Concrete Compressive strength Prediction

Architecture Document

Document Version Control

Date issued	Version	Description	Author
01-Nov-2024	1.0	First Draft	Sanket Jagtap

Contents

Abstract

Being one of the most frequently used building materials, the quality of concrete is determined by its compressive strength, which is measured by crushing a concrete cube or a cylinder until it starts cracking and crushed. The pressure at which the concrete cube or a cylinder starts cracking and eventually crushes is called the Concrete compressive strength and is measured in megapascals (MPa). It takes a long period of 28 days to test like this. With the help of Data science and the Machine learning technology, I developed an application, which allows an engineer to determine the strength of a concrete in just a few seconds of time.

1.Introduction

1.1 Why this 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.Technical specifications

2.1 Dataset overview

For training and testing the model, we used the public data set available in Kaggle, “Concrete Compressive Strength Data Set” by Ahiale Darlington.

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

Following is the data dictionary

Name	Data Type	Measurement	Description
Cement	Quantitative	kg in a m3 mixture	Input variable
Blast Furnace Slag	Quantitative	kg in a m3 mixture	Input variable
Fly Ash	Quantitative	kg in a m3 mixture	Input variable
Water	Quantitative	kg in a m3 mixture	Input variable
Superplasticizer	Quantitative	kg in a m3 mixture	Input variable
Coarse Aggregate	Quantitative	kg in a m3 mixture	Input variable
Fine Aggregate	Quantitative	kg in a m3 mixture	Input variable
Age	Quantitative	Days (1~365)	Input variable
Concrete Compressive Strength	Quantitative	megapascals (MPa)	Output variable

2.2 Predicting the Concrete compressive strength

- The web application must be loaded properly for the users without any technical glitches like server timeouts.
- It must display the input fields and the "Predict" button to the users who accessed the application and allow the user to enter the values with respect to the quantities of various raw materials used to build a concrete and its age.
- The user gives the required information.
- Then the application should be able to predict the compressive strength of the concrete based on the information given by the user.

2.3 Logging

We should be able to log every activity done by the user.

- The system should be able to log every step in the program flow.
- System should not be hung even after using so many loggings.
- Logging makes debugging much easier, like we can directly go to that specific line of code, having bugs.
- In this project, logs will be written in the files "development_logs.log" and the "deployment_logs.log" respectively.

2.4 Exception handling

Used exceptions handling to catch the errors, so that they will be recorded in logs and ensures the smooth run, without getting terminated in the middle. Once the run gets completed, we can check the log files for the errors and can take an appropriate debugging action.

