High Level Design (HLD)

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 HLD

Contents

1. Abstract

Concrete, a fundamental material in construction, requires precise compressive strength prediction for structural integrity. Traditional methods often involve time-consuming experiments and complex calculations. This study proposes a machine learning-based approach to accurately predict concrete compressive strength.

Concrete is the most common and oldest material for construction worldwide. Infrastructure and construction are considered as extremely sensitive domains regarding the safety issue. So, since concrete being the main component or material for the construction it would be important to measure the strength or the power of the concrete using some reliable methods. Concrete is basically a composite material composed of various base materials like cement, water, Coarse Aggregate, Fine Aggregate, and some other components. Compressive strength of concrete is measured using a conventional crushing test on a concrete cylinder. Basically, it takes 28 days of time.

The results demonstrate that machine learning models can significantly outperform traditional methods in predicting concrete compressive strength. The selected model achieved high accuracy and efficiency, offering a promising solution for optimizing concrete production processes and ensuring structural safety. This research contributes to advancements in the construction industry by providing a reliable and time-efficient tool for predicting concrete properties.

2. Introduction

2.1 Why this High-Level Design Document?

The purpose of this High-Level Design (HLD) Document is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding, and can be used as a reference manual for how the modules interact at a high level.

The HLD will:

- Present all of the design aspects and define them in detail
- Describe the user interface being implemented
- Describe the hardware and software interfaces
- Describe the performance requirements
- Include design features and the architecture of the project
- List and describe the non-functional attributes like: o Security
  - o Reliability
  - o Maintainability
  - o Portability
  - o Reusability
  - o Application compatibility
  - o Resource utilization
  - o Serviceability

2.2 Scope

The HLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HLD uses non-technical to mildly-technical terms which should be understandable to the administrators of the system

2.3 Definitions

Database	Collection of all the data in tabular format at one place
Model	A Machine Learning Predictive Model
Paas	Platform as a service
flask	Flask is a micro web framework written in Python
heroku	Heroku is a cloud platform as a service supporting several programming language

3. General Description

3.1 Product Perspective

The Concrete Compressive Strength Predictor is basically a Machine Learning based regression model which can predict compressive strength of concrete from all the materials used to make the concrete, some of those are: cement, water, Coarse Aggregate, Fine Aggregate, fly ash.

3.2 Problem Statement

The quality of concrete is determined by its compressive strength, which is measured using a conventional crushing test on a concrete cylinder. The strength of the concrete is also a vital aspect in achieving the requisite longevity. It will take 28 days to test strength, which is a long period. So, what Will we do now? We can save a lot of time and effort by using Data Science to estimate how much quantity of which raw material we need for acceptable compressive strength

3.3 Approach

The classical machine learning tasks like Data Exploration, Data Cleaning, Feature Engineering, Model Building and Model Testing. Try out different machine learning algorithms that's best fit for the above case.

3.4 Technical Requirements

Here "Python 3..8" is used, and some specific versions are used for the Machine Learning process, which is deployed on the server.
All the module requirements are mentioned in the "requirements.txt".
To get and run on your system:

- ➔ Open prompt on the system
- ➔ cd your-project-directory
- ➔ gitclone https://github.com/Utkarsh2108/Concrete-Compressive-Strength-Prediction.git
- ➔ pip install -r requirements.txt

➔ python app.py

3.5 Data Requirements

For the model training purpose, the data is taken from the Public “kaggle”

