Predicting Compressive Strength of Concrete using IBM Watson Machine Learning

1. Introduction

1.1 OVERVIEW

Every data scientist has to build domain knowledge in every field because we have to tackle each problem that is probably faced by the world. If you don't know what domain knowledge is? it is the knowledge of a specific field or having specialization in any field. we can also say that it is a piece of general knowledge, so as a data scientist we probably solve the real-world problems which were based on machine learning with the domain knowledge of the specific problem.

As a data scientist, we generally faced many real-life problems, like- some social issue, construction, etc.. we have to solve this type of problem using machine learning techniques, let's we take an example of construction, what will you refer the word construction? it is the art and science to form objects, systems, or organizations. what will you imagine from the word construction is that mega buildings, machines, material, etc… but you know that what is used to build these mega buildings, for construction we use material, cement, iron rods, etc.. where the material is a most important part of building making.

1.2 PURPOSE

” Use the best possible materials, and reveal the quality of those materials and the craftsmanship of their assembly “

The meaning of these beautiful words is that if we use the best quality material then you probably be the quality constructor. We can relate this quote with our example, we discuss above that certain things are used for making mega buildings but the material is most important. Material which is nature and man mad we will talk about man-mad material, in this concrete is most important man-mad material for building. concrete is made up of three basic components **Water, aggregate, and portland cement.** We know quality is the most important property for material used in building, if the quality of concrete is less then the build can’t stable but if we use the best quality concrete then the building is stable.

How we know that this concrete is quality proof or not, that we generally check the strength of concrete. In easy words the Compressive Strength of Concrete determines the quality of Concrete, we check it by standard crushing test on a concrete cylinder. Concrete strength is also considered a key factor in obtaining the desired durability. For testing strength it will take 28 days this is a large time, So what we will do now? By use of Data Science, we reduce this lot’s of effort we will predict that in how much quantity we have to use which raw material for good compressive strength.

2 LITERATURE SURVEY

2.1 Existing problem

In earlier days, the concrete strength is measure through other traditional methods like using drill holes, weight spring, or using sensors. But that requires a significant destruction of test sample and thereby increasing the cost. The recommended wait time for testing the cylinder is 28 days to ensure correct results. This consumes a lot of time and requires a lot of labour to prepare different prototypes and test them. Also, this method is prone to human error and one small mistake can cause the wait time to drastically increase.

2.2 Proposed solution

The focus of this project is the application of machine learning process,Artificial nueral networks and their suitability to model concrete compressive strength compared with early models obtained from the literature and compared with some conventional approaches and also a recoomendation system is developed by applying various ML methods,Deep nueral network methods to predict the concrete strength from its components accurately and then looking for the optimal combination of components which increases the strength.

3 THEORITICAL ANALYSIS

3.1 Block diagram

1. Loading the dataset using Pandas and performed basic checks like the data type of each column and having any missing values.
2. Performed Exploratory data analysis:

- First viewed the distribution of the target feature, "Concrete compressive strength", which was in Normal distribution with a very little right skewness.
- Visualized each predictor or independent feature with the target feature and found that there's a direct proportionality between cement and the target feature while there's an inverse proportionality between water and the target feature.
- To get even more better insights, plotted both Pearson and Spearman correlations, which showed the same results as above.
- Checked for the presence of outliers in all the columns and found that the column 'age' is having more no. of outliers. Removed outliers using IQR technique, in which I considered both including and excluding the lower and upper limits into two separate dataframes and merged both into a single dataframe. This has increased the data size so that a Machine learning model can be trained efficiently.

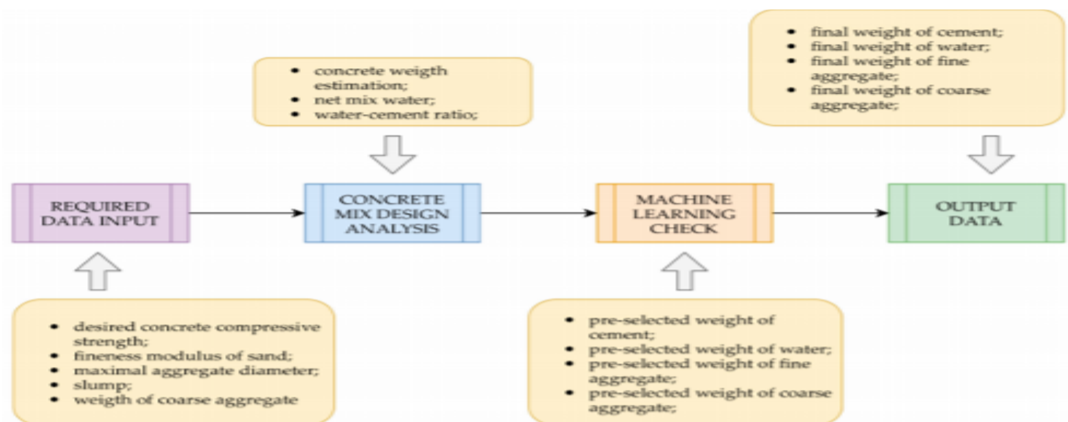
3. Experimenting with various ML algorithms:
    - First, tried with Linear regression models and feature selection using Backward elimination, RFE and the LassoCV approaches. Stored the important features found by each model into "relevant_features_by_models.csv" file into the "results" directory. Performance metrics are calculated for all the three approaches and recorded in the "Performance of algorithms.csv" file in the "results" directory. Even though all the three approaches delivered similar performance, I chose RFE approach, as the test RMSE score is little bit lesser compared to other approaches. Then, performed a residual analysis and the model satisfied all the assumptions of linear regression. But the disadvantage is, model showed slight underfitting.
    - Next, tried with various tree based models, performed hyper parameter tuning using the Randomized SearchCV and found the best hyperparameters for each model. Then, picked the top most features as per the feature importance by an each model, recorded that info into a "relevant_features_by_models.csv" file into the "results" directory. Built models, evaluated on both the training and testing data and recorded the performance metrics in the "Performance of algorithms.csv" file in the "results" directory.

