Low Level Design Document
Concrete Strength Prediction

Written by: Dhananjai Singh

Date: 24/08/2022

Table of contents:

1 Introduction

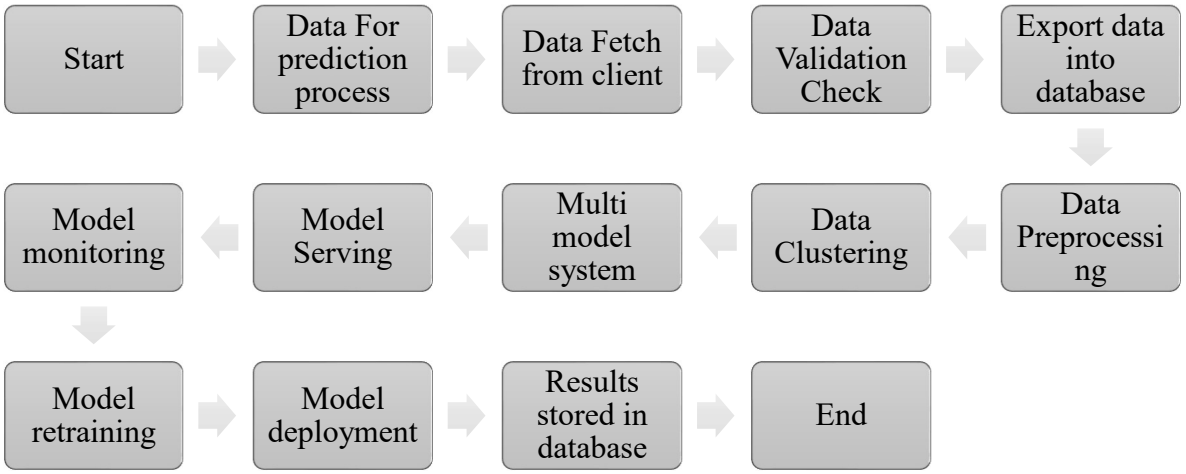
1.1 What is Low Level Design Document?

The goal low-level design document (LLD) is to give the internal logical design of the actual program code for Concrete Strength Prediction. 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

The Concrete Strength Prediction data set is publicly available. The file is in csv (comma separated values) that contains 9 columns (Cement, Blast Furnace Slag, Fly Ash, Water, Superplasticizer, Coarse Aggregate, Fine Aggregate, Age, Strength) and 1030 rows.

3.2 Data Validation

A Data Sharing Agreement is signed with the client that will clearly mention the following:

i. Number of filed transferred by the client
ii. Mode of file transfer
iii. Columns detail
iv. Number of rows in each data set
v. Language
vi. Frequency of the dataset
vii. File name convention and many more

It is mandatory to have data set as per the DSA. If dataset is not satisfying the above condition, it will be informed to client. And the data will be moved to archive folder.

3.3. Data injection into database

Once the data is validated with DSA, it is then loaded into database. Database is created on “MongoDB”. A new database named “Project” will be created that will contain all the tables related to the prediction. Inside the database a table named “Concrete” is created that will store the training data. Prediction data set is loaded into a table named “predictConcrete”. Final predictions are kept in the table “Results”. All the column names and its datatypes will be referred from DSA. Once the table is created then data will be loaded into the database.

3.4 Export Data from Database

Data stored in the MongoDB is exported as a CSV file into the folder ‘DataExtractedFromDB’ and the same is used for data pre-processing and model training.

3.5 Data preprocessing

The data in a stored database is exported directly into the PyCharm IDE and then preprocessing is initiated. Data Pre-processing steps include:

i. Null value handling using KNN imputer
ii. Handling columns with standard zero deviation or below a threshold, etc.
iii. Clustering: K-Means algorithm will be used to create clusters in the pre-processed data. The optimum number of clusters is selected based on silhouette

score. The idea behind clustering is to implement different algorithms to train data in different clusters. The K-means model is trained over preprocessed data and the model is saved for further use in prediction

3.6 Model Building

After clusters are created, we will find the best model for each cluster. For each cluster, algorithms will be passed with the best parameters derived from Grid-SearchCV. The adjusted r^2 score is use to evaluate the model. Similarly, the models will be selected for each cluster. All the models for every cluster will be saved.

3.7 Data from Client

Data from the client is collected in form of csv file. File must contain Cement, Blast Furnace Slag, Fly Ash, Water, Superplasticizer, Coarse Aggregate, Fine Aggregate, Age columns

3.8 Prediction Data Validation

Data given by the client for the prediction is validated against the DSA (Data Sharing Agreement).

3.9 Prediction Data Inserting into the Database

After prediction data validation is done, then it is stored into the MongoDB.

3.10 Clustering of data to be predicted

The data set is clustered as per the model created in the training process. Each cluster is mention against the data point and then stored in the path 'Clustered Data\predict\clustered_cement.csv'

3.11 Prediction

Each model is called for a particular cluster and then the prediction is made.

3.12 Results

Results are stored into the database and the same is returned to the client.

3.13 Deployment:

We will be deploying the model to Heroku. This is a workflow diagram for the concrete strength prediction.

4 Test Cases

Test case description	Prerequisites	Expected Results
Verify whether the Application URL is accessible to the user	1. Application URL should be defined	Application URL should be accessible to the user
Verify whether the Application loads completely for the user when the URL is accessed	1. Application URL is accessible 2. Application is deployed	The Application should load completely for the user when the URL is accessed
Verify whether user can see input fields on fetching URL	1. Application is accessible	User should be able to see input fields on logging in
Verify whether user can edit input fields	1. Application is accessible	User should be able to edit all input fields
Verify whether user gets Submit button to submit the inputs	1. Application is accessible	User should get Submit button to submit the inputs
Verify whether user is presented with results on clicking submit button	1. Application is accessible	User should be presented with results on clicking submit
Verify whether the results are in accordance to the selections user made	1. Application is accessible	The recommended results should be in accordance to the input given by the user