PREDICTING COMPREHENSIVE STRENGTH OF CONCRETE

NAME: PRIYADARSHINI S **REGNO: 39110804**

BRANCH: BE.CSE **BATCH: 2019-23**

1.Introduction

1.1 Overview

- Concrete is a material used in construction that has great versatility and which is used across the globe. Concrete has several advantages, including good compressive strength, durability, workability, construction availability, and low cost. Determining accurate concrete strength is a major civil engineering problem. The traditional way to know the strength of concrete will take about 28 days which is very time-consuming. It is important to wait 28 days to ensure the quality control of the process, although it is very time-consuming.

- This project aims at building a Machine Learning model to predict the compressive strength of concrete to greatly speed up prediction with higher accuracy. A web application is built where the user can enter the required parameters and see the predicted results on Web Application.

1.2 Purpose

- The prediction of compressive strength of concrete has great connotation, if it is brisk and consistent because it offers an option to do the essential modification on the mix proportion used to avoid circumstances where concrete does not attain the mandatory design strength or by avoiding concrete that is gratuitously sturdy and also for more economic use of raw material and fewer construction failures, hence reducing construction cost. 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

2. Literature Survey

2.1 Existing Problem

- Datasets were randomly split into 80% for the training set and 20% for the independent test set. The training data were used to train the ML model. The independent test data were applied for the evaluation of the model's performance. The 10-fold cross-validation procedure helped in the estimation of the ML model skills.
- The ML pre-processing steps were applied to the raw datasets before they could be utilized for the regression method training.
- The ML regression models were trained to predict the f_c and S values. The correlation coefficient (R), root mean squared error (RMSE), and mean absolute error (MAE) metrics were employed to compare the models' prediction performance. According to these statistical results, the most successful regression method was determined to predict the f_c and S values. Afterward, the feature selection method was used to obtain the subset with fewer features, and the prediction accuracy was examined.

2.2 Proposed Solution

- In Linear regression model we have used gradient boosting regression is used as a regression model.
- Gradient boosting regression calculates the difference between the current predict the known correct target value. This difference is called residual. After that Gradient boosting Regression trains a weak model that maps features to that residual. This residual predicted by a weak model is added to the existing model input and thus this process nudges the model towards the correct target.

Repeating this step again and again improves the overall model prediction.

3. Theoretical Analysis

3.1 Block diagram

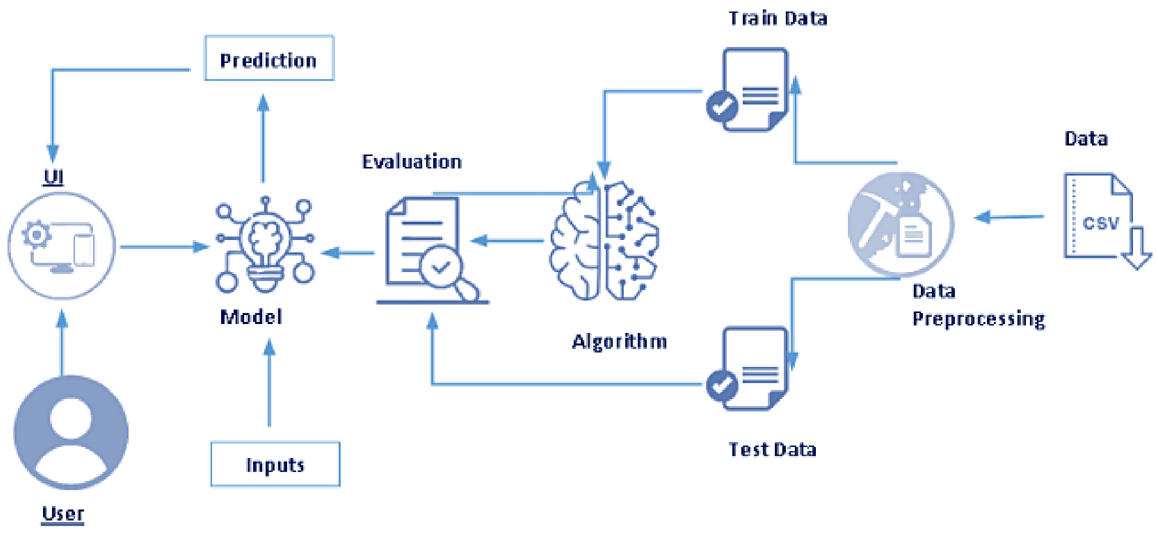


Fig 3.1.1 Block diagram

3.2 Hardware / Software designing

Hardware

- Processor
  - 64-bit, four-core
  - 2.5 GHz minimum per core
- RAM- 8 GB for developer and evaluation use
- Hard disk -4GB

