Low Level Design (LLD)

Concrete Compressive Strength Prediction

By

ANNAMALAI SUBRAMANI

Revision Number: 1.0
Last date of revision: 05/02/2024

Document Version Control

Date Issued	Version	Description Author
05/02/2024	1	Initial LLD - V1.0 Annamalai

Contents

Abstract

The LLD document for the Concrete Compressive Strength Prediction system outlines its architecture and implementation. It details the chosen regression algorithm, training methodologies, and validation techniques. API design considerations cover endpoints, input validation, error handling, and security measures. Deployment strategies include environment specifications and containerization details. Rigorous testing methods ensure the reliability of the model and API. Logging mechanisms and monitoring metrics aid debugging and system health assessment. Security measures address data privacy, and model explainability techniques enhance transparency. The document serves as a concise guide for stakeholders involved in system development.

1 Introduction

1.1 Why this Low-Level Design Document?

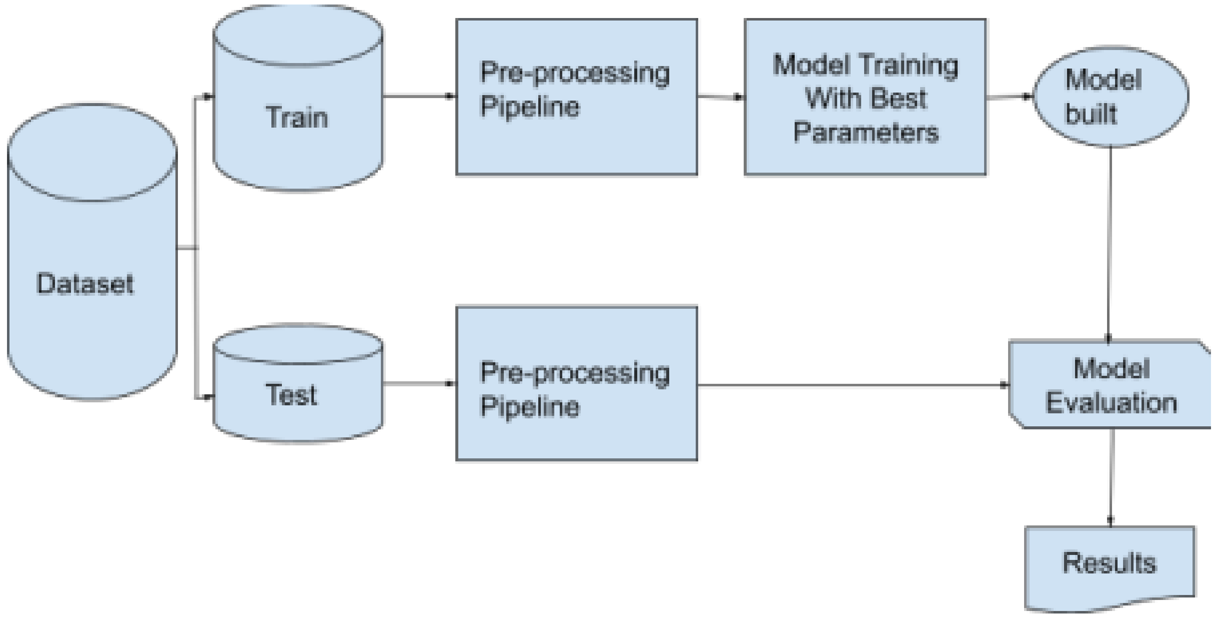
The Low-Level Design (LLD) document for the Concrete Compressive Strength Prediction system is meticulously crafted to unveil the intricate details of the system's internal mechanisms. Its primary objective is to offer a profound insight into the architecture, features, interfaces, operational constraints, and responses to external stimuli within the system. This document is of paramount importance for both stakeholders and developers, serving as a formal proposal for approval from higher management. It extensively elucidates the internal design aspects of the application, concentrating on the interactions and functionalities of various components related to concrete compressive strength prediction.

1.2 Scope

The Low-Level Design (LLD) process signifies a component-level design approach characterized by systematic refinement. This pivotal process contributes to the design of data structures, establishment of software architecture, and development of source code. The overall organization of data, initially defined during requirement analysis, undergoes further refinement in the LLD process. This section defines the scope of the LLD process for the Concrete

Compressive Strength Prediction system, laying the groundwork for subsequent design details and providing a comprehensive framework for the system's intricate structure.

2 System Architecture



3 Machine Learning Components

3.1.1 Data Gathering

The Concrete Compressive Strength Prediction project relies on a dataset obtained from Kaggle.

3.1.2. Tools Used

- Programming Language: Python 3.8, with frameworks such as numpy, pandas, scikit-learn, and other relevant modules for model development.
- Integrated Development Environment (IDE): Vscode.
- Visualization Tools: Seaborn and components of Matplotlib.
- Version Control: GitHub for code versioning.
- Deployment: Netlify is employed for project deployment.

