Low Level Design (LLD)

Automlops Challenge

Revision Number: 1.0

Atufa Shireen

Document Version Control

Date Issued	Version	Description	Author
23th Nov 2021	1.0	First Version of Complete LLD	Atufa Shireen

Table of Contents

Abstract

The principal motive is to create an automated end to end platform which can perform various machine learning and software development life cycle automatically as far as possible.

1 Introduction

1.1 Why this Low-Level Design Document?

The purpose of this document is to present a detailed description of the AutoMl platform. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system and will be proposed to the higher management for its approval.

The main objective of the project is to create a platform for automating machine learning life cycle.

1.2. Scope

This software system will be a web application which will be designed to create a portal for performing ml operations, dashboards and loggings.

1.3. Constraints

We will be implementing only a few mlagos for now.

1.4. Risks

Document specific risks that have been identified or that should be considered.

1.5. Out Of Scope

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

2 Technical specifications

2.1. Input schema For Regression, Classification Problems

Feature name	Datatype	Size	Null/Required	Description
Project Name	String	--	Required	
File Regex	String	--	Required	Regex for ilename
Train Shema	Json	--	Required	Schema containing column names
Test Schema	Json	--	Required	Schema containing column names
Problem Type	String		Required	Regression or Classification
Target Column Name	String		Required	

2.2 Performing Model Training

- Preprocessing pipeline is applied based on the algorithm, and problem type.
- Using a bagging, boosting, stacking and blending approach to select the best model.

2.3 Logging

- We should be able to log every activity done by the user.
- The System identifies at what step logging is required.
- The System should be able to log each and every system flow.
- Developers can choose logging methods. You can choose database logging/ File logging as well.
- 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.

2.4 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 system stores users details in mongodb, files and models in cloud storage and database files in Bigquery warehouse.

2.5 Deployment

- AWS (Amazon Web Services)

3 Technology stack

Front End	HTML/CSS/JS/React
Backend	Python Django
Database	MongoDB/Bigquery
Deployment	AWS

4 Proposed Solution

A brief description of various stages in the ETL pipeline is described below:

1) **Extracting Data**
   Here the platform loads data from user input (though input box) in .csv extension.

- Future Update: Platform will be able to load data from any cloud storage(like AWS, GCP, Azure) or warehouse(Bigquery, Snowflake ,Redshift,Oracle) or ERP with various file extensions such as .h5 or .zip files.

2) **Transforming Data**
   Here the platform transforms data into HDFS using a python package for HDFS and pandas on the server.

- *Future Update: Platform will be able to extract, transform and load data using big data framework such as apache kafka for distributed system and spark for computation engine, for streaming and batch processing.*

3) **Loading Data**
   Platform will load the transformed data into a data warehouse, Bigquery to be able perform sql queries on the transformed data.

- *Future Update: The data warehouse might change based on the volume data source and cost of storing data.*

4) **Model Training**
   Platform provides the ability to the end user to train models using our library (***automl***) on the transformed and validated data based on request.The model training will add the best trained model to the cloud storage.
   Here the best trained model is returned after 4 stages of model training.

- *Future Update: The platform will have the ability to train models on schedules or based on event driven management, and log experiments using mlflow.*

5) **Performing predictions**
   Platform provides the ability to the end user to make predictions models on the transformed and validated data based on request by loading the best trained model from the cloud.

- *Future Update: The platform will have the ability to train models on schedules or based on event driven management.*

6) **Storage Service**
   Platform provides users to store data on chosen platforms(from aws and gcp).

