Predicting Compressive Strength of Concrete

Using Random forest Regressor

Developed by: Yaddanapudi Vignesh

1. Introduction:

Machine learning methods have been successfully applied to many engineering disciplines. Prediction of the concrete compressive strength is important in terms of the desirability of concrete and its sustainability.

Concrete is the most widely used building material around the world due to its various advantages over other materials, e.g., integrity, durability, modularity, economy, etc. For better understanding the behaviour of concrete-made structures under external loadings and developing corresponding design methodologies, it is of great importance to study the mechanical properties of concrete. Among the various indices of properties of concrete, compressive strength is the most fundamental one since it is directly related to the safety of the structures, and it is required in the performance determination of the structures during the whole life-cycle, from new structural design to old structural assessment. However, as it is known to all, concrete is made up of different components, e.g., coarse/fine aggregates, cement pastes, additional mixtures, etc., and these components are randomly distributed over the entire concrete matrix. Such a complicated system makes it a great challenge to accurately predict the compressive strength of concrete material.

Generally, the most direct way to obtain the compressive strength of concrete is through physical experiments. Usually the cubic or cylinder specimens were prepared according to certain designed mixture ratio and then cured for a required time. After that by using the compressive test instrument the compressive strength can be obtained easily. However, this approach is costly in both time and money, thus the working efficiency will be very low. Differing from the traditional experimental ways, also some empirical regression methods are proposed to predict the concrete compressive strength with the given designed mixture ratio of different components in concrete.

Unfortunately, the concrete mixture and compressive strength shows a strongly nonlinear relation, thus it is difficult to derive an accurate regression expression for this problem.The third way to capture the concrete behaviour is numerical simulation. However, as mentioned before, the coupling of randomness and nonlinearity makes it different to accurately reproduce the concrete behaviour.

On the other hand, with the development of artificial intelligence (AI) in recent years, it is a trend to use machine learning (ML) techniques to predict the concrete compressive strength. ML is a branch of AI, and can be used for several objectives, e.g., classification, regression, clustering, etc. Predicting concrete compressive strength is just one application of the regression function of ML. Compared with other traditional regression method, ML adopts certain algorithms that can learn from the input data themselves and gives highly accurate results for the output data, which shows an obvious advantage over the traditional regression methods .

1.1 Overview:

Accurate prediction of the compressive strength of concrete has been a concern since these properties are often required by design codes. The emergence of new concrete mixtures and applications has motivated researchers to pursue reliable models for predicting concrete strength. Empirical and statistical models, such as linear and nonlinear regression, have been widely used. However, these models require laborious experimental work to develop, and can provide inaccurate results when the relationships between concrete properties and mixture composition and curing conditions are complex. To overcome such drawbacks, several Machine Learning (ML) models have been proposed as an alternative approach for predicting the compressvie strength of concrete. The present study examines ML models for predicting compressive strength of concrete, including artificial neural networks, Random forest regression, decision trees, and evolutionary algorithms. The application of each model and its performance are critically discussed and analyzed, thus identifying practical recommendations, current knowledge gaps, and needed future research.

1.2Purpose:

According to the evaluation metrics, such as the correlation coefficient, root mean squared error, and mean absolute error, the fuzzy logic method makes better predictions than any other regression method. Our aim from the project is to make use of pandas, matplotlib, & seaborn libraries from python to extract the libraries for machine learning for the predicting compressive strength of concrete. Train the model by using machine learning algorithms.and in the

end, to predict the compressive strength of concrete using voting ensemble techniques of combining the predictions from multiple machine learning algorithms and withdrawing the conclusions

2. LITERATURE SURVEY :

Data mining is the process of analyzing data from different perspectives and extracting useful knowledge from it. It is the core of knowledge discovery process. The various steps involved in extracting knowledge from raw data as depicted . Different data mining techniques include classification, clustering, association rule mining, prediction and sequential patterns, neural networks, regression etc.Regression is another important and broadly used statistical and machine learning tool. The key objective of regression-based tasks is to predict output labels or responses which are continues numeric values, for the given input data. The output will be based on what the model has learned in training phase. Basically, regression models use the input data features (independent variables) and their corresponding continuous numeric output values (dependent or outcome variables) to learn specific association between inputs and corresponding outputs.

