Low Level Design (LLD)

Concrete Compressive strength Prediction

Written by	Yatrik Shah
Document version	0.1
Last revised date	Sep 1 2022

]

Version	Date	Autor	Comments
0.1	Sep 1, 2022	Yatrik Shah	Final LLD

Contents

1. Introduction....................4
1.1 What is Low-Level design document?....................4
1.2 Scope....................4
2. Architecture....................5
3. Architecture Description....................6
3.1 Data Description....................6
3.2 Data Validation....................6
3.3 Data Insertion into Database....................7
3.4 Export from Database....................7
3.5 Data Pre-processing....................7
3.5.1 Data Scaling....................7
3.5.2 Data Imputation....................7
3.6 Data Clustering....................7
3.7 Model Selection: Cross Validation and Hyperparameter Tuning....................8
3.8 Model Deployment....................8
3.8.1 Cloud Setup....................8
3.8.2 Pushing App to cloud / Start the Application....................8
3.9 Data From client to be predicted....................8

1. Introduction

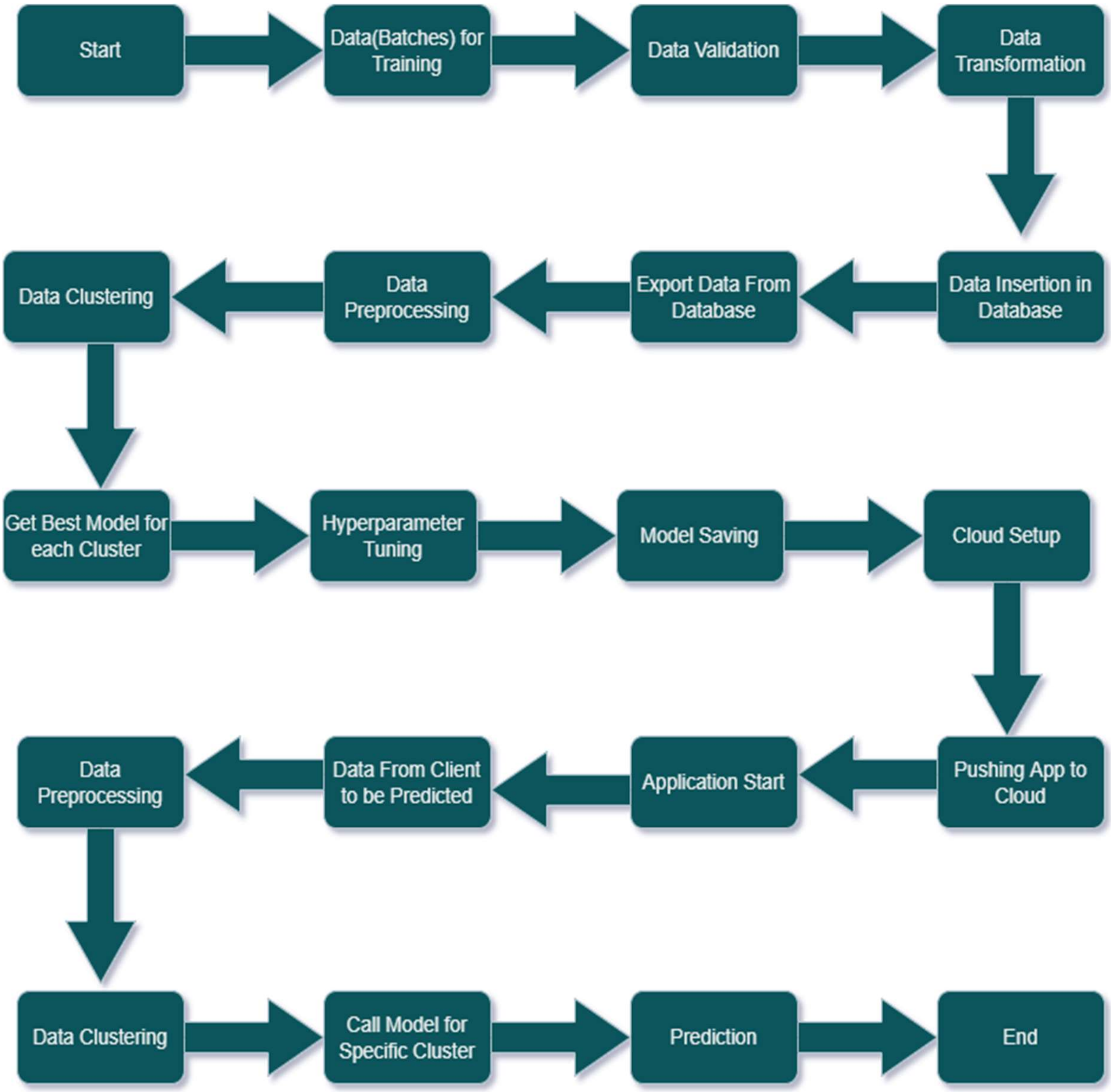
1.1What is Low-Level design document?