- *Future Update: The platform will have the ability to store data in more than two clouds.*
-

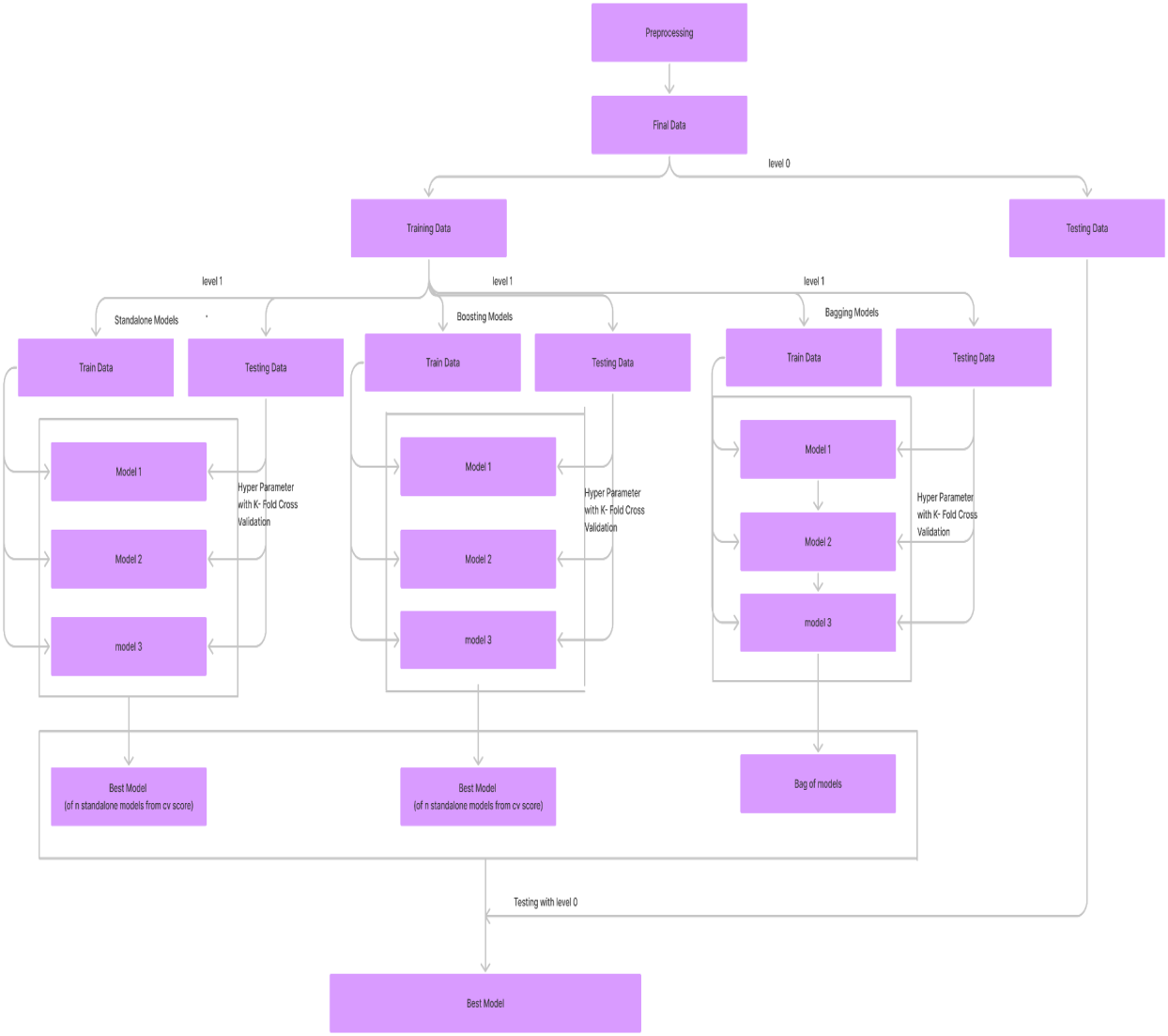
7) **Dashboard**
   Users can see metrics of the models trained using the report management.

