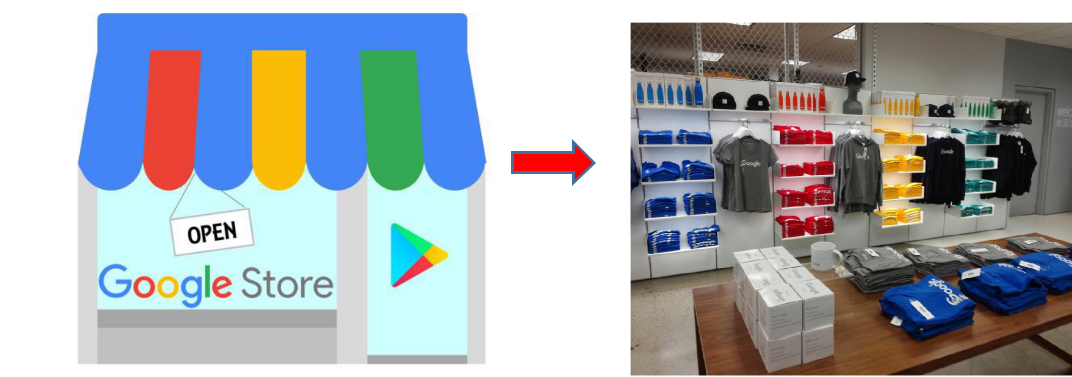
**High Level Design Document**

**Google Analytics Customer Revenue**

**Prediction**

****

**Team Leader: Devleena Banerjee**

**Created By: Amrit, Manish, Satyadeep**

**Introduction**

The Google Merchandise store is the E-commerce section from Google which sells Google **SWAG**(Stuff We All Get).As for every business, the number of customer that generaterevenue will be far less than the total customers that the business interacts with. So for every business it's really important to understand, analyse and predict the areas of its revenue generation. The 80/20 rule has proven true for many businesses–only a small percentage of customers produce most of the revenue. As such, marketing teams are challenged to make appropriate investments in promotional strategies.

**Problem Statement**

* 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’re challenged to analyze a Google Merchandise Store customer dataset to predict revenue per customer.
* In every business it was proven about 80–20 rule., this rule tells us 80% of our revenue will be generated by only 20% of our potential customers. So our goal is to predict the revenue that is going to be generated by those potential customers in the near feature. So that marketing teams will invest appropriate money on promotional strategies to attract potential customers.
* In simple words we are given with the users 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.
* So Google provided Merchandise customer dataset and no.of transactions per customer. 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.

**Objective**

The objective of this document is to present the brief overview of the technical architecture of the Google Store revenue prediction system. It describes the complete journey -> Data Collection to the Best Model Selection to the Deployment of the ML model for end user’s usage.

**Data Collection Strategy**

We imported dataset from **[Kaggle](https://www.kaggle.com/c/ga-customer-revenue-prediction/data)**and we were assigned a task of customer revenue prediction, for their transactions in the future.

**Data Interpretation**

The dataset consists of Traffic source data that contains information about where website visitors originate. This includes data about organic traffic, paid search traffic, display traffic, etc. Also, Content data that contains information about the behaviour of users on the site. This includes the URLs of pages that visitors look at, how they interact with content, etc. And, Transactional data that contains information about the transactions that occur on the Google Merchandise Store website.

* A training set is implemented to build up a model, while a test (or validation) set is to validate the model build.
* We have been provided with train.csv and test.csv.
* train.csv – A training set is implemented to build up a model **(**training dataset **12 columns x 1.4GB)** contains user transactions.
* test.csv – a test set is to validate the model build **(**test dataset **12 columns x 1.25GB)** contains user transactions.
* sample\_submission.csv- a updated sample submission file in the correct format.
* Contains all fullVisitorIds in test*.csv.*
* Our task is to predict the revenue generated by each customer from different groups for future transactions by using the historical data from existing transaction table.

