Detailed Project Report

CONCRETE COMPRESSIVE STRENGTH PREDICTION
USING
RANDOM FOREST REGRESSOR AND LINEAR REGRESSION

DOMAIN: INFRA

Submitted by: Piyush Sharma

Date: 17/10/2022

Contents:

1. Introduction

The main purpose of the documentation is to add necessary details of the project and provide a description of the machine learning model and the written code. This also provides a detailed description of how the entire project has been designed end-to-end.

1.1 Objective

The concrete compressive strength prediction system is a machine learning-based model with the objective to predict the compressive strength of concrete based on the different quantities of materials used. Based on predictions we can determine the best combination of quantities of each material to achieve the best result.

1.2 Benefits

- Eliminates the use of crushing tests multiple times.
- Saves the expense of running and maintaining test equipment.
- Saves time i.e. conventionally 28 days.
- Avoids wastage of raw material used in testing.

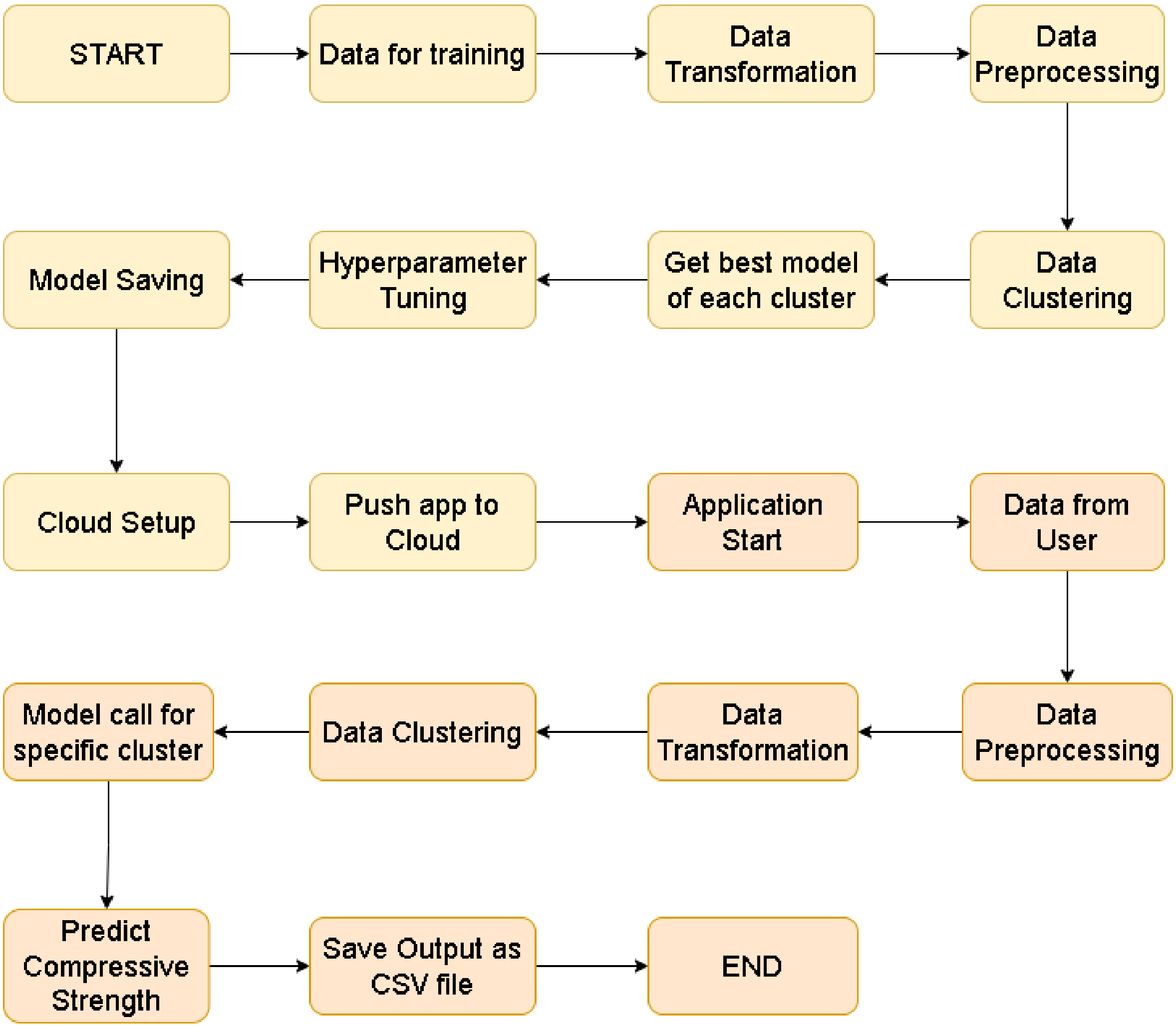
1.3 Technical Requirements