- *Future Update: The Dashboard will have the ability to generate profile,feature importance, all KPIs report of the data.*

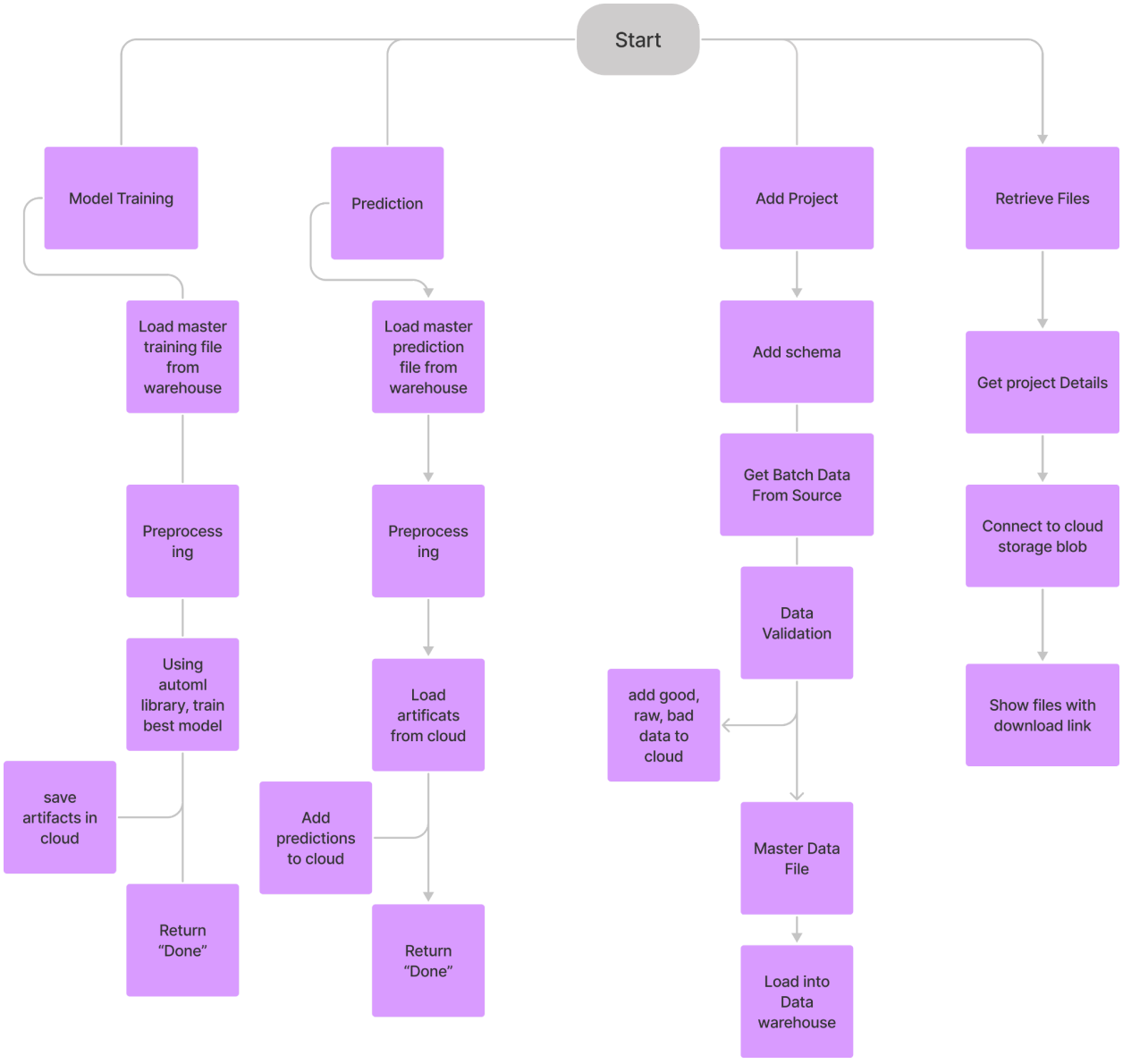
8) **Logging**
   Users can check logs of the various internal processing steps of the project using view logs in the project section.

- *Future Update: Users will have the ability to view preferred logs.*

5 Model training/validation workflow



6 User I/O workflow



9. Key performance indicators (KPI)

For Regression Problems

1. R2 (Adjusted)
2. RMSE
3. MAE

For Classification Problems

1. Classification report