Architecture Document

For

“Market Basket Project on E-Commerce”

Document Version Control

SL. No	Version No.	Date	Prepared By	Reviewed By	Comments
01	1.0	12th April, 2022	Irfan Razi		

Signature :

_____

Prepared by : Irfan Razi

Intern – Data Scientist

Signature : _____

Reviewed by :

Signature :

_____

Approved by : Sudhanshu Kumar

Founder iNeuron, Chief AI Engineer & CEO,

iNeuron.ai

Table of Contents

List of Figures

GENERAL

17th Floor tower A Brigade Signature Towers,
Sannatammanahalli, Bengaluru, 562129, INDIA

1.1 Introduction

The objective of this document is defining the Architecture and details followed for Market Basket Project.

1.2 Scope of this Document

The scope of the document is limited to Architecture approach of the Market Basket Project.

This section provides the general information about the Architecture approach for Market Basket Project on E-Commerce.

ARCHITECTURE

2 Architecture Details

This section describes the architecture details for Market Basket Project on E-Commerce. Kindly, refer the Figure 1 for the same.

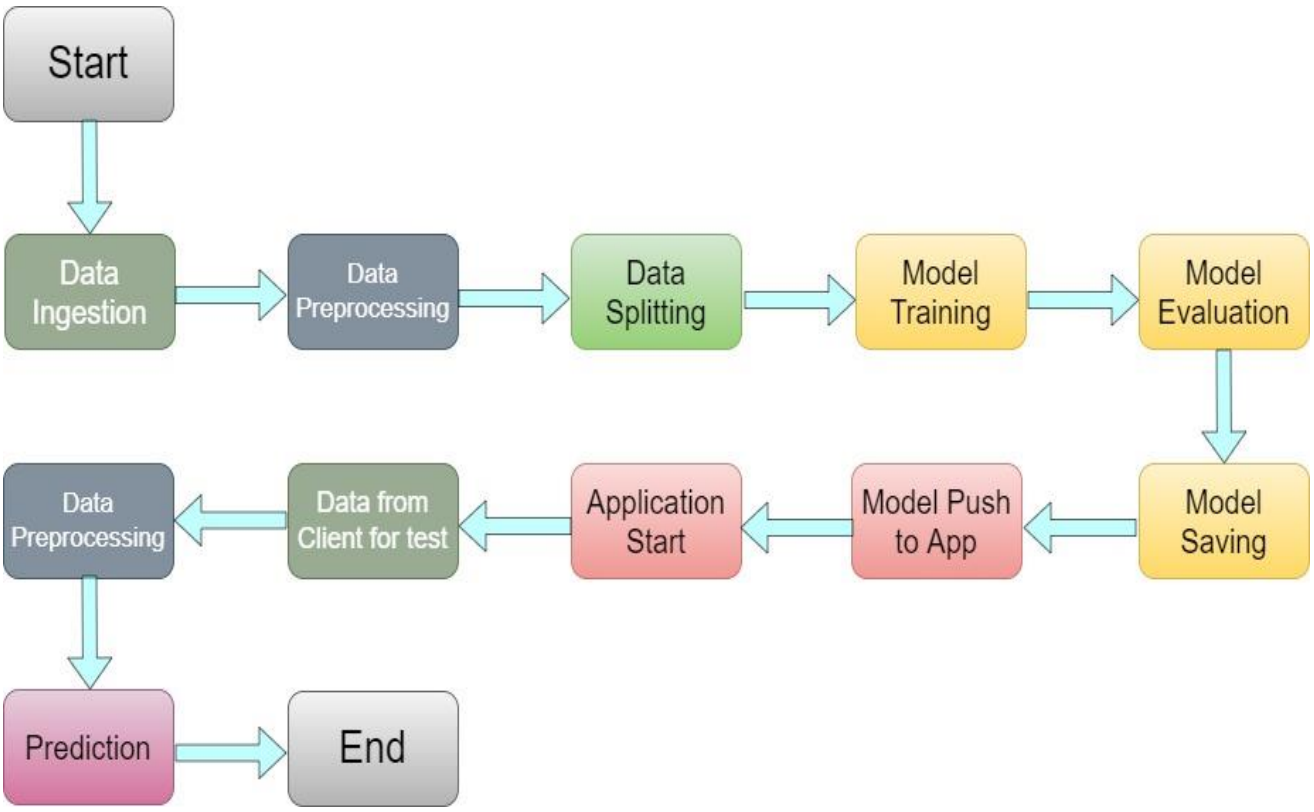

Figure 1: Architecture Details for Market Basket Project

2.1 Data Description

We will be using Brazilian E-Commerce Public Dataset by Olist. This is a Brazilian ecommerce public dataset of orders made at Olist Store. The dataset has information of 100k orders from 2016 to 2018 made at multiple marketplaces in Brazil. Its features allow viewing an order from multiple dimensions: from order status, price, payment and freight performance to customer location, product attributes and finally reviews written by customers. We also released a geolocation dataset that relates Brazilian zip codes to lat/lng coordinates.

2.2 Data Ingestion

Here, we will be ingesting all the batches of data from Cassandra database to our machine in csv format.

2.3 Data Pre-processing

We will do Exploratory Data Analysis of the data in Jupyter Notebook to get the complete understanding of the data. Based on that we can decide the strategy for Data Processing and validation. We may have to drop insignificant columns, handle missing values, handle imbalanced data, etc so that we can get a clean data for model training. For this, we have to write separate modules as per our need.

2.4 Data Splitting

We split the data for model training and model validation.

2.5 Model Training

We train our data with various ML models. Among those, Random Forest Classifier is the best fit model.

2.6 Model Evaluation

Model evaluation is done by classification report. Since, this is a problem of imbalanced data, we have to analyse and improve Recall score and F1-score, not just Accuracy.

2.7 Model Saving

After model training and evaluation, we will save the model for production.

2.8 Model Push to App

We are going to do the cloud setup for our model deployment. We are going to create Flask App and User Interface. We will integrate our model with it.

2.9 Data from Client for testing

Now, our Web-Application is ready and deployed to clouds. We can get the data from our clients and start testing the model.

2.10 Data Pre-processing

Client-data is also required to go through the same process as our train data has gone for model training.

2.11 Prediction

Finally, when we complete the prediction process with client's data, we convert it into csv format and share it to the client.