Software

- Operating System – Windows, Linux
- Front end
  - Design Language - HTML, CSS
- Anaconda Navigator
  - It consists of Spyder-used for flask app development, Jupyter tool-used for python programming
- Programming Language – Python

4.Experimental Investigations

- Gradient Boosting trains many models in a gradual, additive and sequential manner. It can be used for both classification and regression problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees.
- **Quality of Raw Materials**
  - Cement**:** Provided the cement conforms with the appropriate standard and it has been stored correctly (i.e., in dry conditions), it should be suitable for use in concrete.
  - Aggregates**:** Quality of aggregates, its size, shape, texture, strength etc determines the strength of concrete. The presence of salts (chlorides and sulphates), silt and clay also reduce the strength of concrete.
  - Water**:** frequently the quality of the water is covered by a clause stating "...the water should be fit for drinking...". This criterion though is not absolute and reference should be made to respective codes for testing of water construction purpose.
- **Age of concrete**

- The degree of hydration is synonymous with the age of concrete provided the concrete has not been allowed to dry out or the temperature is too low. In theory, provided the concrete is not allowed to dry out, then it will always be increasing albeit at an ever-reducing rate. For convenience and for most practical applications, it is generally accepted that the majority of the strength has been achieved by 28 days.

- **Water / Cement Ratio**

- The higher the water/cement ratio, the greater the initial spacing between the cement grains and the greater the volume of residual voids not filled by hydration products.

- A lower water cement ratio means less water, or more cement and lower workability. However if the workability becomes too low the concrete becomes difficult to compact and the strength reduces. For a given set of materials and environment conditions, the strength at any age depends only on the water-cement ratio, providing full compaction can be achieved.

- **Coarse / fine aggregate ratio**

- If the proportion of fines is increased in relation to the coarse aggregate, the overall aggregate surface area will increase.

- If the surface area of the aggregate has increased, the water demand will also increase.

- Assuming the water demand has increased, the water cement ratio will increase. Since the water cement ratio has increased, the compressive strength will decrease.

- **Temperature**

- The rate of hydration reaction is temperature dependent. If the temperature increases the reaction also increases. This means that the concrete kept at higher temperature will gain strength more quickly than a similar concrete kept at a lower temperature. However, the final strength of the concrete kept at the higher temperature will be lower. This is because the physical form of the hardened cement paste is less well structured and more porous when hydration proceeds at faster rate. This is an important point to remember because temperature has a similar but more pronounced detrimental effect on permeability of the concrete

- **Relative humidity**
- If the concrete is allowed to dry out, the hydration reaction will stop. The hydration reaction cannot proceed without moisture.
- **Curing**
- It should be clear from what has been said above that the detrimental effects of storage of concrete in a dry environment can be reduced if the concrete is adequately cured to prevent excessive moisture loss