The goal of LLD or a Low-level design document is to give an internal logical design of the actual program code for the Concrete Compressive Strength Prediction 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 defined during data design work.

2. Architecture



3. Architecture Description

3.1 Data Description

Concrete is the most important material in civil engineering. The concrete compressive strength is a highly nonlinear function of age and ingredients. These ingredients include cement, blast furnace slag, fly ash, water, superplasticizer, coarse aggregate, and fine aggregate.

Number of instances (observations): 1030
Number of Attributes: 9
Attribute breakdown: 8 quantitative input variables, and 1 quantitative output variable

Source:

Original Owner and Donor
Prof. I-Cheng Yeh
Department of Information Management
Chung-Hua University,
Hsin Chu, Taiwan 30067, R.O.C.

Link:

https://archive.ics.uci.edu/ml/datasets/Concrete+Compressive+Strength

3.2 Data Validation

In Every Automated Model Training process data validation is a very significant part of pipeline. For model training we must check firstly of the data provided is up to the mark or not, i.e. The columns and their datatypes are as expected or not, or the number of columns is same as expected or not.

3.3 Data Insertion into Database

After Data Validation and Data Transformation the necessary step is dump the data into the database. That would be the final raw data for the pipeline. And then for further processing in the algorithm pipeline this would be the base data. i.e. Data Prep recessing

3.4 Export from Database

The data is exported in form of csv for the further processes. That would be used for the data clustering as well as final model training.

3.5 Data Pre-processing

3.5.1 Data Scaling

Here as Standardization is used for data scaling. It is useful in some of the Machine Learning Algorithms, Data Clustering and sometimes reduces the computation complexity hence it is one of the most important factors in the Machine Learning.

3.5.2 Data Imputation

For various reasons, many real world datasets contains missing values, so that missing data problem arises in almost every serious statistical analysis.

In Statistics Imputation is the process of replacing the missing data with the substitute values. So that they could be used in any statistical processes or analysis.

3.6 Data Clustering

K-Means algorithm will be used to create clusters in the pre-processed data. The optimum number of clusters is selected by plotting the Silhouette Score. The idea behind clustering is to implement different algorithms to train data in different clusters. The K-means model is trained over pre-processed data and the model is saved for further use in prediction.

3.7 Model Selection: Cross Validation and Hyperparameter Tuning

After getting the clusters from the data. Now the objective is to get best model with the best hyperparameters after cross validation, which gives the highest R2_score (R squared).

Here the best model, selected with the highest R2 score, having optimum hyperparameters, are saved as the final model for the respective clusters.

3.8 Model Deployment

3.8.1 Cloud Setup

Here the cloud platform for the deployment is chosen. Here Heroku platform is chosen for Deployment, And setting up the App as the requirements of the respective clod requirements and Servers and Frameworks are applies i.e. Flask having the Web Server Gateway Interface [WSGI].

3.8.2 Pushing App to cloud / Start the Application

After the clous setup the app is pushed to the cloud via the git / GitHub. After choosing and setting up the cloud platform, the next and one of the most important steps of ML Engineer is to push and start the application on the clous

3.9 Data From client to be predicted

Here the Application is all to go, and we are getting the data from the client to be predicted form the created Machine Learning Model.

4. Automation

Here Standard Modular coding as well as Automated Training approach is used.

Which basically means that after dumping all the training files into the folder called as “Training_BatchFiles” just press the button called “ReTrain” on the server, that will process following tasks within automatic training pipeline :

- Getting all the Training BatchFiles from the folder called “Training_BatchFiles”
- Perform data validation of those files
- Get continued with the perfectly validated files in data validation step
- Save the final training data into database
- Export data from database for further training process
- Data preprocessing
- Data Clustering
- Perform Cross Validation and Hyperparameter tuning separately for each model
- Select and Save best models for each clusters by replacing those with the previous models.