CONCRETE COMPREHENSIVE STRENGTH PREDICTOR(CCSP)

USING MACHINE LEARNING

1.Purpose to development of CCSP

- It is very difficult to Calculate the concrete comprehensive strength by Manually.
- So, our predictor system helps the civil engineers to easily predict their concrete strength using features like amount of materials added to it.

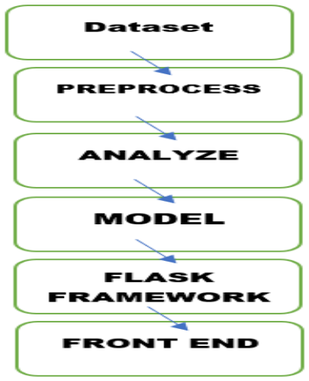
2.Product introduction

- Concrete is the most important material in civil engineering. The concrete compressive strength is a highly nonlinear function of age and ingredients.
- These ingredients include cement, blast furnace slag, fly ash, water, superplasticizer, coarse aggregate, and fine aggregate.
- It is very hard and almost impossible to Calculate the concrete comprehensive strength by Manually.
- With the help of machine learning and with proper preprocessing, we predict the comprehensive strength with the given features .

3.Product explanation

- The dataset is taken from the Kaggle platform.
- Convert the dataset into a dataframe using pandas. And Preprocess the dataset .Analyze the dataset using some data visualization libraries
- We use random forest regressor algorithm . And trained and tested the algorithm using our dataset . And optimize the model better accuracy.
- We created the framework using flask .To give the new input values to Machine learning model

4. Work Plan for development



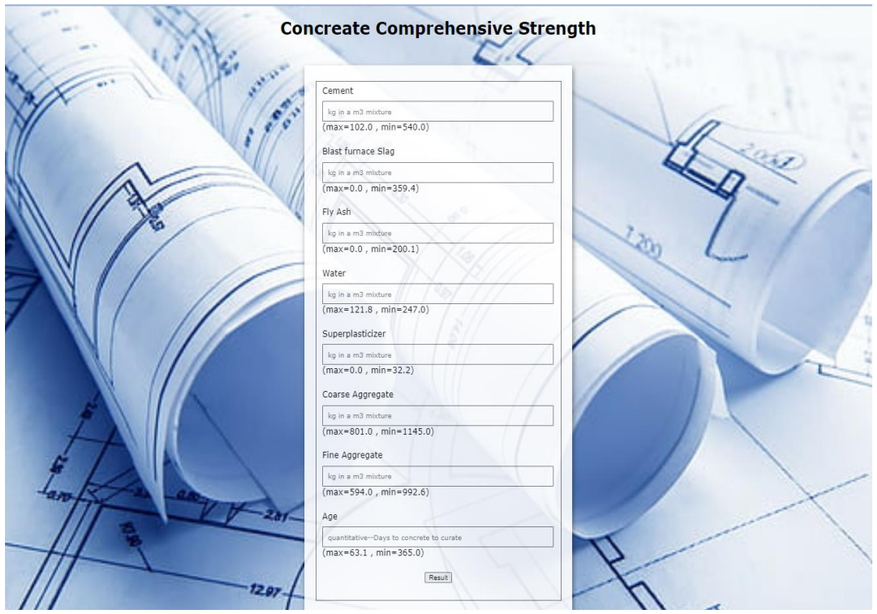
5.Dataset

	cement	blast_furnace_slag	fly_ash	water	superplasticizer	coarse_aggregate	fine_aggregate	age	concrete_compressive_strength
0	540.0	0.0	0.0	162.0	2.5	1040.0	676.0	28	79.99
1	540.0	0.0	0.0	162.0	2.5	1055.0	676.0	28	61.89
2	332.5	142.5	0.0	228.0	0.0	932.0	594.0	270	40.27
3	332.5	142.5	0.0	228.0	0.0	932.0	594.0	365	41.05
4	198.6	132.4	0.0	192.0	0.0	978.4	825.5	360	44.30

6.Flask framework

```
def home():
    return render_template("sanjeev.html")


# Bind predict function to URL
@concrete.route('/predict', methods=['POST'])
def predict():
    # Put all form entries values in a list
    features = [i for i in request.form.values()]
    # Convert features to array
    array_features = np.array(features)
    # Predict features
    prediction = model.predict([array_features])

    output = prediction

    # Check the output values and retrive the result with

    return render_template('sanjeev.html', result=output)


if __name__ == '__main__':
    concrete.run()
```

7.Finished product



8.Cost of product

- This is entirely build on opensource libraries with free of cost. So, our cost of the product is 0
- But , when we want to deploy this product in cloud services like aws or azure. It costs around 35$

9.Student involved

SANJEEV KUMAR M

201CS272

10.Outcome of product

- ✓ Product is presented in epoch'22 and won 3rd prize