5.Flowchart

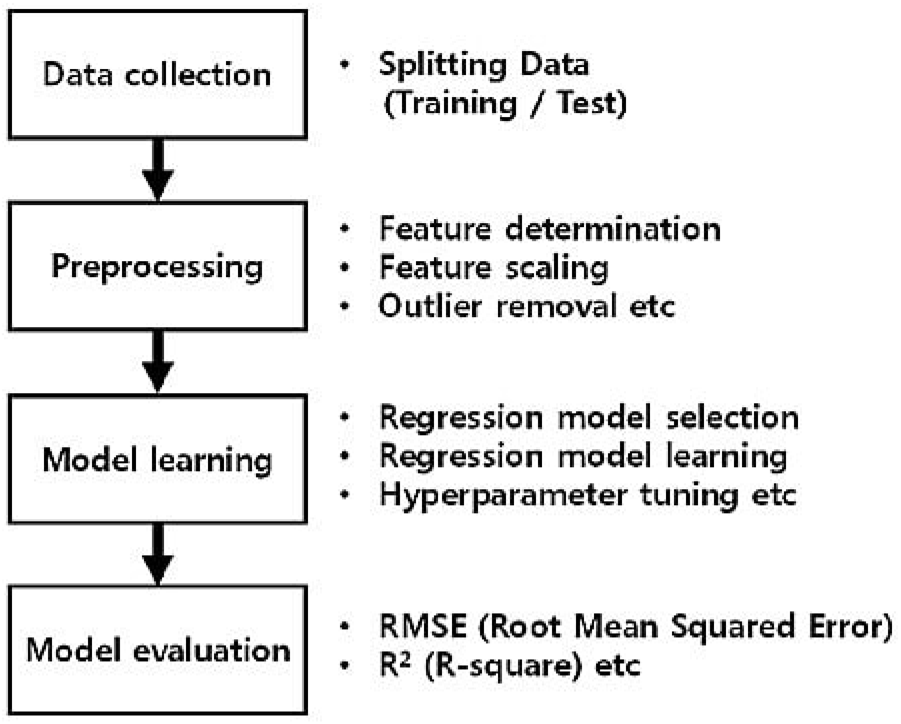

Data collection
Splitting Data (Training / Test)
Preprocessing
Feature determination
Feature scaling
Outlier removal etc
Model learning
Regression model selection
Regression model learning
Hyperparameter tuning etc
Model evaluation
RMSE (Root Mean Squared Error)
R² (R-square) etc

Fig 5.1 Flowchart

6.Result

```
data.boxplot(figsize=(18,7))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x2d7766af2b0>
```

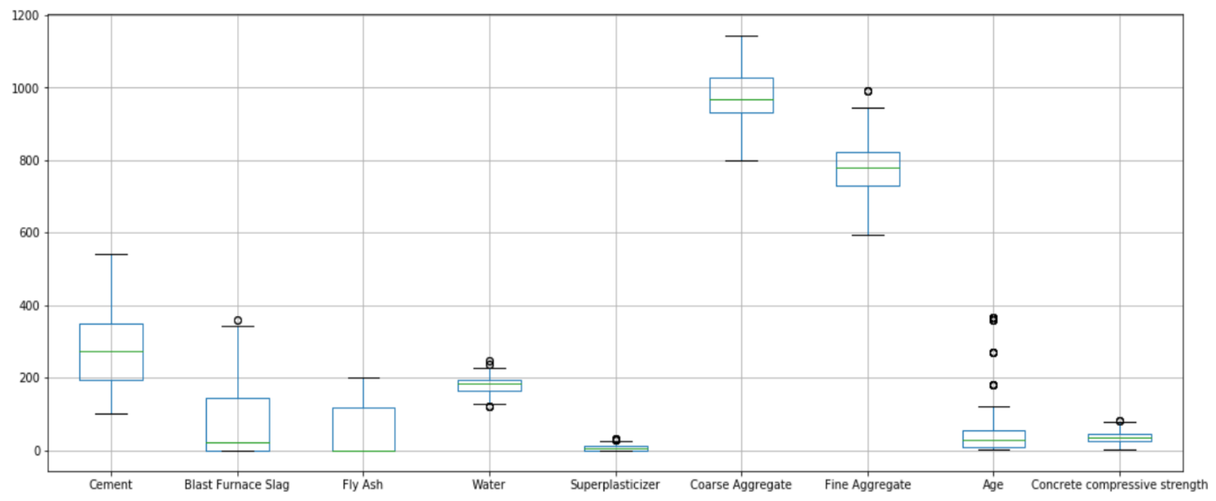
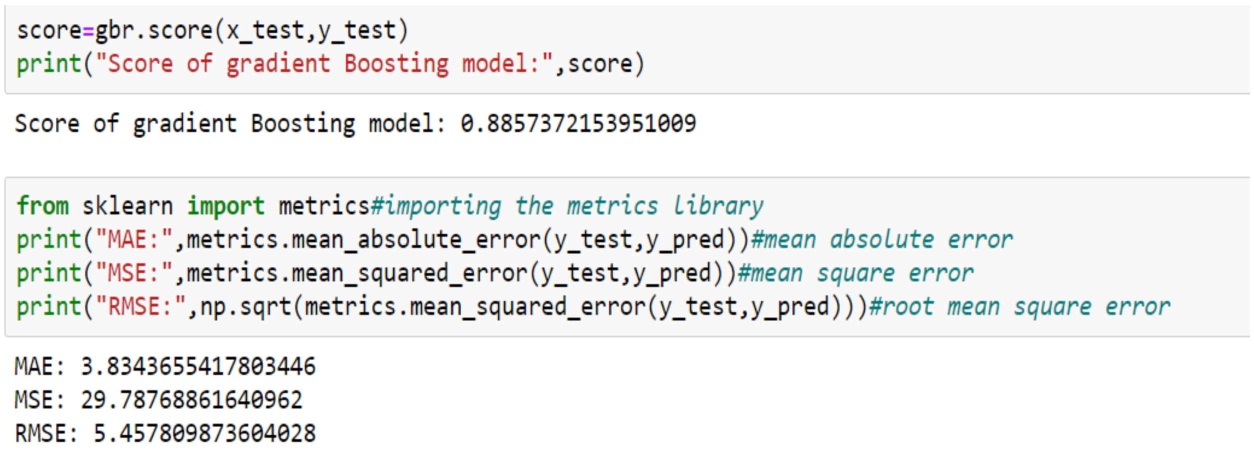