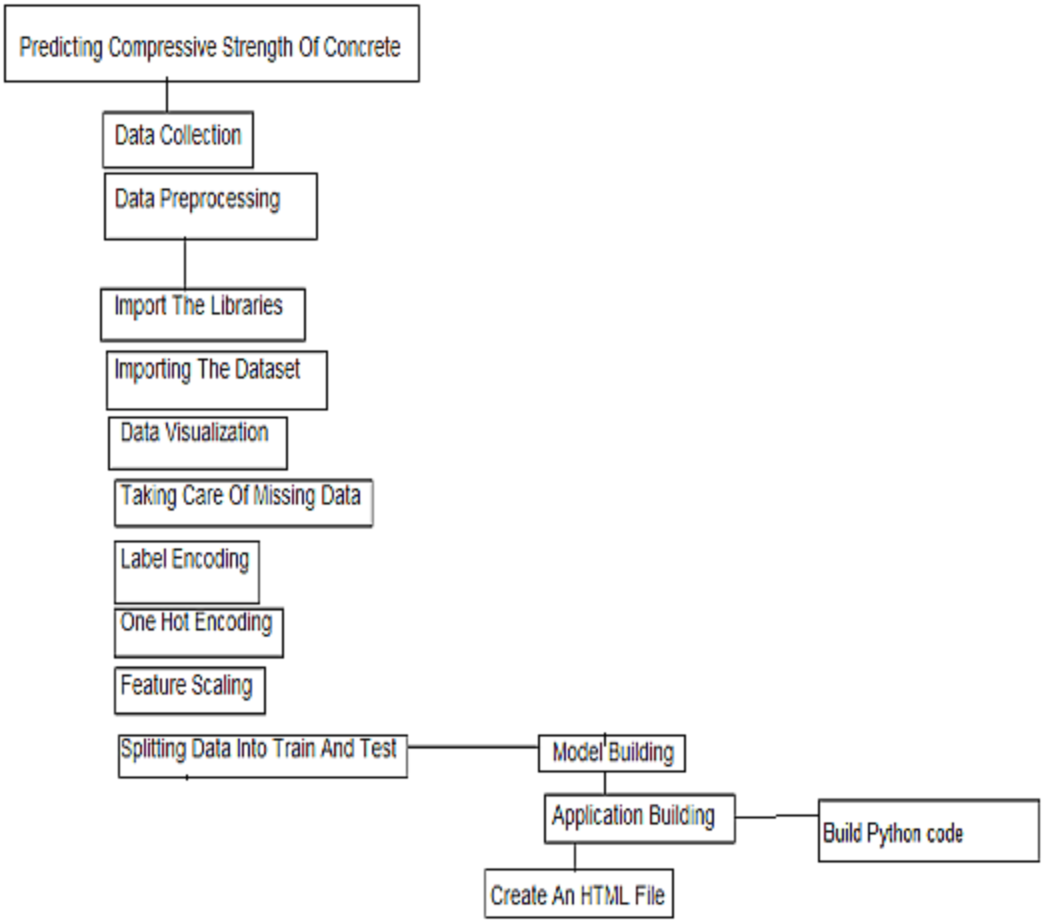
- Based on the performance metrics of both the linear and the tree based models, XGBoost regressor performed the best, followed by the random forest regressor. Saved these two models into the "models" directory.

