CONCRETE COMPRESSIVE STRENGTH PREDICTION

HIGH LEVEL DESIGN

REVISION No. : 1.0
LAST DATE OF REVISION : MAY 2ND,2022

DOCUMENT VERSION CONTROL

DATE	VERSION	DESCRIPTION	AUTHOR
02-05-2022	1	INITIAL HLD	MOHD. USAMA

TABLE OF CONTENTS

Abstract

In the present world, concrete is one of the most widely used construction materials. This can be due not alone to the large choice of applications that it offers, however, besides, its behavior, strength, affordability, durability, and flexibility play vital roles. Therefore, constructing-building works have faith in concrete as a secure, strong, and simple object. Structures have to face many face horizontal and vertical stresses due to dead load, live loads, earthquakes, wind and snow loads etc. We see many cases around the world where people lose their life because of lack of enough strength of concrete to bear such loads.

Hence, a concrete of enough strength to bear these loads is very necessary to avoid such disasters. The concrete strength depends upon the choice and quantity of it's components. It achieves around 99% of it's strength in 28 days. So to predict the strength of our concrete sample we have to wait for that much time which is not feasible. Hence a good data driven system for predicting concrete strength can help to design a concrete of required strength very quickly by removing the time barrier.

1 Introduction

1.1 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 before 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:
    - -Security
    - -Reliability
    - -Maintainability
    - -Portability
    - -Reusability
    - -Application compatibility
    - -Resource utilization
    - -Serviceability

1.2 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 General Description

2.1 Product Perspective

This project will help us to predict the strength of concrete based on it's components and time period.

2.2 Problem Statement

The goal is to create a ML based solution for the prediction of the compressive strength of concrete

2.3 Proposed Solution

The solution is to build a ML based Regression model for the prediction. We have many regression based machine learning algorithms like Linear Regression, Decision Trees, SVR, XGBoost, etc. for this purpose. But before moving ahead with any ML algorithm, we have to preprocess the raw data for handling missing values, duplicate values, categorical data encoding , outliers removal, etc.

2.4 Technical Requirements

In this project we are having a set of requirements and they are given below

a. Model should be exposed through API or User Interface, so that anyone can test model.

b. Model should be deployed on cloud (Azure, AWS, GCP).

c. Cassandra database should be integrated in this project for any kind of user input.

2.5 Data Requirements

Data Requirement completely depend on our problem.

a. For training and testing the model, we are using Concrete Compressive Strength dataset from UCI machine learning repository.

b. Dataset Link: [UCI Machine Learning Repository: Concrete Compressive Strength Data Set]

2 General Description

2.5 Tools Used

This project will help us to predict the strength of concrete based on it's components and time period.

2.6 Technical Requirements

The Concrete Strength Prediction must be user friendly, errors free and users should not be required to know any of the working.

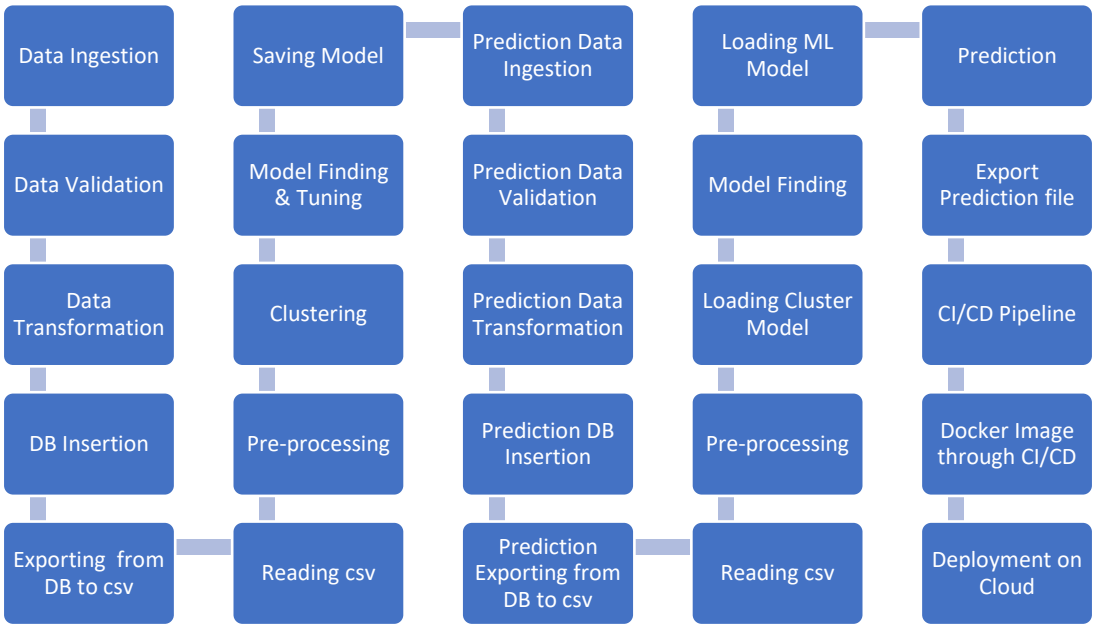
2.7 Assumptions

It is assumed that all the aspect of this project have the ability to work together in the way designer is expecting.

3 Design Details

3.1 Process Flow

The process flow diagram of Training ,Prediction and Deployment phase is as below:



3.2 Logging

The In this Project we are logging every process so that the user will know what process is running internally. We have designed logging in such a way that debugging will be an easy task .

3.3 Error Handling

It is We have designed this project in such a way that, at any step if error occur then our application should not terminate rather it will catch that error and display that error with proper explanation as to what went wrong during process flow.

4 KPI

4.1 Reusability

This project's code Is written in such a way that it could be reused without any kind of problem.

4.2 Application Compatibility

The different module of this project are using Python as an interface between them. Each module will have it's own job to perform and it is the job of the Python to ensure the proper transfer of information.

4.3 Resource Utilization

In this project, when any task is performed, it is likely that the task will use all the processing power available in that particular system until it's job finished.

4.4 Data Deployment

I am deploying my model to AWS, Azure, GCP and Heroku using DockerHub and Circle CI.

5 Conclusion

The Spam-Ham Classifier model will classify the message type in prior so that people can be safe from frauds, unwanted messages and it will also save the time of people by classifying the message which are important.