# ***Predicting Compressive Strenght Of Concrete***

### About our project :-

Our project mainly focuses on "Predicting the strenght of Cement". Concrete is a composite material composed of fine and coarse aggregate bonded together with a fluid cement(cement paste) that hardens (cures) over time. 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 and low cost. 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 long. This project aims to build a Machine Learning model to predict the compressive strength of concrete to greatly speed up prediction with high accuracy. A web application is build where the user can enter the required parameters and see the predicted results on Web Application.

### Purpose :-

Concrete plays a more vital role than you may think. Without it, there wouldn't be safe roads, safe travelling in general, comfortable roads or sturdy buildings to work in. This project focuses on the use of computational intelligence techniques, especially Random Forest, Artificial Neural Network (ANN) and Support Vector Machine (SVM), to analyze the prediction of concrete compressive strength, emphasizing accuracy and efficiency, and their potential to deal with experimental data. This study also aims to contribute to the knowledge of the application of computational models in the prediction of compressive strength of concrete, using machine learning and pre-processing methods such as GridsearchCV and cross-validation, comparing the obtained results with other studies in the available literature. Atlast the Strenght Of Concrete can be achieved.





### Existing Problems :-

Errors made during construction can include adding improper amounts of water to the concrete mix, inadequate consolidation, and improper curing can cause distress and deterioration of the concrete. Proper mix design, placement, and curing of the concrete, as well as an experienced contractor are essential to prevent construction errors from occurring. Construction errors can lead to some of the problems discussed later in this fact sheet such as scaling and cracking. Honeycombing and bugholes can be observed after construction. When critically saturated concrete (when 90% of the pore space in the concrete is filled with water) is exposed to freezing temperatures, the water in the pore spaces within the concrete freezes and expands, damaging the concrete. In addition, acidic substances in the surrounding soil and water can cause disintegration of the concrete surface due to a reaction between the acid and the hydrated cement.

### *Solution :-*

taking all things into consideration there should be a machine which gives the perfect strenght of concrete so that all the cracks lumps will be reduced as well as construction errors

So this is how it goes :-

Age

Result

Predict

Blast Furance Slag

Superplasticizer

Fine Aggregate

Fly Ash

Water

Coarse Aggregate

Cement

### Appendix : -

### source code

import numpy as np

from flask import Flask, request, jsonify, render\_template

import pandas as pd

import requests

import json

# NOTE: you must manually set API\_KEY below using information retrieved from your IBM Cloud account.

API\_KEY = "f4r9s-TGhqyKHhpTfD7tuIcC-Q60CnG51kvlHpN8qsFH"

token\_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey": API\_KEY, "grant\_type": 'urn:ibm:params:oauth:grant-type:apikey'})

mltoken = token\_response.json()["access\_token"]

header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}

# NOTE: manually define and pass the array(s) of values to be scored in the next line

app = Flask(\_name\_) # initializing a flask app

@app.route('/')# route to display the home page

def home():

return render\_template('home.html') #rendering the home page

@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']) # route to show the predictions in a web UI

def index():

cement = request.form["Cement"]

slag = request.form["Blast Furnace Slag"]

ash = request.form["Fly Ash"]

water = request.form["Water"]

plast = request.form["Superplasticizer"]

ca = request.form["Coarse Aggregate"]

fa = request.form["Fine Aggregate"]

age = request.form["Age"]

t = [[float(cement),float(slag),float(ash),float(water),float(plast),float(ca),float(fa),int(age)]]

payload\_scoring = {"input\_data": [{"field": ["Cement","Blast Furnace Slag","Fly Ash","Water","Superplasticizer","Coarse Aggregate","Fine Aggregate","Age"] , "values": t }]}

response\_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/0a03a4f4-576c-4396-b380-4bdd2b78053b/predictions?version=2021-06-05', json=payload\_scoring, headers={'Authorization': 'Bearer ' + mltoken})

print("Scoring response")

print(response\_scoring.json())

predictions = response\_scoring.json()

print(predictions['predictions'][0]['values'][0])

prediction = predictions['predictions'][0]['values'][0]

# showing the prediction results in a UI

return render\_template('result2.html',prediction\_text=prediction)

if \_name\_ == "\_main\_":

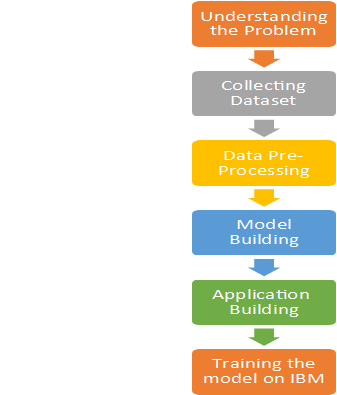
#app.run(host='127.0.0.1', port=8001, debug=True)