The system doesn’t require much computing power so, the user will be able to run the system on a basic computer/laptop with an internet connection. Users should have a basic understanding of input variable fields.

1.4 Tools Used

- Programming Languages : Python, HTML, CSS
- Libraries : Pandas, NumPy, scikit-learn, matplotlib
- API Framework: Flask
- IDE : VSCode
- Deployment: Heroku, Docker, GitHub

2. Architecture



2.1 Detailed Architecture

Data Description

Given below are the variable name, type, measurement unit, and description. The order of data in the table corresponds to the placement of data in the file.

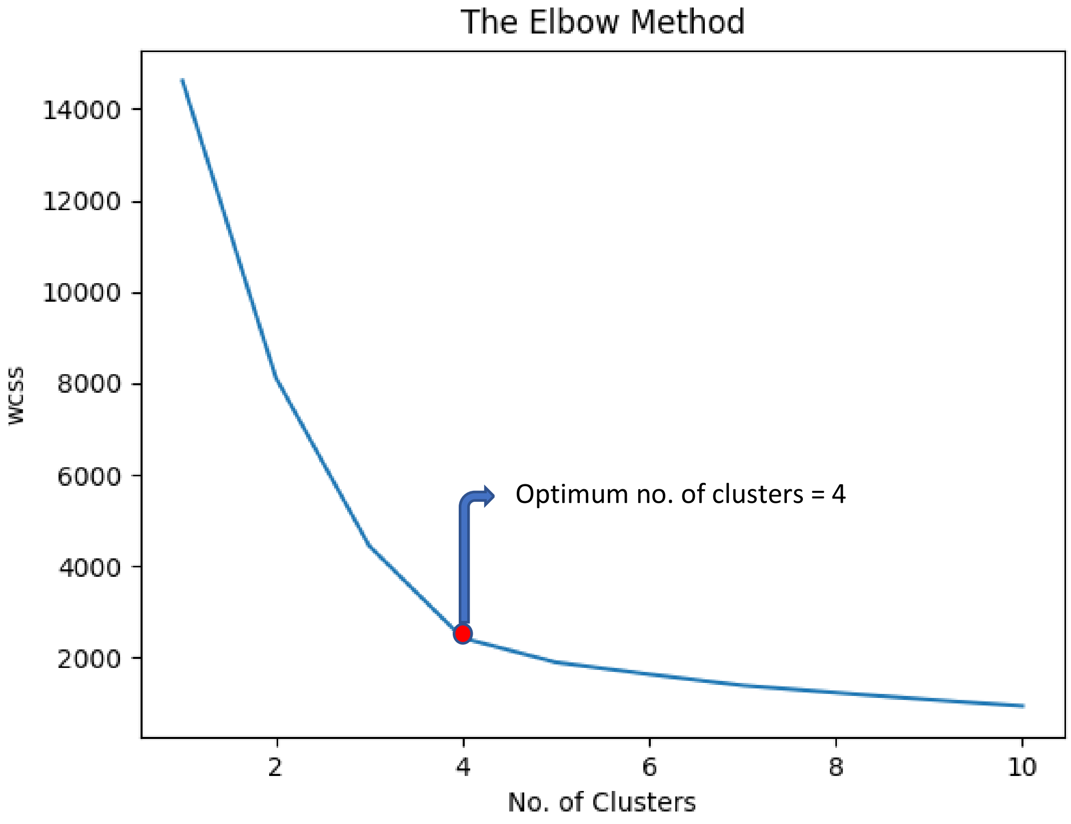
Name	Data Type	Measurement	Description
Cement (component 1)	quantitative	kg in a m3 mixture	Input Variable
Blast Furnace Slag (component 2)	quantitative	kg in a m3 mixture	Input Variable-- Blast furnace slag is a nonmetallic coproduct produced in the process. It consists primarily of silicates, aluminosilicates, and calcium-alumina-silicates
Fly Ash (component 3)	quantitative	kg in a m3 mixture	Input Variable- it is a coal combustion product that is composed of the particulates (fine particles of burned fuel) that are driven out of coal-fired boilers together with the flue gases.
Water (component 4)	quantitative	kg in a m3 mixture	Input Variable
Superplasticizer (component 5)	quantitative	kg in a m3 mixture	Input Variable-- Superplasticizers (SP's), also known as high range water reducers, are additives used in making high strength concrete. Their addition to concrete or mortar allows the reduction of the water to cement ratio without negatively affecting the workability of the mixture, and enables the production of self-consolidating concrete and high performance concrete
Coarse Aggregate (component 6)	quantitative	kg in a m3 mixture	Input Variable-- construction aggregate, or simply "aggregate", is a broad category of coarse to medium grained particulate material used in construction, including sand, gravel, crushed stone, slag, recycled concrete and geosynthetic aggregates
Fine Aggregate (component 7)	quantitative	kg in a m3 mixture	Input Variable—Similar to coarse aggregate, the constitution is much finer.
Age	quantitative	Day (1~365)	Input Variable
Concrete compressive strength	quantitative	MPa	Output Variable