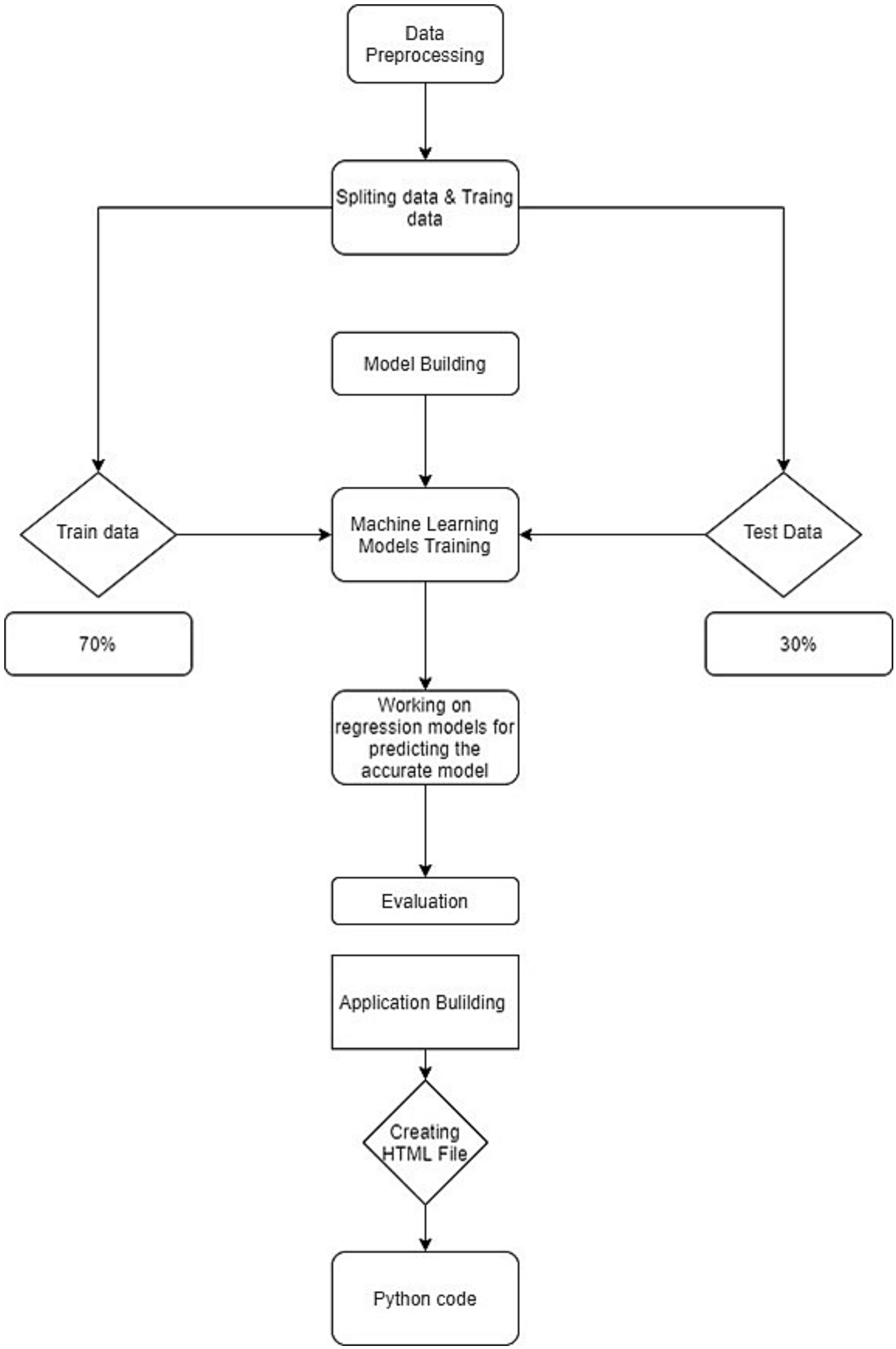
4. Deployment: Deployed the XGBoost regressor model using Flask, which works in the backend part while for the frontend UI Web page, used HTML5.

At each step in both development and deployment parts, logging operation is performed which are stored in the development_logs.log and deployment_logs.log files respectively.

So, now we can find the Concrete compressive strength quickly by just passing the quantities of the raw materials as an input to the web application .




Predicting Compressive Strength Of Concrete
Data Collection
Data Preprocessing
Import The Libraries
Importing The Dataset
Data Visualization
Taking Care Of Missing Data
Label Encoding
One Hot Encoding
Feature Scaling
Splitting Data Into Train And Test
Model Building
Application Building
Build Python code
Create An HTML File


Data Preprocessing
Spliting data & Traing data
Model Building
Train data
Machine Learning Models Training
Test Data
70%
30%
Working on regression models for predicting the accurate model
Evaluation
Application Bulilding
Creating HTML File
Python code

3.2 Hardware / Software designing

Step 1:

Anaconda Navigator :

Anaconda Navigator is a free and open-source distribution of the Python and R programming languages for data science and machine learning related applications. It can be installed on Windows, Linux, and macOS.Conda is an open-source, cross-platform, package management system. Anaconda comes with great tools like JupyterLab, Jupyter Notebook, QtConsole, Spyder, Glueviz, Orange, Rstudio, Visual Studio Code.