Fig 6.1 Boxplot



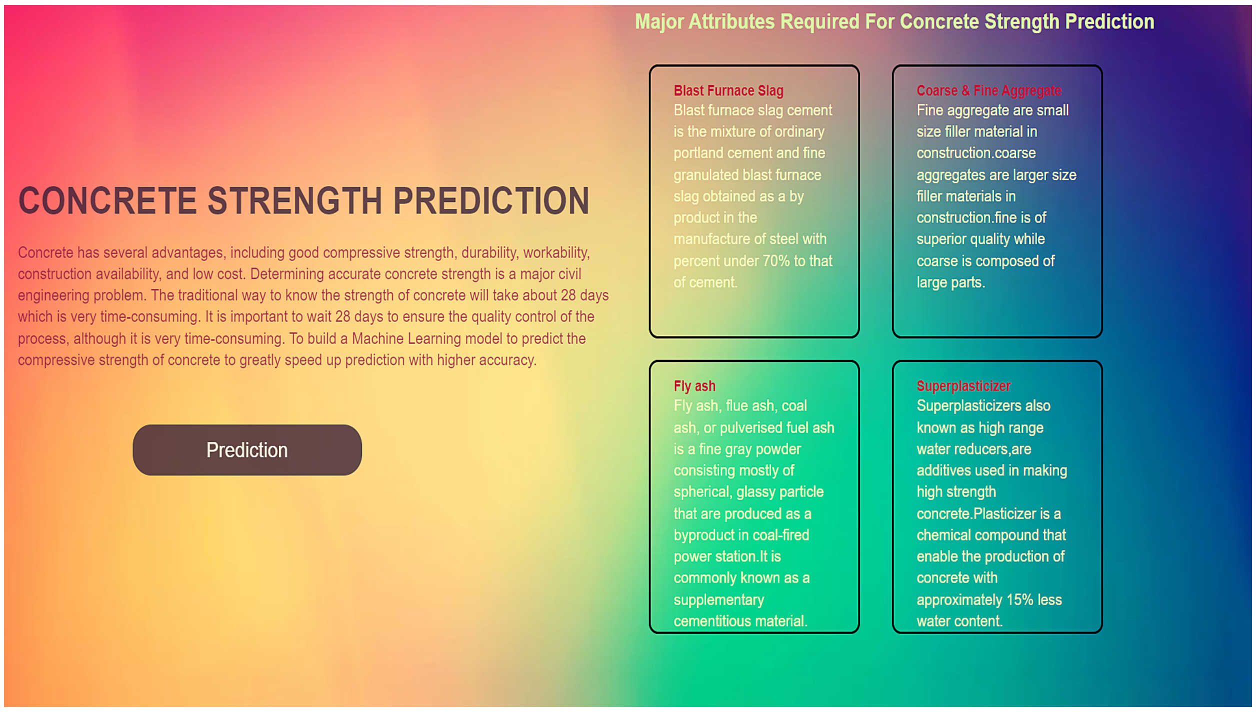
```
score=gbr.score(x_test,y_test)
print("Score of gradient Boosting model:",score)
```

```
Score of gradient Boosting model: 0.8857372153951009
```

```
from sklearn import metrics#importing the metrics library
print("MAE:",metrics.mean_absolute_error(y_test,y_pred))#mean absolute error
print("MSE:",metrics.mean_squared_error(y_test,y_pred))#mean square error
print("RMSE:",np.sqrt(metrics.mean_squared_error(y_test,y_pred)))#root mean square error
```

```
MAE: 3.8343655417803446
MSE: 29.78768861640962
RMSE: 5.457809873604028
```