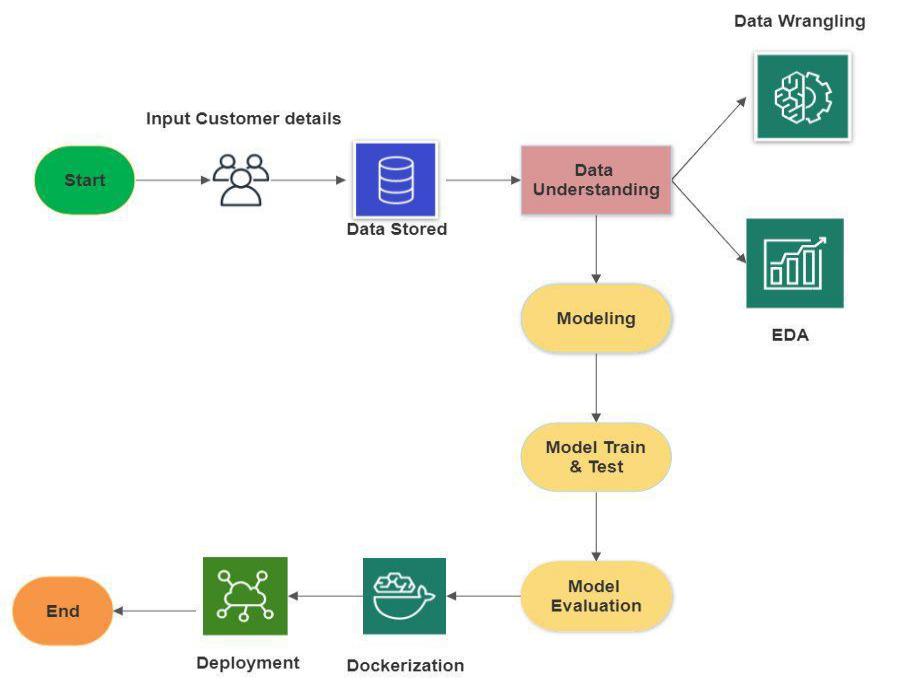
**Features/Attributes in Google store dataset -**

* **fullVisitorId** — A unique identifier for each user of the Google Merchandise Store. Ourfinal submission will be dependent on this and will be used for aggregation.
* **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.(*json column*)
* **geoNetwork** — This section contains information about the geography of the user.(*json**column*)
* **socialEngagementType** — Engagement type, either “Socially Engaged” or “Not SociallyEngaged”.
* **totals** — This section contains aggregate values across the session.(*json column*)
* **trafficSource** — This section contains information about the Traffic Source from whichthe session originated.(*json column)*
* **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 setto 1.
* **visitStartTime** — The timestamp (expressed as POSIX time).
* **hits** — This row and nested fields are populated for any and all types of hits. Provides arecord of all page visits.
* **customDimensions** — This section contains any user-level or session-level customdimensions that are set for a session. This is a repeated field and has an entry for each dimension that is set



*Total revenue generation variable is our top target variable.*

.**Architecture - Flow Diagram**

****

**Flow Diagram - Explanation**

* **Data wrangling** - The process of cleaning and unifying messy and complex data setsfor easy access and analysis. We are handling complex JSON objects, parsing, flattening such data objects to create structured dataset.
* **EDA** - Exploratory Data Analysis (EDA) is an approach/philosophy for data analysisthat employs a variety of techniques (mostly graphical) to.
* **Modeling** - The process of modeling means training a machine learning algorithm topredict the labels from the features, tuning it for the business need, and validating it on holdout data. The output from modeling is a trained model that can be used for inference, making predictions on new data points.
* **Model Evaluation** - After data splits into Train & Test and predicts the result.
* **Dockerizing** - An application is the process of converting an application to run withina Docker container.
* **Deployment** - Deployment of an ML-model simply means the integration of themodel into an existing production environment which can take in an input and return an output that can be used in making practical business decisions.

**Data Mining**

Data mining is the process of extracting useful information from large amounts of data.

Different Data Mining techniques:

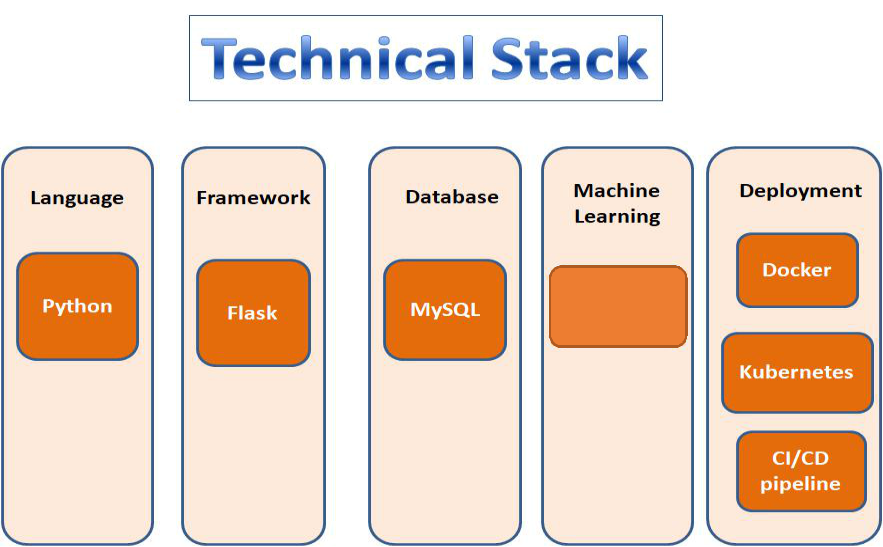
* Classification
* Regression
* Clustering
* Association rule learning
* Anomaly detection