Data description
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

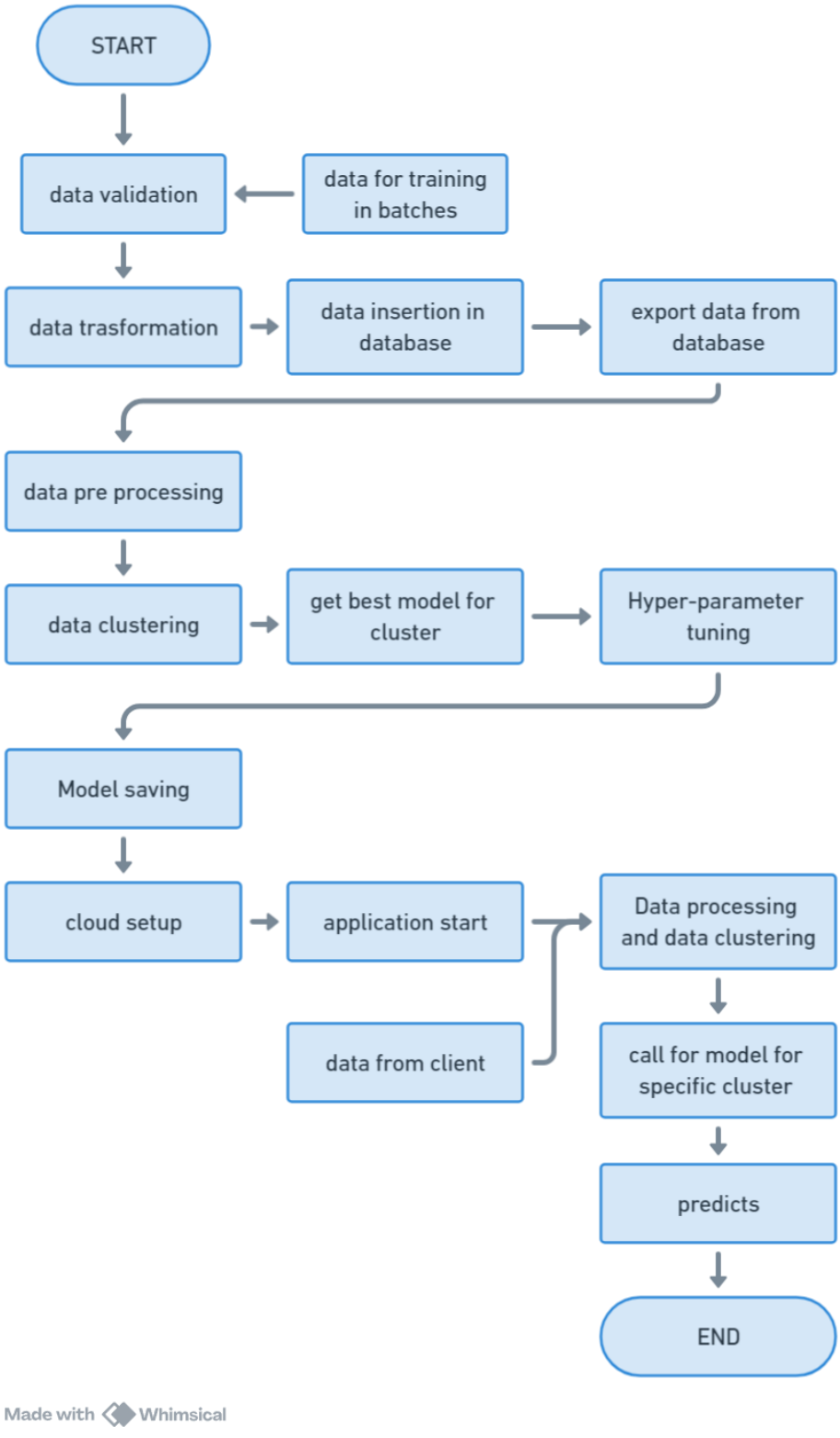
Source :

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

3.6 Tools used and technology used

4. Design Details

4.1 Process flow



4.2 Event Logs

An event log is a file that records significant actions or occurrences that a computer system recognizes. Event logs can be part of the operating system or specific to an application

Logging is a very essential part of end-to-end systems. By the logs we can make sure what is going on internally and if the system crashed then why it crashed. Here logging is used in every part of the project so the perfect scene can be imagined as the exact flow of the program is!

4.3 Error Handling

Error handling is a process that helps systems and applications respond to unexpected situations, such as missing or corrupted data

Here in every module the exception handling is done, so that it does not cause system failure or system error.

In case of any warnings or errors occurring, it would be noted in the logs.