2.1Existing Problem

The previous models have high time complexity and space complexity whereas this model is constrained with the lot of advantages and with a higher accuracy than any other model already proposed. In this model we used Machine learning algorithm named Random Forest Regressor which give an accuracy more then the previously predicted problem and there is an user friendly user interface to check the compressive strength of concrete ,and lot of the previous models haven't included the UI (User interface) which is so friendly and convenient for the users.

2.2Proposed Solution

Machine learning(Random Forest):

Using a Decision Tree Regressor has improved our performance, we can further improve the performance by ensembling more trees. Random Forest Regressor trains randomly initialized trees with random subsets of data sampled from the training data, this will make our model more robust.

Random forests are an ensemble model of machine learning with their roots in Decision Trees.These decision trees individually may over fit the data set and thus they come together to form a much stronger model.Numerous decision trees are first built

and based on these by performing random sampling of the attributes,a group of decision trees are assembled to form a Random Forest.A Random Forest is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees.After preparing the data, we can fit different models on the training data and compare their performance to choose the algorithm with good performance. As this is a regression problem, we can use RMSE (Root Mean Square Error) and R^2 score as evaluation metrics.

And also we have created an UI using the Flask for the loan status prediction, this UI will allow the users to predict the loan status very easily and the User interface is user friendly not at least one complication in using the interface, and it can be used just by entering some necessary details into the UI in real time it'll give the predicted value

3. THEORETICAL ANALYSIS

While selecting the algorithm that gives an accurate prediction we gone through lot of algorithms which gives the results abruptly accurate and from them we selected only one algorithm for the prediction problem that is RandomForest Regression.Random forest is a Supervised Learning algorithm which uses ensemble learning method for classification and regression.

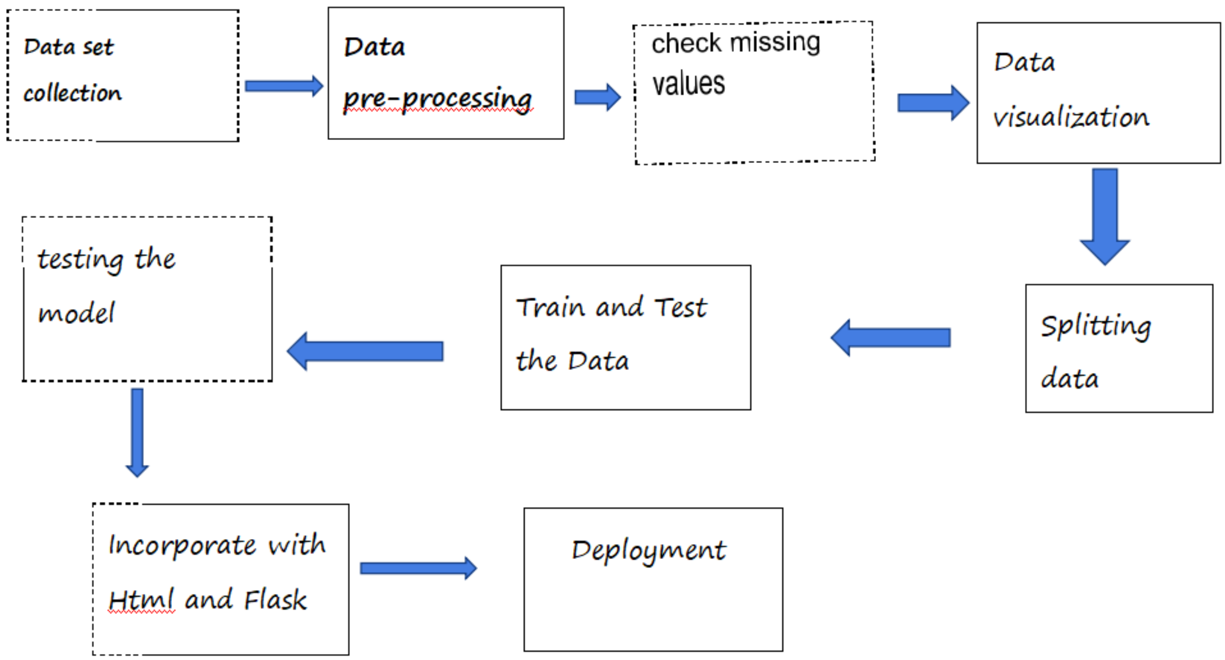
Random forest is a bagging technique and not a boosting technique. The trees in random forests are run in parallel. There is no interaction between these trees while building the trees.It operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

A random forest is a meta-estimator (i.e. it combines the result of multiple predictions) which aggregates many decision trees, with some helpful modifications:

1. The number of features that can be split on at each node is limited to some percentage of the total (which is known as the hyperparameter). This ensures that the ensemble model does not rely too heavily on any individual feature, and makes fair use of all potentially predictive features.