For this project, we will be using **Jupyter** notebook and **Spyder**

To install Anaconda navigator and to know how to use **Jupyter** Notebook & **Spyder** using Anaconda watch the video

Step 2:

To build Machine learning models you must require the following packages

Sklearn: Scikit-learn is a library in Python that provides many unsupervised and supervised learning algorithms.

NumPy: NumPy is a Python package that stands for 'Numerical Python'. It is the core library for scientific computing, which contains a powerful n-dimensional array object

Pandas: pandas is a fast, powerful, flexible, and easy to use open-source data analysis and manipulation tool, built on top of the Python programming language.

Matplotlib: It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits

Flask: Web framework used for building Web applications.

If you are using anaconda navigator, follow the below steps to download the required packages:

1. Open anaconda prompt.
2. Type “pip install numpy” and click enter.
3. Type “pip install pandas” and click enter.
4. Type “pip install matplotlib” and click enter.
5. Type “pip install scikit-learn” and click enter.
6. Type “pip install Flask” and click enter.

If you are using **Pycharm IDE**, you can install the packages through the command prompt and follow the same syntax as above.

4 EXPERIMENTAL INVESTIGATIONS

Data is obtained from UCI Machine Learning Repository and this dataset is used for all ML Algorithms. https://archive.ics.uci.edu/ml/datasets/Concrete+Compressive+Strength

Number of instances - 1030 Number of Attributes - 9 Attribute breakdown - 8 quantitative inputs, 1 quantitative output

Attribute information:

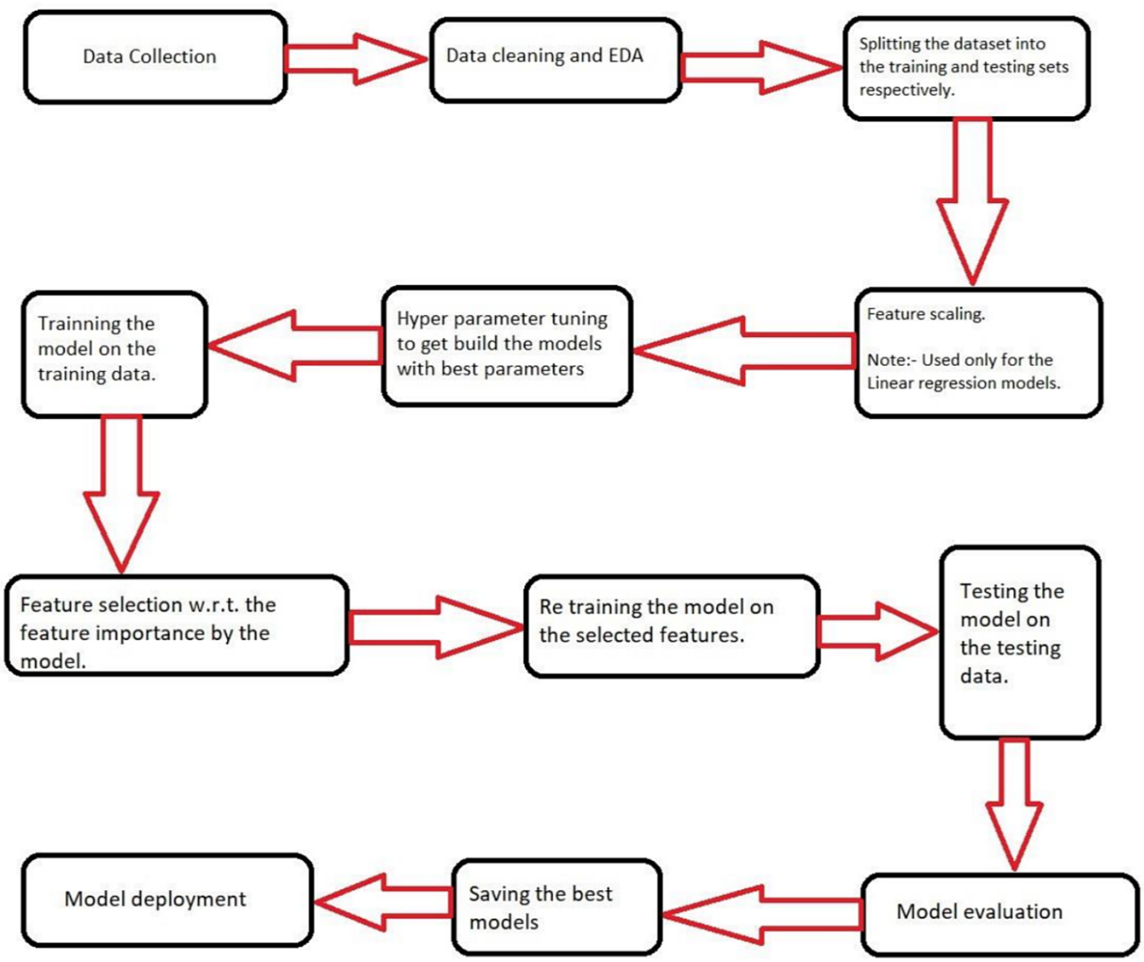
Inputs: Cement, Blast Furnace Slag, Fly Ash, Water, Superplasticizer, Coarse Aggregate, Fine Aggregate, All above features measured in kg/, Age (in days),

