**Architecture**

**CEMENT CONCRETE STRENGTH PREDICTION**

Revision Number – 1.0

Last Date of Revision – 22-05-2023

**SHAHIN ANJUM**

**Liabaries Requirment**

Flask

seaborn

pandas

numpy

scikit-learn

**Content**

# **Abstract -----------------------------------------------------------------------------------4**

**Introduction ---------------------------------------------------------------------------- 4**

**Why the Architecture Design Documents? ------------------------------ 4**

**1.Architecture ------------------------------------------------------------------------ 4**

**2.Architecture Design ------------------------------------------------------------- 5**

**2.1 Data gathering from main source --------------------------------------- 5**

**2.2 Data description -------------------------------------------------------------- 5**

**2.3 Upload data inti Cassandra ----------------------------------------------- 5**

**2.4 Export data from database ----------------------------------------------- 5**

**2.5 Data pre-processing ------------------------------------------------------- 5**

**2.6 EDA ------------------------------------------------------------------------------ 6**

**2.7 Modelling ---------------------------------------------------------------------- 6**

**2.8 UI integration ---------------------------------------------------------------- 6**

**2.9 Data from user --------------------------------------------------------------- 6**

**2.10 Data validation -------------------------------------------------------------- 6**

**2.11 Rendering the results ---------------------------------------------------- 6**

**2.12 Deployment ------------------------------------------------------------------ 6**

**Abstract**

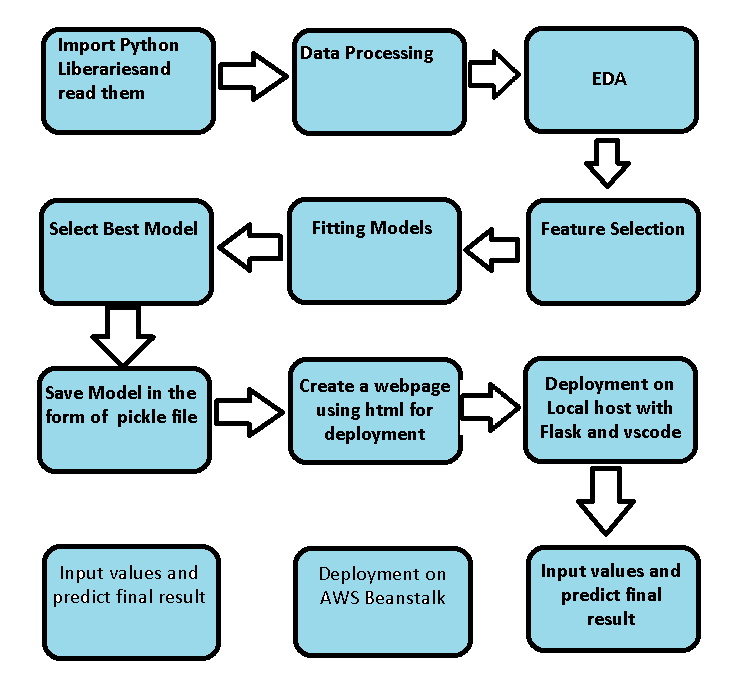
In this paper, an intelligent approach based on the machine learning technique is proposed for predicting the [compressive strength](https://www.sciencedirect.com/topics/materials-science/compressive-strength) of concrete. This approach employs the adaptive boosting algorithm to construct a strong learner by integrating several weak learners, which can find the mapping between the input data and output data. The weak learner whose predicting error is small will have a larger weight in the entire system, thus the overall accuracy of the strong learner will be enhanced. A total of 1030 sets of concrete compressive strength tests is collected to train and test the learners, in which the concrete mixture components (e.g., coarse/fine aggregates, cement, water, additive, etc.) and the curing time are set as the input data while the compressive strength value is set as the output data.

**1 Introduction**

Concrete is the most widely used building material around the world due to its various advantages over other materials, e.g., integrity, durability, modularity, economy, etc. For better understanding the behaviour of concrete-made structures under external loadings and developing corresponding design methodologies, it is of great importance to study the mechanical properties of concrete. Among the various indices of properties of concrete, compressive strength is the most fundamental one since it is directly related to the safety of the structures, and it is required in the performance determination of the structures during the whole life-cycle, from new structural design to old structural assessment. However, as it is known to all, concrete is made up of different components, e.g., coarse/fine aggregates, cement pastes, additional mixtures, etc., and these components are randomly distributed over the entire concrete matrix.

Why this Architecture Design documentation?

The main objective of the Architecture design documentation is to provide the internal logic understanding prediction code. The Architecture design documentation is designed in such a way that the programmer can directly code after reading each module description in the documentation.

1 Architecture 

**2 Architecture design**

This project is to create an interface for the user to know the **Concrete compressive strength**, in addition to this, in need of getting the real time project experience we are importing the gathered data into our own database and then start the project from the scratch.



**2.1 Data gathering from main source**

The data for the current project is being gathered from kaggle, the link to the data is:

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

**2.2 Data description**

The Census Income dataset has 1030 entries. Each entry contains the following information about an individual:

Cement, Blast, Fly Ash, Water, Superplasticize, Coarse Aggregat, Fine Aggregat, Age and Concrete compressive strength.

**2.3 Import data into Cassandra**

Created an api for the upload of the data into the Cassandra database, steps performed are:

* Connection is made with the database.
* Created a database with name Energy Efficiency.
* Cqlsh command is written for creating the data table with required parameters.
* And finally, a cqlsh command is written for uploading the dataset into data table by bulk insertion.
* .

2.4 Export data from database

In the above created api, the download url is also being created, which downloads the data into a csv file format.

2.5 Data pre-processing

Steps performed in pre-processing are:

* First rename all columns
* The data types are being checked and found all columns are of type integer.
* Checked for null values as there are no null values.
* Checking outliers as looking box plot we don’t need to worry about outliers.
* Scaling is performed for required data.
* And the data is ready for passing to the machine learning algorithm.

2.6 EDA

I first plot bar graph of targets columns with all independent columns. By these graphs I concluded many things which is written in EDA Jupyter notebook.

**2.7 Modelling**

The pre-processed data is then visualized, and all the required insights are being drawn. Although from the drawn insights, the data is randomly spread but still modelling is performed with different machine learning algorithms to make sure we cover all the possibilities. And finally, as expected random forest regression performed well.

**2.8 UI integration**

Both CSS and HTML files are being created and are being integrated with the created machine learning model. All the required files are then integrated to the app.py file and tested locally. Note I copy the code from internet and make the CSS and HTML File.

**2.9 Data from user**

The data from the user is retrieved from the created HTML web page.

**2.10 Data validation**

The data provided by the user is then being processed by app.py file and validated. The validated data is then sent for the prediction.

**2.11 Rendering the results**

The data sent for the prediction is then rendered to the web page.

**3 Deployment**

I deployed on AWS Beanstalk. Link is:

6

**ScreenShot of the App Interface which I will deploy**