Data Preprocessing

- Check for null values
- Impute missing values using KNN Imputer
- Remove columns with zero standard deviation
- Transform features using log transformation
- Scale features using StandardScaler

Data Clustering

- Kmeans clustering is used to create clusters of preprocessed data.
- The idea behind using clustering is to use different models on each cluster.
- The optimum number of clusters is determined by using the elbow plot's "Knee Locator" function.
- After training Kmeans on preprocessed data the model is saved.



Model Building

- After clusters are created, we find the best model for each cluster.
- We are using two algorithms that are Random Forest Regressor and Linear Regression for each cluster.
- For each cluster, algorithms will be passed with the best parameters which are derived by GridSearchCV.
- We will calculate the r squared error score (r2 score) of each model and the model with the best r2 score will get selected.
- Similarly the models will be saved for each cluster that will be used in prediction.

Receiving data from users

- Users have to provide data in CSV format for training and prediction.

Data Clustering

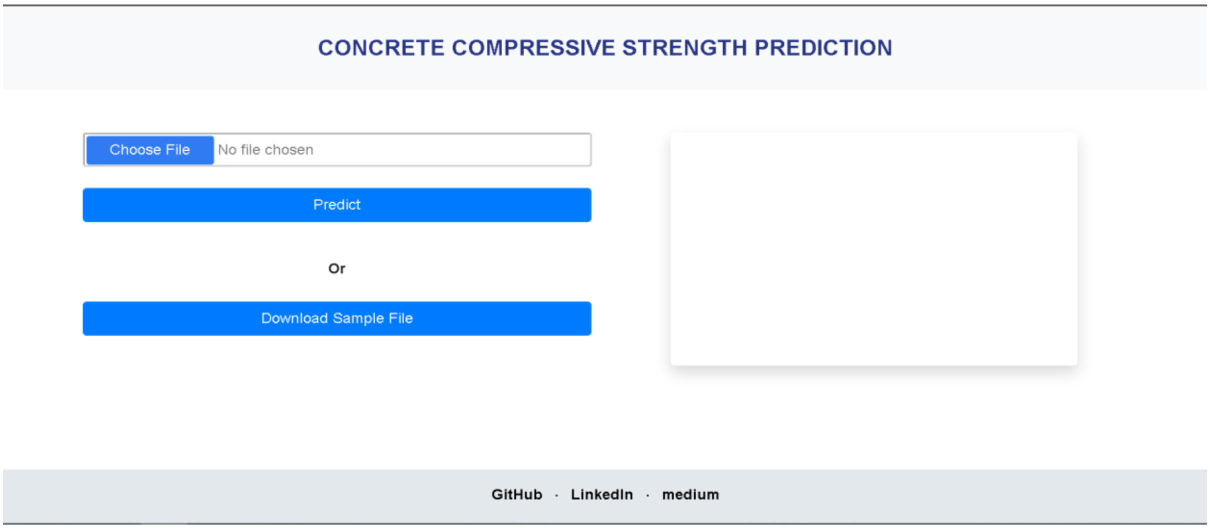
- The Kmeans model saved during training will be loaded again to predict the cluster of user's data.

Model call for specific cluster

- Based on cluster number, the respective model will be loaded and will be used to predict the data for that cluster.

Deployment

- We will be deploying the model on Heroku cloud platform.


CONCRETE COMPRESSIVE STRENGTH PREDICTION
Choose File
No file chosen
Predict
Or
Download Sample File
GitHub · LinkedIn · medium

3. Event log

The system logs every event so that the user will know what process is running internally.

Initial Step-By-Step Description

- The system identifies at what step logging is required.
- The system logs each and every system flow.
- The logging system will log every function entry point and result (successful/unsuccessful) resulting in easy debugging.

4. Error Handling

Point of program failure and exception message will be displayed.

5. Performance

Concrete compress strength prediction is used to predict the strength of concrete with different combinations of materials.

6. Reusability

The code written and the components used have the ability to be reused with no problem.

7. Application Compatibility

The different components for this project will be using Python as an interface between them. Each component will have its own task to perform, and it is the job of Python to ensure the proper transfer of information.

8. Resource Utilisation

When any task is performed, it will likely use all the processing power available until that function is finished.

9. Deployment

The system is containerized using Docker and deployed on Heroku using GitHub actions.

10. Conclusion

Hence, the concrete compressive strength prediction system will predict the compressive strength of concrete and will be able to determine the best set of approximate quantities of materials that the user can use to determine combinations by which he/she will be able to make the product with good strength.

Frequently Asked Questions

1. What is the source of data?

The data for training is provided by the client in multiple batches and each batch contains multiple files.

2. What is the type of data?

The data was of a quantitative type

3. What is the complete flow you followed in this project?

Refer to Page no for a better understanding.

4. How logs are managed?

We used a custom logging method. Different logs were maintained at different steps like data insertion log, model training log, model prediction log, file handling log, and so on.

5. What are the techniques you have used for data pre-processing?

- Removing unwanted attributes.
- Visualising relation and correlation between dependent and independent features.
- Log transform and scale data
- Clustering

6. How training is done and what models are used?

The data was preprocessed and divided into training and validation sets. Algorithms like K-means clustering, Random Forest Regressor, and Linear Regression were used. The best model for each cluster was found and saved accordingly.

7. How the result is predicted?

The same life cycle is repeated on the dataset on which the prediction is to be done. The model is then loaded and the output is saved as csv file.

8. What are the different stages of deployment?

- The project folder must contain Dockerfile, workflows folder, and Procfile in order to deploy the container to the Heroku platform.
- Add Heroku secret keys to GitHub actions secrets.
- Push code to the GitHub repository.
- Connect GitHub account to Heroku account.
- Search for repository.
- Deploy to Heroku.