Output: Concrete Compressive Strength (Mpa)

ML Algorithms used: Linear regression, Lasso regression, Ridge regression, Decision Trees, Random Forests, Deep Nueral Nework

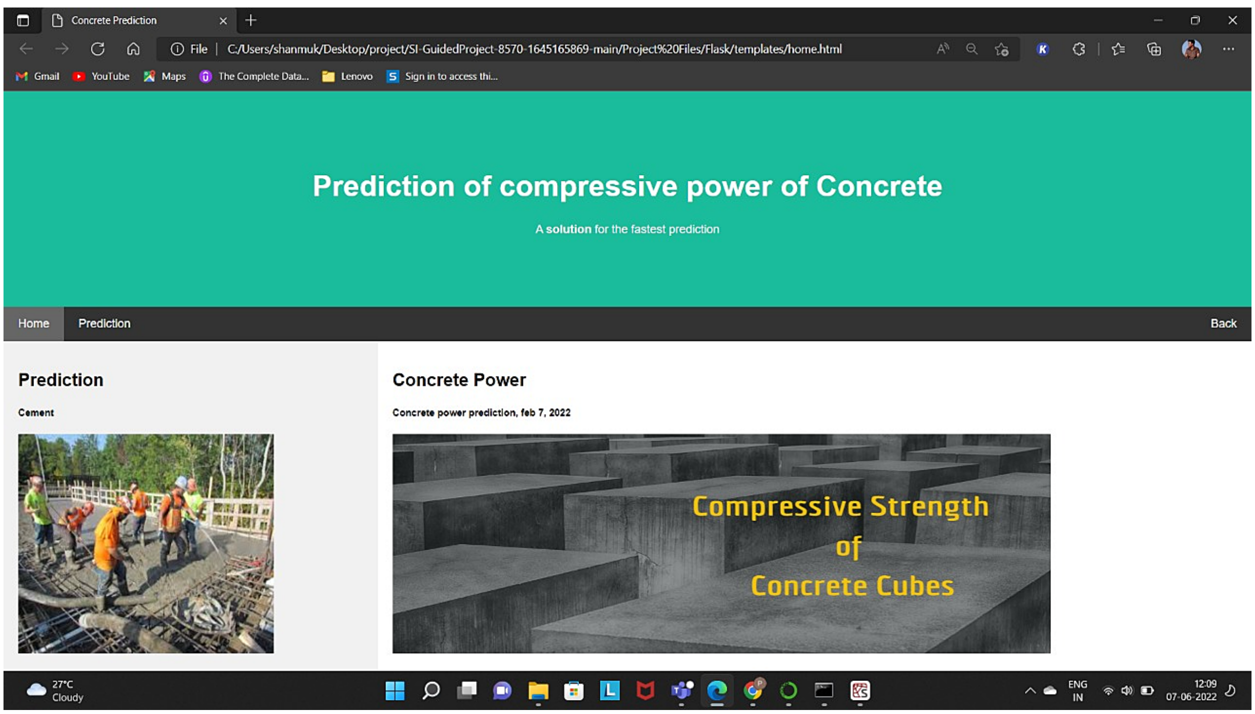
Metric - Since the target variable is a continuous variable, regression evaluation metric RMSE (Root Mean Squared Error) and R2 Score (Coefficient of Determination) have been used. And a recommendation system is developed as which algorithm is best choice for predicting accurate concrete strength.

