Low Level Design(LLD)

Concrete Compressive strength Prediction

Written By	Utkarsh Vataliya
Document Version	0.1
Last Revised Date	06 – sep -2024

Document Control

Change Record:

Version	Date	Author	Comments
0.1	6-sep-2024	Utkarsh Vataliya	Last final LLD

Contents

1. Introduction

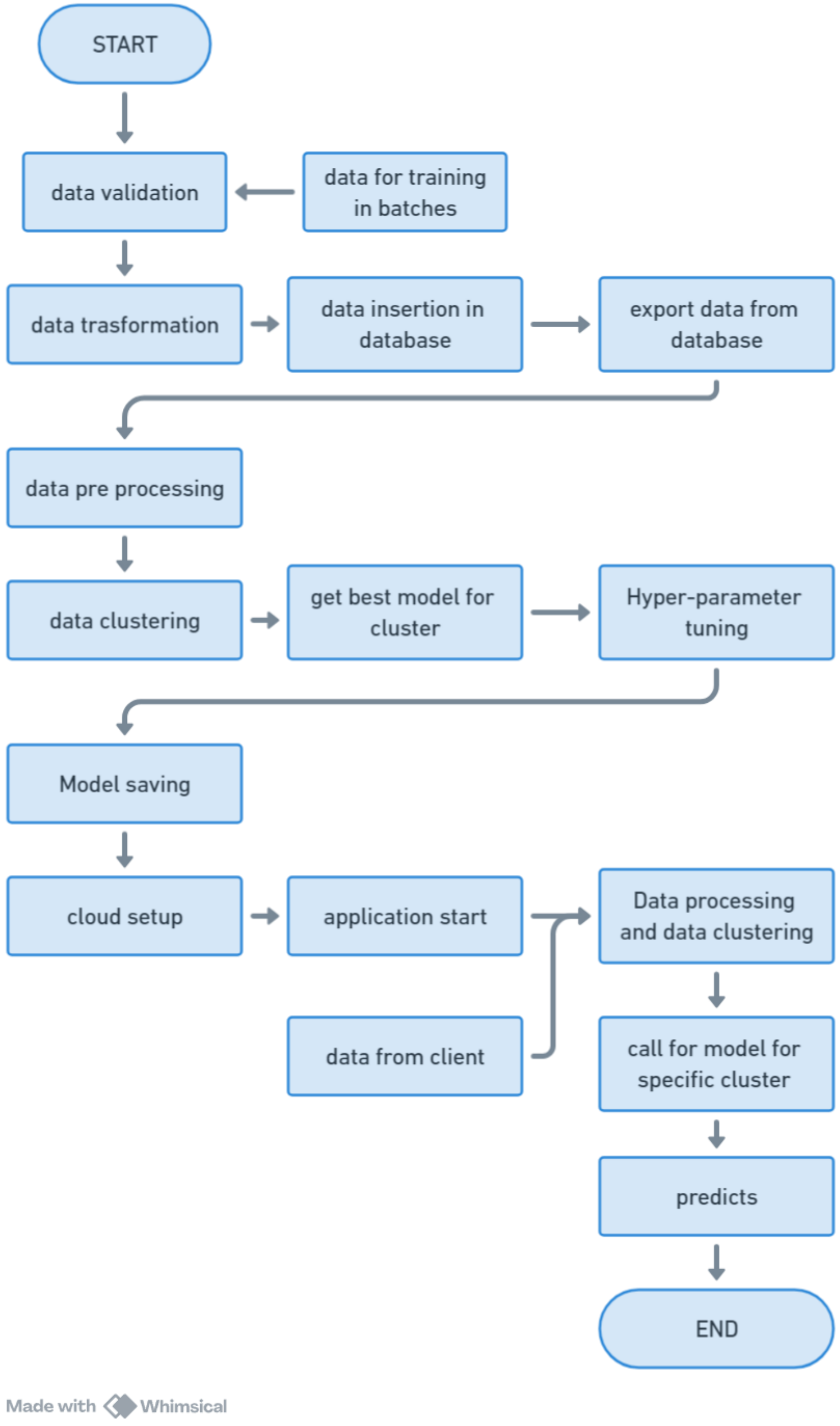
1.1 What is a Low-Level design document?