The given problem of predicting revenue for a perticular group of customer is a **regression** problem. With the given data various ML algorithms like Decision Tree classifier, Support vector Machines, K nearest neighbor can be used.

For greater control of model variance ensemble approaches like Bagging (Random Forest Classifier) or Boosting (Gradient Boosting classifier, Adaboost) could be used.

**Technical Stack**

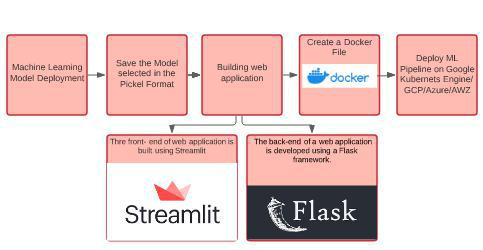
The technical landscape of the Google Store revenue prediction system is shown below.



**Regression**

**Deployment Strategy**

The deployment of machine learning models is the process of making models available in production where web applications, enterprise software and APIs can consume the trained model by providing new data points and generating predictions.



**Step1 — Model building**

The machine learning model which is finalized is saved as a pickle file.

**Step2 — Building Web Application**

Now that our machine learning pipeline and model are ready we will start building a web application that can connect to them and generate predictions on new data in real-time. This application will support ‘Online’ as well as ‘Batch’ predictions through a csv file upload. There are two parts of this application:

**Front-end (designed using Streamlit)** -Streamlit is an open-source Python library thatmakes it easy to build beautiful custom web-apps for machine learning and data science. **Back-end (developed using Flask in Python)**-

The back-end of a web application is developed using a Flask framework. It is a framework that allows you to build web applications. A web application can be a commercial website, a blog, e-commerce system, or an application that generates predictions from data provided in real-time using trained models. Before we publish the applicat ion on Google Cloud platforms to test the web app locally. Open Anaconda Prompt and navigate to the folder where ‘app.py’ is saved on your computer and run the python file python app.py. Once executed, copy the URL into a browser, and it should open a web application hosted on your local machine (127.0.0.1). Try entering test values to see if the predict function is working.

**Step3 — Create a Docker file**

A container is a type of software that packages up an application and all its dependencies so the application runs reliably from one computing environment to another. Docker is a company that provides software (also called Docker) that allows users to build, run and manage containers while Docker’s containers are the most common to containerize our application for deployment we need a Docker image that becomes a container at runtime. A Docker image is created using a Docker file. A Docker file is just a file with a set of instructions. The Docker file is case-sensitive and must be in the project folder with the other project files.

**Step4 —Deploy a ML pipeline on GKE/Cloud (AWS/GCP/Azure):**

Google Kubernetes is a powerful open-source system developed by Google back in 2014, for running and managing containerized applications across a cluster of applications.

1. Sign-in to GCP console and go to Manage Resources and then click on Create New Project followed by importing the Project Code.
2. Set Environment Variable
3. Build the Docker image of the application followed by authenticating to the Container Registry.
4. Create Cluster
5. Deploy Application -To deploy and manage applications on a GKE cluster, you must communicate with the Kubernetes cluster management system
6. Expose the application to the internet.
7. Check Service

**Conclusion**

A customer revenue prediction system that will help marketing/business team to identify and invest on potential customer group.These customer base may include in different categories such as gender or age, shopping patterns etc, so that they can be targeted on the basis of their categories and marketing teams can be given the task to do some promotional activities on them.

**Contact Us**

If you would like to proceed with our High Level Design then you can sign the first page and return a copy to us by email. In any case please feel free to call us to discuss the quote, request more information or for any other reason.

We look forward to hearing from you soon!