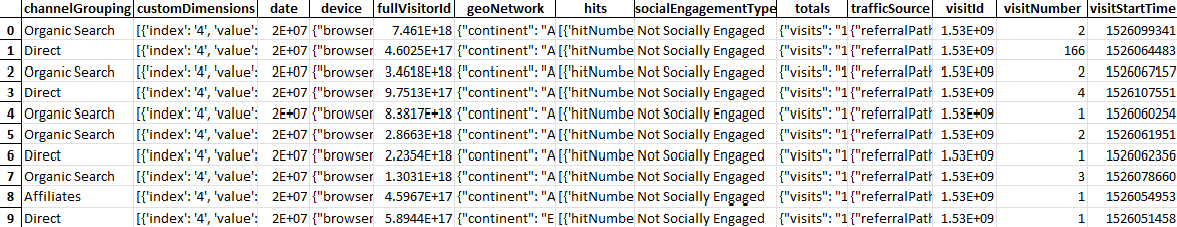
## Google Analytics Customer Revenue Prediction Dataset Overview

Consists of two tables one for training the model and other to test the accuracy.

### Training database



### Test Database



### Input Schema

|  |  |  |
| --- | --- | --- |
| **Feature Name** | **Datatype** | **Required** |
| Channel Grouping | String | Required |
| Custom Dimension | JSON format | Not Required |
| Date | Date format | Required |
| Device | JSON format | Required |
| FullVisitorId | String | Required |
| GeoNetwork | JSON format | Required |
| Hits | JSON format | Not Required |
| SocialEngagementType | String | Not Required |
| Totals | JSON format | Required |
| TrafficSource | JSON format | Required |
| VisitId | String | Not Required |
| VisitNumber | Integer | Required |
| VisitStartTime | Time Format | Required |

## Predicting Revenue

* The model evaluates the logarithm of revenue generated per user.
* The system gives the required information.
* The predicted database is stored into client’s database.

## Logging

We should be able to log every activity done by the user.

* The System identifies at what step logging required
* The System should be able to log each system flow.
* System should not be hung even after using so many loggings.
* Logging just because we can easily debug issues, so logging is mandatory to do.

## Database

System needs to store every request into the database, and we need to store it in such a way that it is easy to retrain the model as well.

* The user has given all the required information
* The system stores each data given by the user.

# Technology stack

|  |  |
| --- | --- |
| Frond End | HTML/CSS |
| Backend | Python Flask |
| Database | Cassandra |

# Proposed Solution

Based on the actual research paper, if we are using history of the revenue to predict the future revenue then we might want to consider using tree algorithm. However, drawing a baseline in the form of some Machine Learning algorithm would be helpful. Here i have used LightGBM (Light Gradient Boosting Machine).

# Model Training or validation workflow

LightGBM splits the tree leaf-wise as opposed to other boosting algorithms that grow tree level-wise. It chooses the leaf with maximum delta loss to grow. Since the leaf is fixed, the leaf-wise algorithm has lower loss compared to the level-wise algorithm. Leaf-wise tree growth might increase the complexity of the model and may lead to overfitting in small datasets.  
Below is a diagrammatic representation of Leaf-Wise Tree Growth

Chart, scatter chart

Description automatically generated

# User I/O workflow

Exceptional Handling

Ask relevant info from user

Check the info

Prediction

Predict

Storing data in the database

Yes

No

Data Pre-processing

Show error with proper exception handling

Data Validation

Success

Failure

# Key Performance Indicators (KPI)

* Number of transactions of user.
* Session Quality Dimension.
* First session from the period start.
* Number of page views.
* Operating System.
* Country.