LOW LEVEL DOCUMENT

Introduction:

1.1. 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 Food Recommendation System. 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.

1.2. 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

2. Architecture

ML PIPELINE



DATA INGESTION:

Data ingestion is the process of collecting and importing data from various sources into a storage system, such as a data warehouse or data lake, where it can be further processed and analyzed. The data sources can be structured, semi-structured or unstructured data, and can come from a wide range of sources such as databases, APIs, files, sensors, logs, and more.

The process of data ingestion typically involves extracting data from the source, transforming it into a common format, and loading it into the destination storage system. This process can be automated using tools like ETL (extract, transform, load) or ELT (extract, load, transform) processes.

Data ingestion is a critical step in the data processing pipeline because it sets the foundation for further analysis and decision-making. By ingesting data from diverse sources, organizations can gain a comprehensive view of their operations, customers, and market trends. This data can then be used to identify patterns, make predictions, and drive business growth.

MODEL TRAINER

A model trainer is a software tool or framework used to build machine learning models. It allows users to specify the architecture of the model, the data to be used for training, and the optimization algorithm to be used to learn the parameters of the model.

The model trainer takes the input data and applies various transformations and pre-processing steps, such as normalization or feature extraction, to prepare the data for training. It then trains the model using the specified algorithm and hyperparameters. During training, the model is repeatedly presented with training data, and the algorithm adjusts the model parameters to minimize a predefined loss function, which measures the difference between the predicted output and the actual output.

The performance of the trained model is evaluated using a validation set or test set, which contains data that was not used during training. The model trainer can help users to fine-tune the model parameters and choose the best model based on the evaluation results.

Model trainers can be used for a wide range of machine learning tasks, including classification, regression, clustering, and natural language processing. Some popular model trainers include TensorFlow, PyTorch, scikit-learn, Keras, and Apache MXNet.

To evaluate a model, the dataset is usually split into a training set and a test set, or a training set, a validation set, and a test set. The model is trained on the training set, and the performance is evaluated on the test set or the validation set.

MODEL PUSHER:

It allows us to push the model on aws ,azure etc.

Model deploy:

We will be deploying the model to AWS.