Fig 6.2 Error calculations



Major Attributes Required For Concrete Strength Prediction

CONCRETE STRENGTH PREDICTION

Concrete has several advantages, including good compressive strength, durability, workability, construction availability, and low cost. Determining accurate concrete strength is a major civil engineering problem. The traditional way to know the strength of concrete will take about 28 days which is very time-consuming. It is important to wait 28 days to ensure the quality control of the process, although it is very time-consuming. To build a Machine Learning model to predict the compressive strength of concrete to greatly speed up prediction with higher accuracy.

Prediction

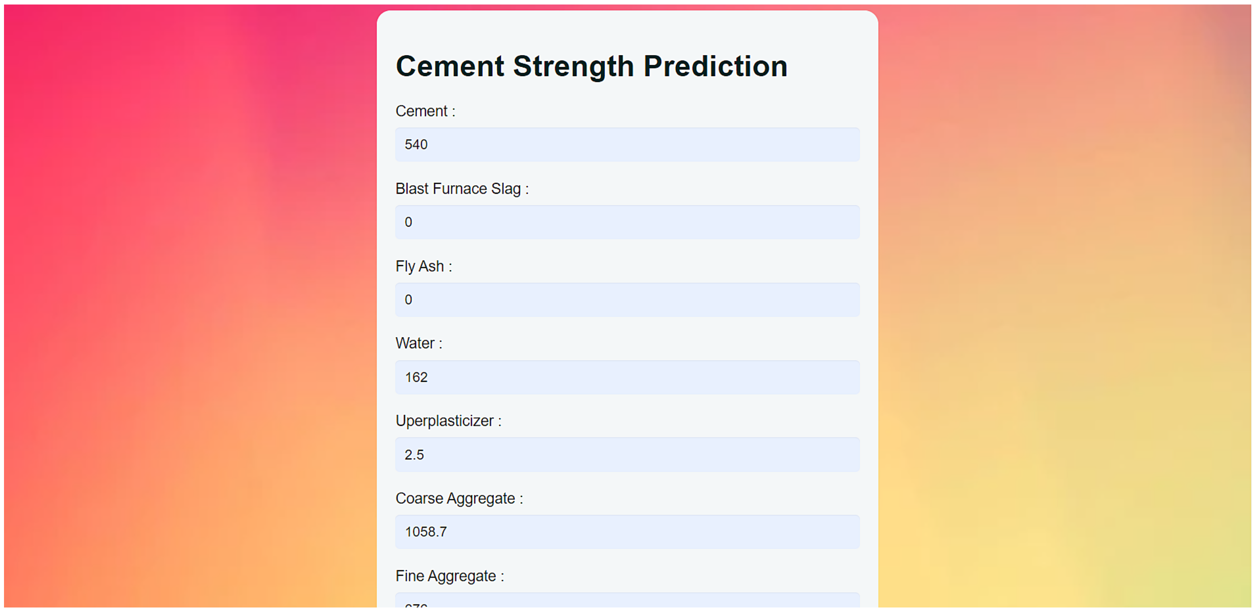
Blast Furnace Slag
Blast furnace slag cement is the mixture of ordinary portland cement and fine granulated blast furnace slag obtained as a by product in the manufacture of steel with percent under 70% to that of cement.

