**Project**

**On**

Predicting Compressive Strength of Concrete

Using

Machine learning Algorithms

By Upender Singh

**Table of Contents**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1** | |  | **INTRODUCTION** | |
|  | |  | 1.1 Overview | |
|  | |  | 1.2 Purpose | |
| **2** | |  | **LITERATURE SURVEY** | |
|  | |  | 2.1 Existing problem | | |
|  | |  | 2.2 Proposed solution | | |
| **3** | |  | **THEORITICAL ANALYSIS** | |
|  | |  | 3.1 Block diagram | |
|  | |  | 3.2 Hardware / Software designing | | |
| **4** | |  | **EXPERIMENTAL INVESTIGATIONS** | | |
| **5** | |  | **FLOWCHART** | |
| **6** | |  | **RESULT** |  |
| **7** | |  | **ADVANTAGES & DISADVANTAGES** | | |
| **8** | |  | **APPLICATIONS** | |
| **9** | |  | **CONCLUSION** | |
| **10** | |  | **FUTURE SCOPE** | |
| **11** | |  | **BIBILOGRAPHY** | |
|  | |  | **APPENDIX** | |
|  |  | | A. Source code | |

1. **Introduction** 
   1. **Overview:**

Concrete is a complex composite material. The predictability of concrete properties is extremely low. Therefore, it is challenging to model the concrete properties according to the effect variables. The biggest challenge of experimental designs is a high number of effect variables affecting the response variables. Multiple effect variables increase the number of trials. The higher amount of uncontrollable variables makes it difficult to obtain the real response function.

Generally, the one-factor-at-a-time method is used in experimental designs to determine the concrete properties. The major disadvantage of this approach is that it does not consider the interaction between the factors (interaction terms). The higher the number of the controlled and uncontrolled effect variables that influence the concrete properties, the lesser the predicted accuracy. Despite this, a few experimental designs have been suggested by considering the controllable effect variables and interaction terms between them [1].

**1.2 Purpose:**

In recent years, the ML methods have become popular as they allow researchers to improve the prediction accuracy of concrete properties and are used for various engineering applications. The ML methods have been used to increase the prediction accuracy of concrete properties, and the data derived from the literature sources were used

Regression models tend to be used for the prediction of the compressive strength of high-strength concrete. These models also demonstrate how the concrete compressive strength depends on the mixing ratios.

Previous studies evaluated the amount of the concrete component materials and compared their results to the published data. In this study, the ML regression methods were compared to predict the compressive strength and slump values of the cube samples. The samples were prepared by accounting for seven simultaneously controllable effect variables in the laboratory. The study aimed to determine the most successful regression method by comparing the decision tree (DT), random forest (RF), support vector machine (SVM)

1. **Literature Survey**

**2.1 Existing Problem**

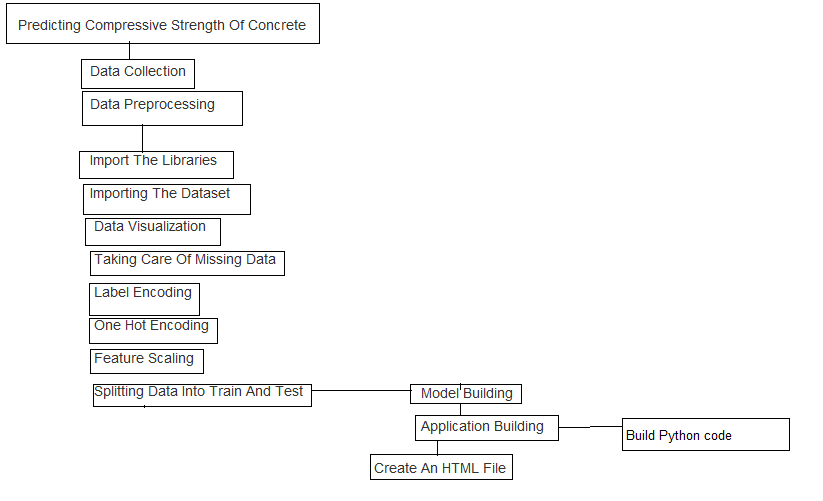
This is generally determined by a standard crushing test on a concrete cylinder. This requires engineers to build small concrete cylinders with different combinations of raw materials and test these cylinders for strength variations with a change in each raw material. The recommended wait time for testing 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**

This is generally determined by a standard crushing test on a concrete cylinder. This requires engineers to build small concrete cylinders with different combinations of raw materials and test these cylinders for strength variations with a change in each raw material. 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.