The goal of LLD or a low-level design document (LLDD) is to give the internal logical design of the actual program code for 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 refined 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	1030
Number of Attributes	9
Attribute breakdown	8 quantitative input variables, and 1 quantitative output variable
Missing Attribute Values	

Variable Information:
Given is the variable name, variable type, the measurement unit and a brief description. The concrete compressive strength is the regression problem. The order of this listing corresponds to the order of numerals along the rows of the database.
Name -- Data Type -- Measurement -- Description
Cement (component 1) -- quantitative -- kg in a m3 mixture -- Input Variable
Blast Furnace Slag (component 2) -- quantitative -- kg in a m3 mixture -- Input Variable
Fly Ash (component 3) -- quantitative -- kg in a m3 mixture -- Input Variable
Water (component 4) -- quantitative -- kg in a m3 mixture -- Input Variable
Superplasticizer (component 5) -- quantitative -- kg in a m3 mixture -- Input Variable
Coarse Aggregate (component 6) -- quantitative -- kg in a m3 mixture -- Input Variable
Fine Aggregate (component 7) -- quantitative -- kg in a m3 mixture -- Input Variable
Age -- quantitative -- Day (1~365) -- Input Variable

Source :

Kaggle : https://www.kaggle.com/datasets/elikplim/concrete-compressive-strength-data-set

3.2 Data Validation

Data validation is the process of checking the accuracy, integrity, and structure of data before using it for any purpose. It's a type of data cleansing that's often performed to ensure data is fit for financial accounting or regulatory compliance.

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

3.3 Data Insertion into Database

3 Easy Ways to Insert Data into a Database Table. There are several ways to add data to a database table. One of the most common methods is using SQL statements: CREATE TABLE and INSERT INTO. If your data is in an Excel or CSV file with many rows and columns, you can insert data more quickly using Coginiti Data Insert.

After Data Validation and Data Transformation the necessary step is to 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 Preprocessing

3.4 Export from Database

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

3.5 Data preprocessing

Data preprocessing is the process of cleaning, organizing, and transforming data before it's used for analysis or modeling. It's a crucial step in the data mining process, and it helps ensure that the data is accurate, reliable, and efficient.

3.6 Data preprocessing

Here 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 Machine Learning.

3.7 Data imputation

for various reasons, many real world datasets contain 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.

3.8 Data Clustering

Data clustering is a method of data mining that groups similar data points together. The goal of cluster analysis is to divide a dataset into groups (or clusters) such that the data points within each group are more similar to each other than to data points in other group

The 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.

3.9 Model Selection : Cross validation and Hyperparameter Tuning

Cross validation
A technique that helps you evaluate and compare the performance of different models by splitting a dataset into multiple subsets. The model is trained on one subset and tested on the remaining data, and the process is repeated multiple times. This provides a more accurate estimate of the model's performance than a single train-test split.

Hyperparameter tuning
A technique that helps you find the best set of hyperparameters for a model. You can use cross-validation to compare the performance of different hyperparameter settings during the tuning process. The model is trained and tested for each hyperparameter setting, and the results are summarized to provide an overall performance score

After getting the clusters from the data. Now the objective is to get the 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.

3.10 Model Deployment

Model deployment is the process of integrating a machine learning (ML) model into an existing software service so that it can be used to make predictions or classifications. This process is important because it allows organizations to automate decision-making, increase efficiency, and drive innovation

3.11 Pushing App to cloud / Start the Application

After the cloud 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 cloud

3.11 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 from the created Machine Learning Mode

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.