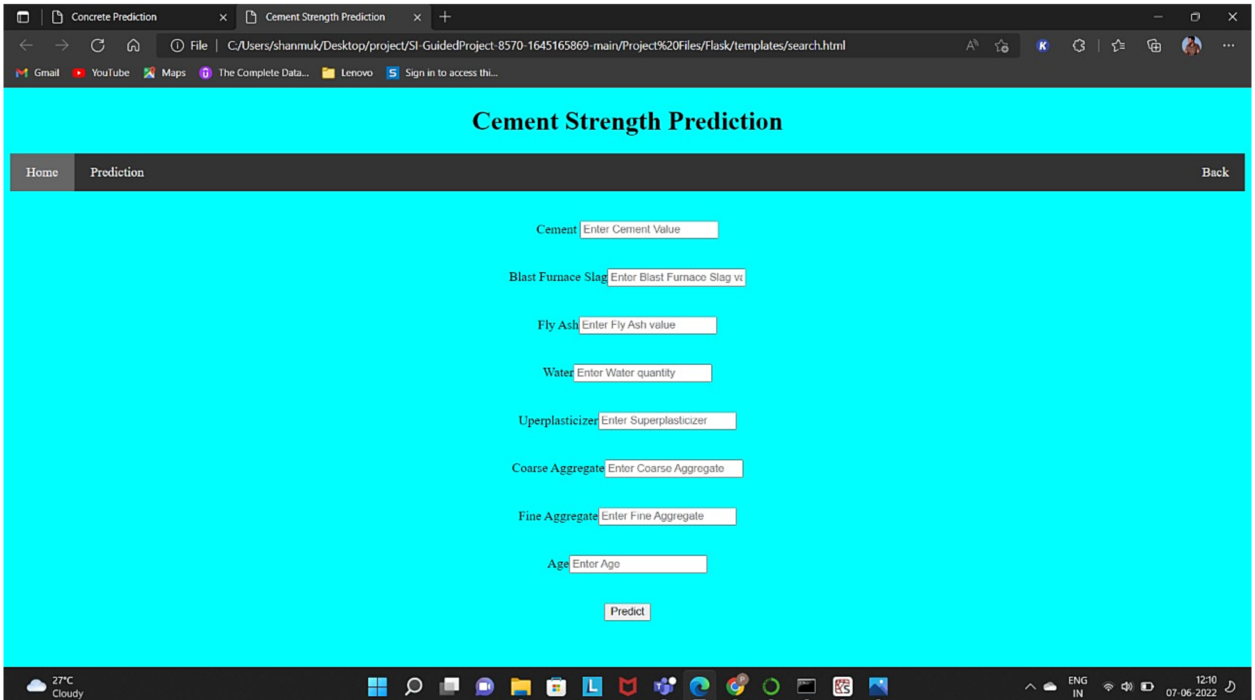
5 FLOWCHART

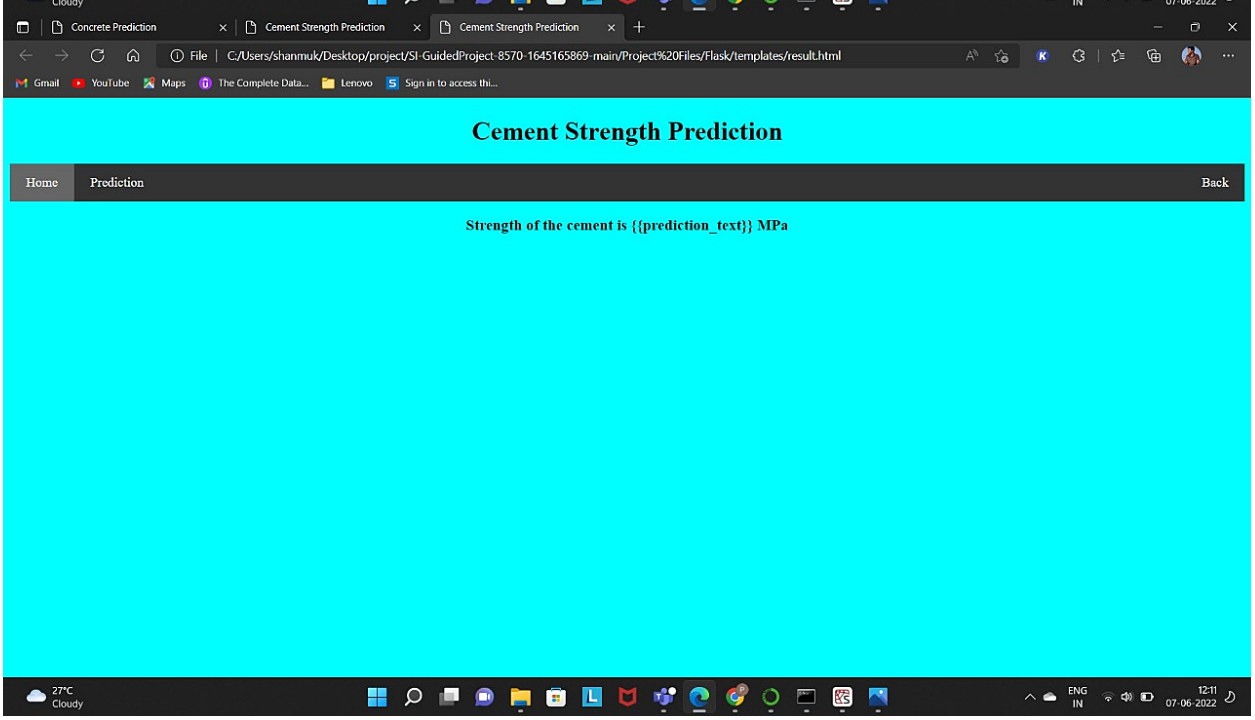


6 RESULT

The objective of the present study was to explore the applicability of the suggested models, that is, model 1 and model 2, for the prediction of concrete compressive strength. This section presents the comparative investigation of results obtained from these approaches and quantitative assessment of the models' predictive abilities. For model 1, the LM algorithm is used for training, whereas tan-sigmoid is used as an activation function for evaluating the prediction accuracy parameters. The results, as presented in Table 7, give the values of and RMSE for prediction of concrete compressive strength for both types of mixtures, namely, R1 (dataset with no substitution of cement with FA) and R2 (dataset with substitution of cement with 0.15 FA). From the results in Table 7, it can be observed that, for all the curing days, in both the cases, either R1 or R2, is above 0.90 except for R1 at a curing age of 28 days wherein it is 0.898. The low values of RMSE for all the mixes at different curing ages also indicate that the model can predict compressive strength of the mixes with high reliability. Also it can be seen that model 1, with LM as the training function, retrieves the result in just a few epochs. The maximum number of epochs taken by the model is just five, which clearly indicates that the time taken for the prediction is also very much less.