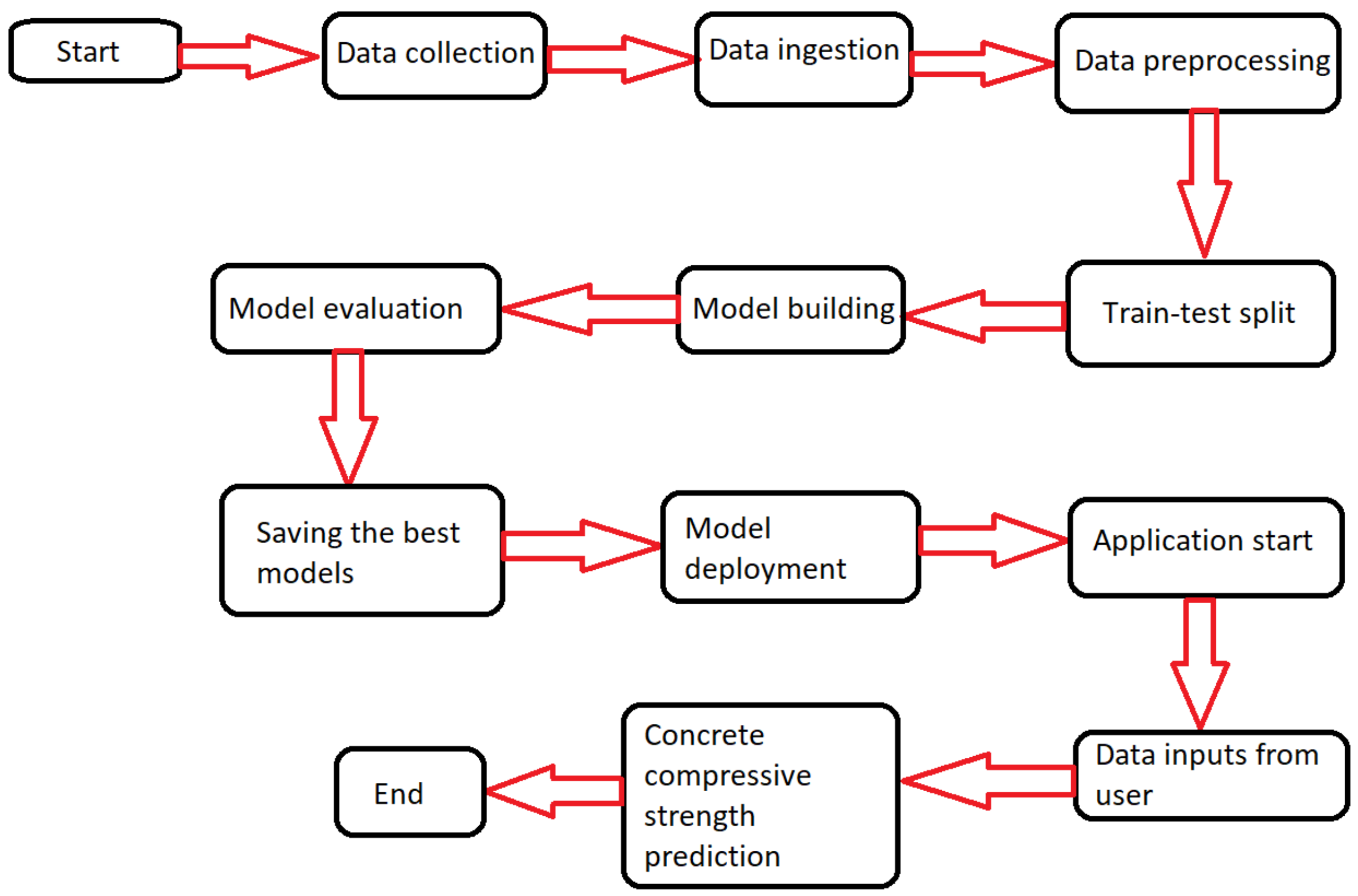
3.Technology stack

Front-end	HTML 5 with CSS styling
Back-end	Python version 3.7, Flask version 2.0.1
Deployment	Koyeb, gunicorn version 20.1.0

4.Proposed solution

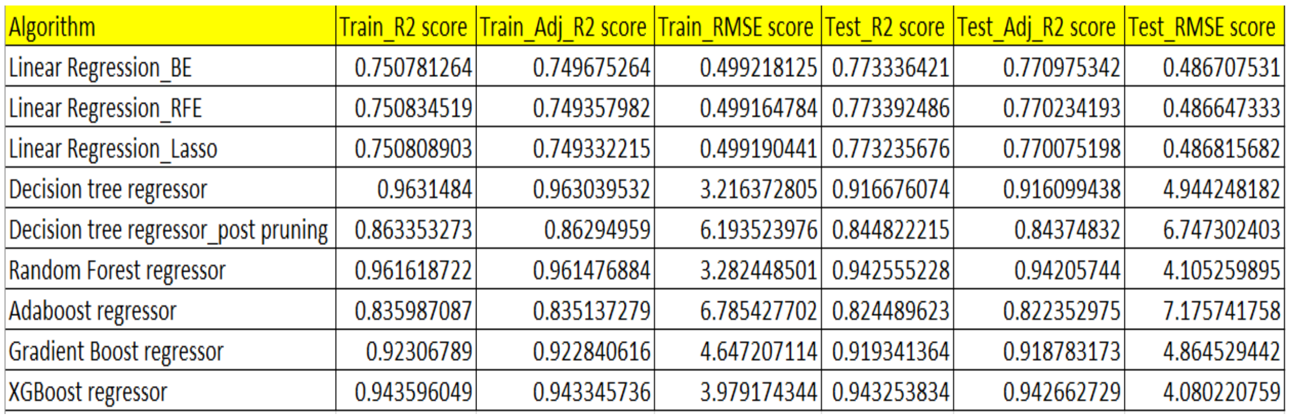
The solution proposed here is a web application, which takes the details of the main ingredients of a concrete specimen which contributes to its compressive strength and those details will be taken by an XGBoost regressor model in the backend, which predicts the compressive strength in MPa and displays in the front-end page to the user.