Coarse & Fine Aggregate
Fine aggregate are small size filler material in construction.coarse aggregates are larger size filler materials in construction.fine is of superior quality while coarse is composed of large parts.

Fly ash
Fly ash, flue ash, coal ash, or pulverised fuel ash is a fine gray powder consisting mostly of spherical, glassy particle that are produced as a byproduct in coal-fired power station.It is commonly known as a supplementary cementitious material.

Superplasticizer
Superplasticizers also known as high range water reducers,are additives used in making high strength concrete.Plasticizer is a chemical compound that enable the production of concrete with approximately 15% less water content.

Fig 6.3 Home page



Cement Strength Prediction

Cement :

540

Blast Furnace Slag :

0

Fly Ash :

0

Water :

162

Uperplasticizer :

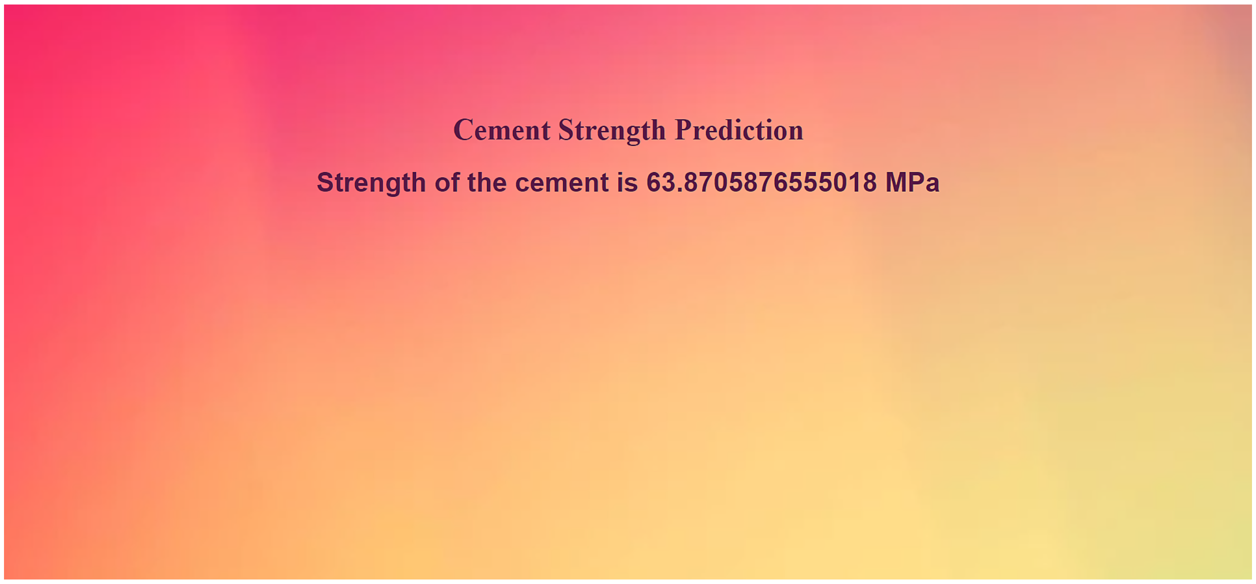
2.5

Coarse Aggregate :

1058.7

Fine Aggregate :

Fig 6.4 Prediction page



Cement Strength Prediction

Strength of the cement is 63.8705876555018 MPa

Fig 6.5 Result page

7. Advantages and Disadvantages

7.1 Advantages

- The traditional way to know the strength of concrete will take about 28 days which is very time-consuming. It is important to wait 28 days to ensure the quality control of the process, although it is very time-consuming. This project

Helps to predict the compressive strength of concrete to greatly speed up prediction with higher accuracy.

