LOW-LEVEL DESIGN (LLD)

Cement concrete compressive strength predictor

Vishvesh Shirolkar
vish.shirolkar@gmail.com

Contents

1 Introduction

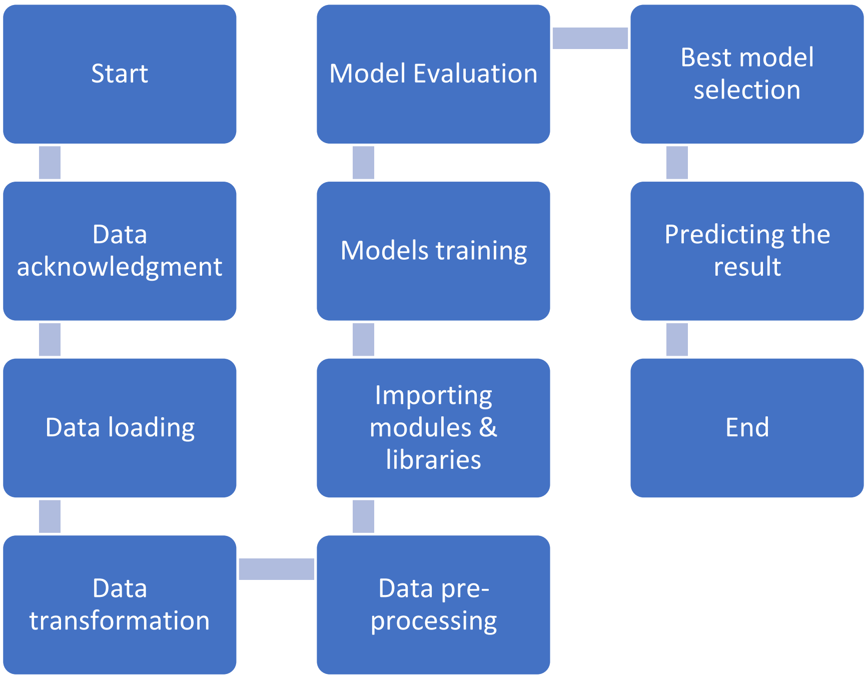
1.1 What is Low-Level Design document?

The goal of low-level design document is to present the logical design of the program code for Cement concrete compressive strength predictor. It describes all the logic behind the program so that it helps others to code from it. It tells about every step that is taken into consideration to build the program.

1.2 Scope

Low-level design (LLD) is elemental level design process which showcases each steps in a program code. With the help of this, one can determine approach of coding, datasets, software architecture, libraries, and modules to use. Hence, it is supporting an SRS(Software Requirement Specification) document and based on that, it helps to design the code for developers.

2 Architecture



3 Architecture description

3.1 Data description

The dataset is taken from Kaggle dataset/repository, which is an open-source database. Data is contained in a csv file. It contains cement mixture ingredients and their amounts in comma separated form. It has 1030 data entries or rows and 9 columns. From the 9 columns, one column is the label column i.e. compressive strength & rest are ingredients: cement, blast furnace slag, fly ash, water, superplasticizer, coarse aggregate, fine aggregate & age of the mixture.

3.2 Data loading

In this, one should load the data from available dataset onto python environment/ workbook. This data should be in excel form or csv form.

3.3 Data transformation

Data should be transformed into csv format if not already done. Once it's done, it should be properly scaled to make it even throughout.

3.4 Data pre-processing

Data should then be pre-processed for training the models. Pre-processing can include tasks like data splitting into training, testing/validation. Any null/void values should be imputed.

3.5 Model building

Pre-processed data is then fitted into various machine learning models. This should contain model from various types like regression, trees, linear/non-linear models etc. so that one should analyse it from different perspective.

3.6 Model evaluation

Once the data is fitted, the trained model should be evaluated based on various metrices. This will give us the best model to choose for given dataset.

3.7 Model selection & optimization

From evaluated models, the model with higher score is selected. Further, the model should be optimized for better results using appropriate conditions/arguments

3.8 Prediction

Once, the model is optimized, the program code is ready to use for prediction of compressive strength.