5. Performance

This Machine Learning based Concrete compressive strength predictor application is completely based on the data provided to the model like other Machine Learning models. In the model itself Cross validation and Hyperparameter tuning is done at its best. So that it should be as accurate as possible. So that it would not mislead
the clients / users.
And also, as the performance is focused then, it should be noted that as the time goes, for the model to perform its best and stay updated with the data it should be retrained as well.

5.1 Reusability

The code written and the components used have an ability to be reused without any problem.

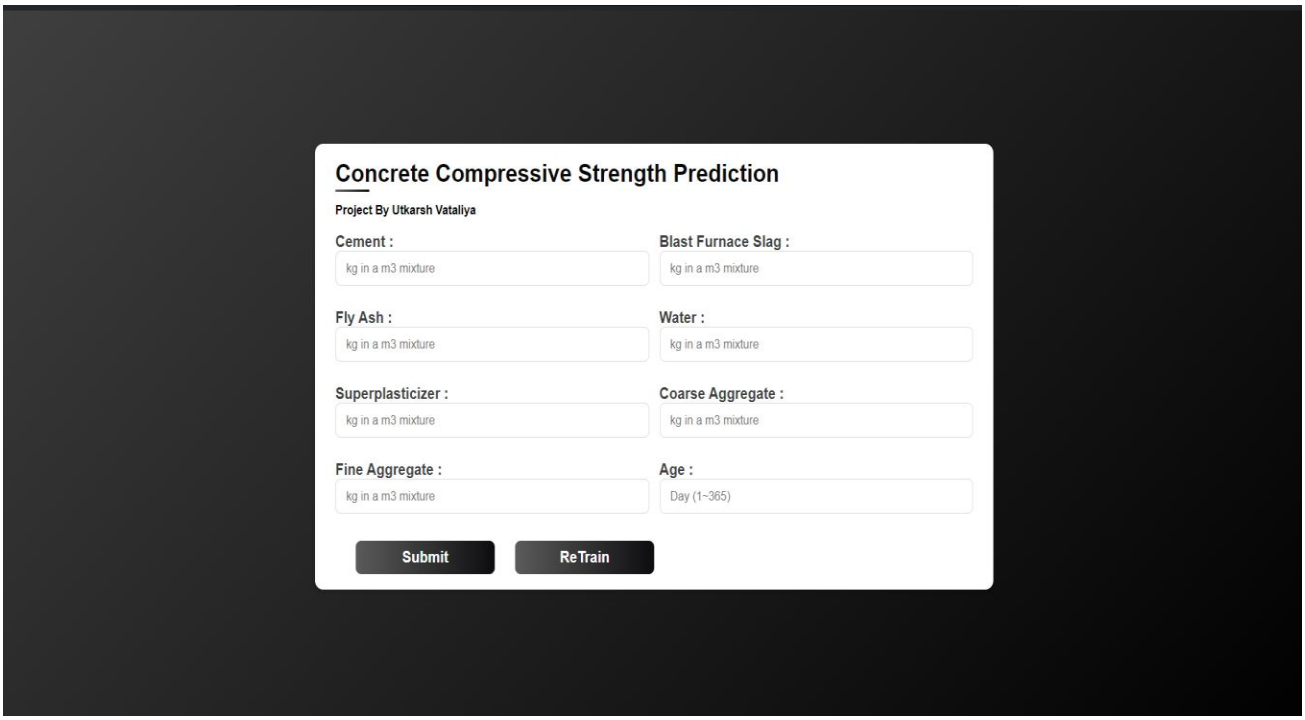
5.2 Application compatibility

The different components or modules of this project use python as their interface between them. Each component has its own task to perform, and it is the job of the python version to ensure proper transfer of the information.

5.3 Deployment

The application deployment with the Flask framework and also front-end with HTML5 and CSS is done on the Heroku Server.

6 User Interface



7. Conclusion

Traditionally, the strength of concrete is also a vital aspect in achieving the requisite longevity. It will take 28 days to test strength, which is a long period. So using the Machine Learning based Compressive strength prediction approach is obviously a better choice. If updated with more and good quality of data, it would surely be a better alternative so far.