7 ADVANTAGES & DISADVANTAGES

This project will be based on a dataset obtained from the UCI Repository. The dataset consists of 1030 observations under 9 attributes. The attributes consist of 8 quantitative inputs and 1 quantitative output. The dataset does not contain any missing values. The dataset is focused on the compressive strength of a concrete. The attributes include factors that affect concrete strength such as cement, water, aggregate (coarse and fine), and fly ash etc… The objective of this project is trying to predict the concrete compressive strength based important predictors. The study will consist of evaluating the impact of different factors such as cement, water, age, fly ash, and or additives. We will evaluate the components that are highly correlated with concrete compressive strength and other components that are less influential and can be neglected through visualization or correlation matrix. In this study, we will use different machine learning techniques to predict the concrete compressive strength. Different modeling techniques will be used for the prediction. The modeling technique will include multiple linear regression, decision tree, and random forest, etc. A comparative analysis will be performed to identify the best model for our prediction in terms of accuracy. The best model will be helpful for civil engineers in choosing the appropriate concrete for bridges, houses construction.

8 APPLICATIONS

1. Loading the dataset using Pandas and performed basic checks like the data type of each column and having any missing values.
2. Performed Exploratory data analysis:
   - First viewed the distribution of the target feature, "Concrete compressive strength", which was in Normal distribution with a very little right skewness.
   - Visualized each predictor or independent feature with the target feature and found that there's a direct proportionality between cement and the target feature while there's an inverse proportionality between water and the target feature.
   - To get even more better insights, plotted both Pearson and Spearman correlations, which showed the same results as above.
   - Checked for the presence of outliers in all the columns and found that the column 'age' is having more no. of outliers. Removed outliers using IQR technique, in which I considered both including and excluding the lower and upper limits into two separate dataframes and merged both into a single dataframe. This has increased the data size so that a Machine learning model can be trained efficiently.
3. Experimenting with various ML algorithms:
   - First, tried with Linear regression models and feature selection using Backward elimination, RFE and the LassoCV approaches. Stored the important features found by each model into

"relevant_features_by_models.csv" file into the "results" directory. Performance metrics are calculated for all the three approaches and recorded in the "Performance of algorithms.csv" file in the "results" directory. Even though all the three approaches delivered similar performance, I chose RFE approach, as the test RMSE score is little bit lesser compared to other approaches. Then, performed a residual analysis and the model satisfied all the assumptions of linear regression. But the disadvantage is, model showed slight underfitting.

- Next, tried with various tree based models, performed hyper parameter tuning using the Randomized SearchCV and found the best hyperparameters for each model. Then, picked the top most features as per the feature importance by an each model, recorded that info into a "relevant_features_by_models.csv" file into the "results" directory. Built models, evaluated on both the training and testing data and recorded the performance metrics in the "Performance of algorithms.csv" file in the "results" directory.
- Based on the performance metrics of both the linear and the tree based models, XGBoost regressor performed the best, followed by the random forest regressor. Saved these two models into the "models" directory.

4. Deployment: Deployed the XGBoost regressor model using Flask, which works in the backend part while for the frontend UI Web page, used HTML5.

At each step in both development and deployment parts, logging operation is performed which are stored in the development_logs.log and deployment_logs.log files respectively.

So, now we can find the Concrete compressive strength quickly by just passing the quantities of the raw materials as an input to the web application .

9 CONCLUSION

Analysed the Compressive Strength and used Machine Learning to Predict the Compressive Strength of Concrete. We have used Linear Regression and its variations, Decision Trees and Random Forests to make predictions and compared their performance. Random Forest Regressor has the lowest RMSE and is a good choice for this problem. Also, we can further improve the performance of the algorithm by tuning the hyperparameters by performing a grid search or random search.
NN approaches combine the complexity of many statistical techniques with machine learning techniques and attributed as a black-box which allows NN to be applied in all

engineering disciplines. It comes out as the best possible model for the prediction of compressive strength of concrete. It has predicted with high accuracy for all the curing ages, that is, 28, 56, and 91 days.

10 FUTURE SCOPE

So prediction of compressive strength of concrete has been an active area of research. The aim of the present study is to compare two emerging soft computing techniques, that is, Artificial Neural Network and Genetic Programming (GP), used for concrete compressive strength prediction, by using the experimental data.