#app.run(debug=False) # running the app

app.run(debug=True) #local host 8080

### Theoretical Analysis :-

### Diagrammatic overview of the project



### 

### Hardware and Software Design :-

No essential hardware used apart from the computer and laptops and as far as Software goes we have used Anaconda which is a free and opensource distribution of ‘Python’ and ‘R’, editors such as Jupyter notebook and spyder, Watson Studio in IBM was used in the final stages of the project and finally HTML and CSS were used to build the web based application for the user to interact with the software finally created

### 

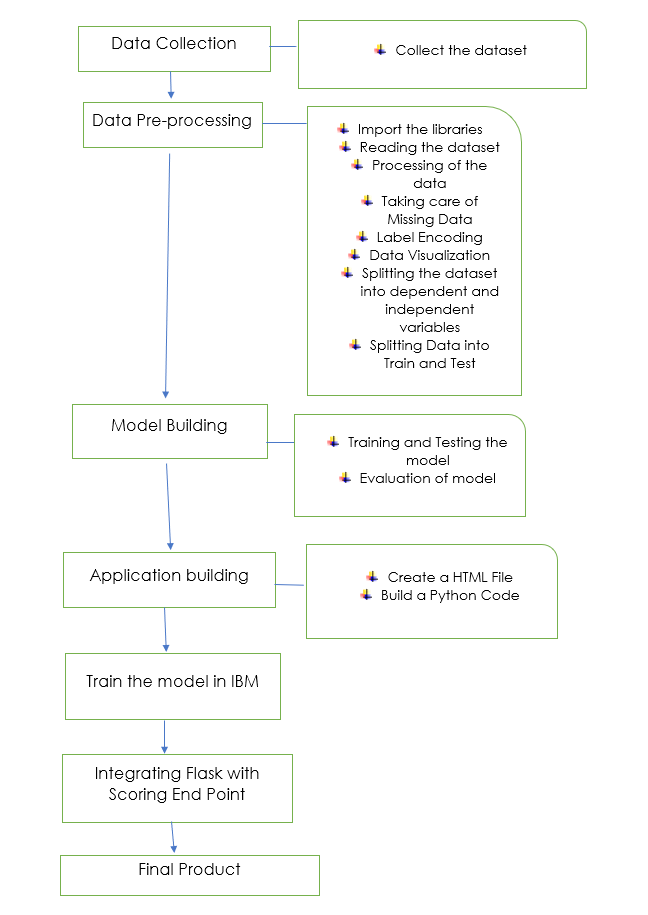
### Experimental Investigation :-

During the course of our project we tried to understand the underlying problems that arise do to in-appropriate concrete strength. Concrete being the material it is, the most popular and widely used material in constructions, it is not easy to make. In the making of concrete involvement of errors is quite often. A common concrete problem is scaling. Scaling occurs when the concrete surface breaks off to 2 inches and keeps peeling away. This normally occurs due to inadequate concrete strength. Interestingly, we found out that changing a parameter in making of concrete even by a slightest bit will effect the overall comprehensive strength of the output material. Hence in such a case the model that we built will be of very great help in foreseeing the strength of our concrete.

### Flowchart :-

### Diagram showing the control flow of the solution

As we can see here 1st data collection rakes place i.e. we 1st create a data set sutabile for our project and link it up. Next we import the libraries and read the dataset which is a crucial part. After that we take care of the missing data. Next data encoding and visulaization takes place were a lot of graphs of co-relation appears. And now we split the data into test and train. After splitting model building is done were we elevate the model and it gives us the best Regressor sutabile for our project. after that we have to do application building were we create a html pages in which user can enter the values and the predicted result. After all these things the main part we have to train our model in IBM. Integrating Flask with score end points is a crucial step were we have the source code. below is the flowchart of our project



### 

### *Advantages* *:-*

* The range of test force measurement become wider, more accurate, more stable.
* The control process is not affected by temperature and performance is more stable.

*Disadvantages :-*

* Low tensile strength.
* Concrete has Low specific strength.
* Concrete has Low Toughness.

### Applications :-

* The results show that the neural networks are very efficient in predicting the compressive strength of concrete with good accuracy.
* The application of this technology in predicting the compressive strength of concrete is expected to contribute to the assurance of concrete quality for manufacturing of optimal concrete.

### CONCLUSION :-

We have done the data collection, data Pre processing, Model Building and Application Building. We had done the Python program .Html program with the home page entering the values and we get the output. Created an IBM account. Finally we get the strength of the cement values in MPa

### FUTURE SCOPE :-

One way of reducing the wait time and reducing the number of combinations to try is to make use of digital simulations, where we can provide information to the computer about what we know and the computer tries different combinations to predict the compressive strength. This way we can reduce the number of combinations we can try physically and reduce the amount of time for experimentation. But, to design such software we have to know the relations between all the raw materials and how one material affects the strength. It is possible to derive mathematical equations and run simulations based on these equations, but we cannot expect the relations to be same in real-world. Also, these tests have been performed for many numbers of times now and we have enough real-world data that can be used for predictive modelling.

### Output :-

### Home page

### Page were you enter values



### *Final Result*