7.2 Disadvantages

- Although machine learning has been transformative in some fields, machine-learning programs often fail to deliver expected results. Reasons for this are numerous: lack of (suitable) data, lack of access to the data, data bias, privacy problems, badly chosen tasks and algorithms, wrong tools and people, lack of resources, and evaluation problems

8. Applications

- The use of machine learning (ML) techniques to predict the mechanical properties of concrete will be beneficial to the field of civil engineering because it will save time, effort, and resources. The proposed techniques are efficient to forecast the strength properties of concrete containing supplementary cementitious materials (SCM) and pave the way towards the intelligent design of concrete elements and structures.

9.Conclusion

- This study presents a comparative study between two construction models, namely the single model and ensemble models, in predicting the concrete compressive strength based on concrete samples. These samples were used to form a database that was used to create a prediction model as well as to test the accuracy of the prediction model formed. Four accuracy indicators are used in evaluating the performance of each method, including RMSE, MSE, and MAE.

10. Future Scopes

- Overall, these results establish machine learning as a promising approach to predict the strength of commercial concrete based on the sole knowledge of their mixture proportions.
- The accuracy of the project is 88% .To enhance the further accuracy of the project we are in a plan to implement this concept with any other machine learning concept for much more precise output.

11.Bibilography

- [3] Chandra, B. Paul V, P, “A Robust Algorithm for Classification Using Decision Trees”, Dept. of Math, IIT, New Delhi, December(2006).
- https://www.cin.ufpe.br/~cavmj/Machine%20-%20Learning%20-%20Tom%20Mitchell.pdf

Appendix

A. **Source Code**

Cement.iynb

```
import numpy as np
import pandas as pd
from collections import Counter as c
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score,mean_squared_error,mean_absolute_error
import pickle from sklearn.linear_model import LinearRegression
data=pd.read_csv('concrete.csv')
```

```
type(data)
data.head()
data.tail()
data.describe()
data.isnull().any()
data.isnull().sum()
data.columns=[col[:col.find("(")].strip() for col in data.columns]
data.head(1)
data.boxplot(figsize=(18,7))
x=pd.DataFrame(data,columns=data.columns[:8])
y=pd.DataFrame(data,columns=data.columns[8:])
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=1)
from sklearn.ensemble import GradientBoostingRegressor
gbr=GradientBoostingRegressor()#object of GradientBoostRegressor
gbr.fit(x_train,y_train)
y_pred=gbr.predict(x_test)#predicting using predict method
print("Prediction made by Gradient Boosting model:",y_pred)
score=gbr.score(x_test,y_test)
print("Score of gradient Boosting model:",score)
from sklearn import metrics#importing the metrics library
print("MAE:",metrics.mean_absolute_error(y_test,y_pred))#mean absolute error
print("MSE:",metrics.mean_squared_error(y_test,y_pred))#mean square error
print("RMSE:",np.sqrt(metrics.mean_squared_error(y_test,y_pred)))
import pickle
pickle.dump(gbr,open('cement.pkl','wb'))
```

app.py

```
import numpy as np
```

```
from flask import Flask, request, render_template
import pickle
import pandas as pd
app = Flask(__name__) # initializing a flask app
model =pickle.load(open('cement.pkl', 'rb'))
@app.route('/')# route to display the home page
def home():
   return render_template('home.html')
@app.route('/Prediction',methods=['POST','GET'])
def prediction():
   return render_template('index1.html')
@app.route('/Home',methods=['POST','GET'])
def my_home():
   return render_template('home.html')
@app.route('/predict',methods=['POST'])
def index():
         #  reading the inputs given by the user
   input_features = [float(x) for x in request.form.values()]
   features_value = [np.array(input_features)]
   features_name = ['Cement', 'Blast Furnace Slag', 'Fly Ash', 'Water',
                'Superplasticizer','Coarse Aggregate', 'Fine Aggregate','Age']
   x = pd.DataFrame(features_value, columns=features_name)
   #x_log=np.log(x)#performing log transformation
   # predictions using the loaded model file
   prediction=model.predict(x)
   print('prediction is', prediction)
   # showing the prediction results in a UI
   return render_template('result2.html',prediction_text=prediction[0])
if __name__ == "__main__":
   app.run(host='127.0.0.1', port=8001, debug=True)
```