11 BIBILOGRAPHY

References

1. H.-G. Ni and J.-Z. Wang, "Prediction of compressive strength of concrete by neural networks," *Cement and Concrete Research*, vol. 30, no. 8, pp. 1245–1250, 2000.View at: Publisher Site | Google Scholar

2. A. Baykasoglu, T. Dereli, and S. Taniş, "Prediction of cement strength using soft computing techniques," *Cement and Concrete Research*, vol. 34, no. 11, pp. 2083–2090, 2004.View at: Publisher Site | Google Scholar

3. S. Akkurt, G. Tayfur, and S. Can, "Fuzzy logic model for the prediction of cement compressive strength," *Cement and Concrete Research*, vol. 34, no. 8, pp. 1429–1433, 2004.View at: Publisher Site | Google Scholar

4. A. Öztaş, M. Pala, E. Özbay, E. Kanca, N. Çağlar, and M. A. Bhatti, "Predicting the compressive strength and slump of high strength concrete using neural network," *Construction and Building Materials*, vol. 20, no. 9, pp. 769–775, 2006.View at: Publisher Site | Google Scholar

5. M. Pala, E. Özbay, A. Öztaş, and M. I. Yuce, “Appraisal of long-term effects of fly ash and silica fume on compressive strength of concrete by neural networks,” *Construction and Building Materials*, vol. 21, no. 2, pp. 384–394, 2007.View at: Publisher Site | Google Scholar

6. N. P. Rajamane, J. A. Peter, and P. S. Ambily, “Prediction of compressive strength of concrete with fly ash as sand replacement material,” *Cement and Concrete Composites*, vol. 29, no. 3, pp. 218–223, 2007.View at: Publisher Site | Google Scholar

7. M. Ozturan, B. Kutlu, and T. Ozturan, “Comparison of concrete strength prediction techniques with artificial neural network approach,” *Building Research Journal*, vol. 56, pp. 23–36, 2008.View at:

8. M. M. Alshihri, A. M. Azmy, and M. S. El-Bisy, “Neural networks for predicting compressive strength of structural light weight concrete,” *Construction and Building Materials*, vol. 23, no. 6, pp. 2214–2219, 2009.View at: Publisher Site | Google Scholar

9. M. Saridemir, “Prediction of compressive strength of concretes containing metakaolin and silica fume by artificial neural networks,” *Advances in Engineering Software*, vol. 40, no. 5, pp. 350–355, 2009.View at: Publisher Site | Google Scholar | Zentralblatt MATH

10. M. Saridemir, “Genetic programming approach for prediction of compressive strength of concretes containing rice husk ash,” *Construction and Building Materials*, vol. 24, no. 10, pp. 1911–1919, 2010.View at: Publisher Site | Google Scholar

11. A. M. Diab, H. E. Elyamany, A. E. M. Abd Elmoaty, and A. H. Shalan, “Prediction of concrete compressive strength due to long term sulfate attack using neural network,”

Alexandria Engineering Journal, vol. 53, no. 3, pp. 627–642, 2014.View at: Publisher Site | Google Scholar

12. M. Kumar, *Reliability based design of structural elements [Ph.D. thesis]*, Thapar Institute of Engineering and Technology Patiala, Patiala, India, 2002.
13. P. K. Simpson, Ed., *Neural Networks, Theory, Technology, and Applications*, IEEE, New York, NY, USA, 1996.
14. K. Knight, "Connectionist ideas and algorithms," *Communications of the ACM*, vol. 33, no. 11, pp. 58–74, 1990.View at: Publisher Site | Google Scholar | MathSciNet
15. A. K. Sharma, R. K. Sharma, and H. S. Kasana, "Prediction of first lactation 305-day milk yield in Karan Fries dairy cattle using ANN modeling," *Applied Soft Computing Journal*, vol. 7, no. 3, pp. 1112–1120, 2007.View at: Publisher Site | Google Scholar
16. A. Kamalloo, Y. Ganjkhanlou, S. H. Aboutalebi, and H. Nouranian, "Modeling of compressive strength of Metakaolin based geopolymers by the use of artificial neural network," *International Journal of Engineering*, vol. 23, no. 2, pp. 145–152, 2010.View at: Google Scholar
17. J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, The MIT Press, Cambridge, Mass, USA, 1992.
18. A. Ghodratnamaa, R. Tavakkoli-Moghaddam, and A. Babolic, "Comparing three proposed meta-heuristics to solve a new P-HUB location-allocation problem," *International Journal of Engineering*, vol. 26, no. 9, pp. 1043–1058, 2013.View at: Google Scholar

19. J. Namyong, Y. Sangchun, and C. Hongbum, “Prediction of compressive strength of In-situ concrete based on mixture proportions,” *Journal of Asian Architecture and Building Engineering*, vol. 3, no. 1, pp. 9–16, 2004.View at: Publisher Site | Google Scholar

APPENDIX : https://github.com/Shanmugapolisetty/project.git