1. **Theoretical Analysis**

**3.1 Block Diagram**

****

**3.2 Hardware / Software designing**

Python, Python Web Frame Works, Python for Data Analysis, Python For Data Visualization, Data Pre-processing Techniques, Machine Learning, Regression Algorithms

1. **Experimental Investigation**

The compressive strength data for the present work was obtained from the experiments. For generating a reliable data bank on concrete compressive strength, he had considered five parameters, namely, water-cementitious material ratio, cementitious content, water content, workability, and curing ages in the experimental program.

The casting and testing of specimens for generating the data bank were performed in controlled laboratory conditions.

**Range of various parameters**

Cement (component 1)(kg in a m^3 mixture) = 102 – 540

Blast Furnace Slag (component 2)(kg in a m^3 mixture) = 0 – 359.4

Fly Ash (component 3)(kg in a m^3 mixture) = 0 – 200.1

Water (component 4)(kg in a m^3 mixture) = 121.75 - 247

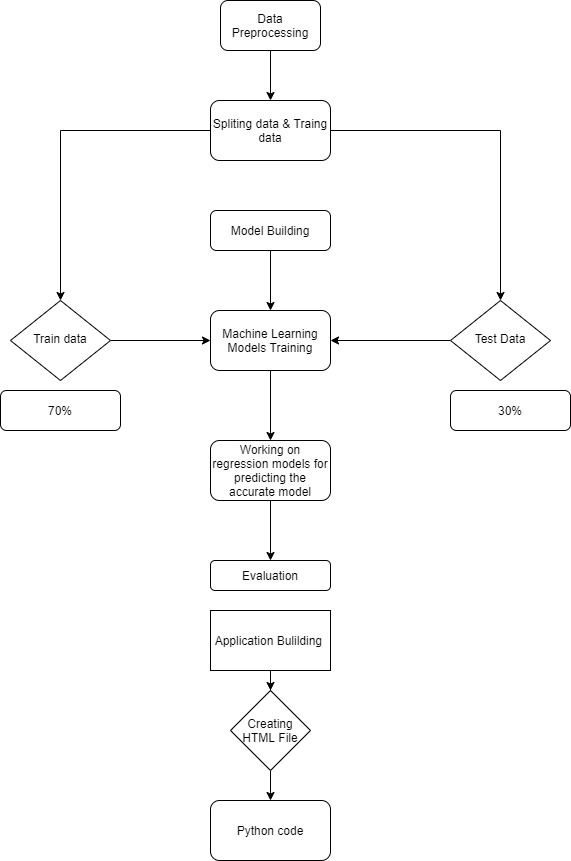
Superplasticizer (component 5)(kg in a m^3 mixture) = 0-32.2

Coarse Aggregate (component 6)(kg in a m^3 mixture) = 801- 1145

Fine Aggregate (component 7)(kg in a m^3 mixture) = 594 – 992.6

Age (day) = 1 - 365

1. **Flowchart**



1. **Result**

We have analysed the Compressive Strength Data and used Machine Learning to Predict the Compressive Strength of Concrete. We have used Linear Regression and its variations, Lasso, Ridge and Random Forests to make predictions and compared their performance. Random Forest Regressor has highest accuracy and is a good choice for this problem. Random Forest Regressor trains randomly initialized trees with random subsets of data sampled from the training data, this will make our model more robust

1. **Advantages and Disadvantages**

**Advantages:**

Using Machine learning to predict the strength of the concrete will be time and more accuracy in predicting the approximately close value can be done easily. Its more trust worthy and cost effective .It also reduces the man power for doing the experiments to find the strength of the concrete in different unknown situations.

**Disadvantages :**

There is a 3 % chances that the outcome will not predict the approximate value in that situation it can be troublesome.