5.Workflow


Start
Data collection
Data ingestion
Data preprocessing
Train-test split
Model building
Model evaluation
Saving the best models
Model deployment
Application start
Data inputs from user
Concrete compressive strength prediction
End

6.Key performance indicators (KPI)

- Time and workload reduction using the regression models.
- Comparison of the R2 scores and the Adjusted R2 scores of the model on both the training and the testing data.
- Comparison of the RMSE scores of the model on both the training and the testing data.



Algorithm	Train_R2 score	Train_Adj_R2 score	Train_RMSE score	Test_R2 score	Test_Adj_R2 score	Test_RMSE score
Linear Regression_BE	0.750781264	0.749675264	0.499218125	0.773336421	0.770975342	0.486707531
Linear Regression_RFE	0.750834519	0.749357982	0.499164784	0.773392486	0.770234193	0.486647333
Linear Regression_Lasso	0.750808903	0.749332215	0.499190441	0.773235676	0.770075198	0.486815682
Decision tree regressor	0.9631484	0.963039532	3.216372805	0.916676074	0.916099438	4.944248182
Decision tree regressor_post pruning	0.863353273	0.86294959	6.193523976	0.844822215	0.84374832	6.747302403
Random Forest regressor	0.961618722	0.961476884	3.282448501	0.942555228	0.94205744	4.105259895
Adaboost regressor	0.835987087	0.835137279	6.785427702	0.824489623	0.822352975	7.175741758
Gradient Boost regressor	0.92306789	0.922840616	4.647207114	0.919341364	0.918783173	4.864529442
XGBoost regressor	0.943596049	0.943345736	3.979174344	0.943253834	0.942662729	4.080220759

- Feature importance by each model, as shown below

Algorithm	Imp_Features
Linear Regression_BE	['cement', 'blast_furnace_slag', 'fly_ash', 'water', 'superplasticizer', 'age']
Linear Regression_RFE	Index (['cement', 'blast_furnace_slag', 'fly_ash', 'water', 'superplasticizer', 'coarse_aggregate', 'fine_aggregate ', 'age'], dtype='object')
Linear Regression_Lasso	['cement', 'age', 'blast_furnace_slag', 'fly_ash', 'superplasticizer', 'fine_aggregate', 'water']
Decision tree regressor	['cement', 'age', 'water', 'blast_furnace_slag']
Decision tree regressor_post pruning	['cement', 'age', 'water', 'blast_furnace_slag']
Random Forest regressor	['age', 'cement', 'water', 'blast_furnace_slag', 'superplasticizer']
Adaboost regressor	['cement', 'age', 'water', 'blast_furnace_slag', 'superplasticizer', 'fine_aggregate ', 'coarse_aggregate']
Gradient Boost regressor	['age', 'cement', 'water', 'blast_furnace_slag']
XGBoost regressor	['age', 'cement', 'water', 'fly_ash', 'superplasticizer', 'blast_furnace_slag']

- Low water to cement ratio is proportional to the strength of the concrete.