3.1.3. Data Description

The Concrete Compressive Strength Prediction dataset is extensive, comprising

1030 rows and 9 columns.

3.2 Preprocessing Pipeline

The Preprocessing Pipeline prepares the dataset for model training by handling missing values and scaling features.

Imputation: Define the strategy for handling missing values.
Scaling: Describe the method used for feature scaling.
● Cleaning and transforming raw data into a format suitable for model training.
● Ensuring data consistency and compatibility with the chosen machine learning algorithm

3.3 Model Training & Hyper-parameter Tuning

The Model Training & Hyper-parameter Tuning Component focuses on training a machine learning model to predict concrete compressive strength.

Algorithm Selection: Specify the chosen classification algorithm (e.g., Random Forest, Linear Regression).
Hyper-parameter Tuning: Define the hyper-parameters to be tuned.
Cross-validation: Describe the cross-validation strategy.

- Training the machine learning model on the preprocessed dataset.
- Fine-tuning hyper-parameters to enhance model performance.

3.4 Model Evaluation

The Model Evaluation Component for Concrete Compressive Strength Prediction focuses on assessing the effectiveness of the trained model in predicting concrete strength. Key components of this evaluation include:

Metrics:Define metrics like Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared (R2) to quantitatively measure the accuracy and precision of concrete compressive strength predictions.

Cross-validation Evaluation:Specify a cross-validation strategy, outlining how the model's performance will be assessed across various data subsets to ensure robustness and generalizability in predicting concrete compressive strength.

- Evaluating model performance on both training and validation sets.
- Calculating and reporting relevant metrics to assess prediction quality.

4 API Design

4.1 Front End Application

Develop a front end application using Flask for API and HTML as user-interface to accept input data to the model

4.2 API Endpoints

Expose API endpoints for the model to consume requests from users and revert the prediction as an output.

5 Model Deployment

5.1 Streamlit

5.1 Streamlit Deployment

Prepare Your Streamlit App:

Ensure your Streamlit application is developed and ready for deployment.

Save or export your Streamlit app code in a format compatible with the deployment platform of your choice.

Choose Deployment Platform:

Select a platform for deploying your Streamlit app, such as Streamlit Sharing, Netlify, or Vercel.

Follow platform-specific instructions for straightforward deployment without the need for AWS or Heroku.

Configuration and Dependencies:

Adjust configuration settings as needed on your chosen platform and ensure all dependencies are correctly specified.

Security Considerations:

Implement necessary security measures depending on the chosen deployment platform, such as access controls and SSL encryption.

Monitoring and Logging:

Explore monitoring and logging options provided by the platform to track usage, identify potential issues, and enhance performance.

6 Testing

6.1 Unit Testing

Unit testing is crucial to ensure the correctness of individual components in the concrete compressive strength prediction.

6.1.1 Unit Test Cases

Test Case Description	Pre-Requisites	Expected Results
Verify whether the User Interface URL is accessible to the user.	1. User Interface URL should be defined.	User Interface URL should be accessible to the user.
Verify whether the User Interface loads completely for the user when the URL is accessed.	1. User Interface URL is accessible. 2. User Interface is deployed.	The User Interface should load completely for the user when the URL is accessed.
Verify whether user is able to edit all input fields.	1. User Interface is accessible.	User should be able to edit all input fields.

7 Conclusion

This Low-Level Design document offers a comprehensive insight into the intricacies of the concrete compressive strength prediction machine learning project. Each component is meticulously crafted to fulfill specific roles, collectively contributing to a robust and efficient prediction solution.

The outlined design decisions and specifications form the foundation for the successful implementation of the system. From data gathering and preprocessing to model training, evaluation, and deployment, every aspect is carefully considered to ensure the accuracy and reliability of the concrete compressive strength prediction model.

By leveraging tools such as Python and relevant machine learning libraries, the project aims to achieve optimal performance across diverse scenarios. The integration of suitable algorithms and frameworks, coupled with data storage considerations, adds versatility and scalability to the system.

Deployment strategies, including considerations for cloud services or other hosting platforms, signify a commitment to providing users with easy access to the concrete compressive strength prediction system. Security measures, such as user authentication and encryption, are implemented to safeguard both the model and user data.

Documentation, including code documentation and a user guide, ensures clarity for both developers and end-users. Robust unit testing procedures are established to validate critical functions, enhancing the overall reliability of the system.

In conclusion, this Low-Level Design document serves as a roadmap for the development and evolution of the concrete compressive strength prediction system. It aims to deliver accurate predictions while maintaining transparency and security, laying the groundwork for adaptability to future advancements in the field of machine learning and prediction.