1. **Applications:**

* Can predict the strength of the concrete using the inputs provided.
* Implementable on the website

1. **Conclusion**

In this study, a prediction model of compressive strength was established by Random Forest Regressor. A total of 1030 sample data collected from the experimental test were used to develop the Random Forest Regression model for predicting compressive strength. The Random Forest model was first calibrated and then verified using the experimental data from concrete samples. Conclusions can be drawn as follows:

* Compare to all other Machine Learning Models Random Forest was best suitable for this data.
* Random Forest Regressor gave the maximum accuracy when tested using r2\_score confusion matrix.
* Maximum accuracy received is 97.25 %.

1. **Future Scope**

This model can predict the outcome with many different inputs within seconds. The model will save a lot of time of the construction companies and the civil engineers. Experiment cost is also reduced with creates a bigger opportunity for construction companies in cost effectiveness work.

1. **Bibliography**

**Books**

Hastie, Friedman, and Tibshirani, The Elements of Statistical Learning, 2001

Bishop, Pattern Recognition and Machine Learning, 2006

Ripley, Pattern Recognition and Neural Networks, 1996

Duda, Hart, and Stork, Pattern Classification, 2nd Ed., 2002

Tan, Steinbach, and Kumar, Introduction to Data Mining, Addison-Wesley, 2005.

**Data repositories**

Kaggle.com

**Algorithms**

Thesmartbridgeteachable.com

1. **Appendix**

Source Code :

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

%matplotlib inline

Importing The DataSet

dataset=pd.read\_excel(r"C:\Users\Om Sai Ram\Downloads\Concrete\_Data.xls")

#Simplifying Column names, since they appear to be too lengthy

req\_col\_names = ["Cement", "BlastFurnaceSlag", "FlyAsh", "Water", "Superplasticizer",

"CoarseAggregate", "FineAggregare", "Age", "CC\_Strength"]

curr\_col\_names = list(dataset.columns)

mapper = {}

for i, name in enumerate(curr\_col\_names):

mapper[name] = req\_col\_names[i]

dataset = dataset.rename(columns=mapper)

len(dataset)

dataset

dataset.describe()

Data Visualization

#Checking the pairwise relations of Features

sns.pairplot(dataset)

plt.show()

#There seems to be no high correlation between independant variables (features). This can be further confirmed by plotting the Pearson Correlation coefficients between the features.

corr = dataset.corr()

sns.heatmap(corr, annot=True, cmap='Blues')

b, t = plt.ylim()

plt.ylim(b+0.5, t-0.5)

plt.title("Feature Correlation Heatmap")

plt.show()

#There are'nt any high correlations, except between Cement and Compressive Strength of Concrete. Which should be the case for strength.

EDA

ax = sns.distplot(dataset.CC\_Strength)

ax.set\_title("Compressive Strength Distribution")

fig, ax = plt.subplots(figsize=(10,7))

sns.scatterplot(y="CC\_Strength", x="Cement", hue="Water", size="Age", data=dataset, ax=ax, sizes=(20, 200))

ax.set\_title("CC Strength vs (Cement, Age, Water)")

ax.legend(loc="upper left", bbox\_to\_anchor=(1,1))

plt.show()

Taking Care Of Missing Values

dataset.isnull().any()

Label Encoding

OneHotEncoding

#Separating Input Features and Target Variable.

x=dataset.iloc[:,0:8].values

y=dataset.iloc [:,8:9].values

Feature Scalling

from sklearn.preprocessing import StandardScaler

sc=StandardScaler()

x=sc.fit\_transform(x)

y=sc.fit\_transform(y)

#Standardizing the data i.e. to rescale the features to have a mean of zero and standard deviation of 1.

Splitting Data Into Train And Test

from sklearn.model\_selection import train\_test\_split

x\_train,x\_test,y\_train,y\_test=train\_test\_split(x,y,test\_size=0.2,random\_state=0)

Model Building

Training And Testing The Model

from sklearn.ensemble import RandomForestRegressor

reg = RandomForestRegressor(n\_estimators=100)

reg.fit(x\_train, y\_train)

ypred1= reg.predict(x\_test)

Evalution

from sklearn.metrics import r2\_score

accuracy = r2\_score(y\_test,y\_pred1)

accuracy