2. Each tree draws a random sample from the original data set when generating its splits, adding a further element of randomness that prevents overfitting.

3.1 Block Diagram



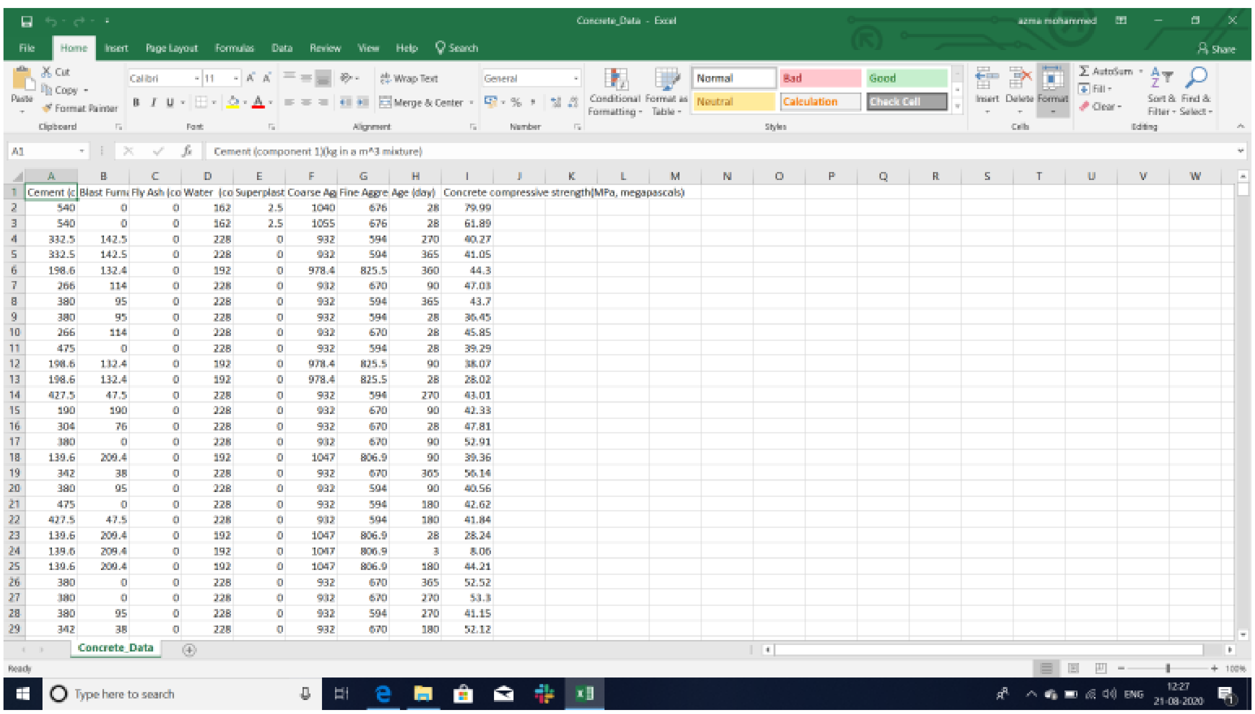
3.2. Software Designing

- Jupyter Notebook Environment
- Spyder Ide
- Machine Learning Algorithms
- Python (pandas, numpy, matplotlib, seaborn, sklearn)
- HTML
- Flask

We developed this prediction of compressive strength of concrete by using the Python language which is a interpreted and high level programming language and usng the Machine Learning algorithms. for coding we used the Jupyter Notebook environment of the Anaconda distributions and the Spyder, it is an integrated scientific programming in the python language. For creating an user interface for the prediction we used the Flask. It is a micro web fr amework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions, and a scripting language to create a webpage in HTML by creating the templates to use in th functions of the Flask andHTML.

4. EXPERIMENTAL INVESTIGATION :

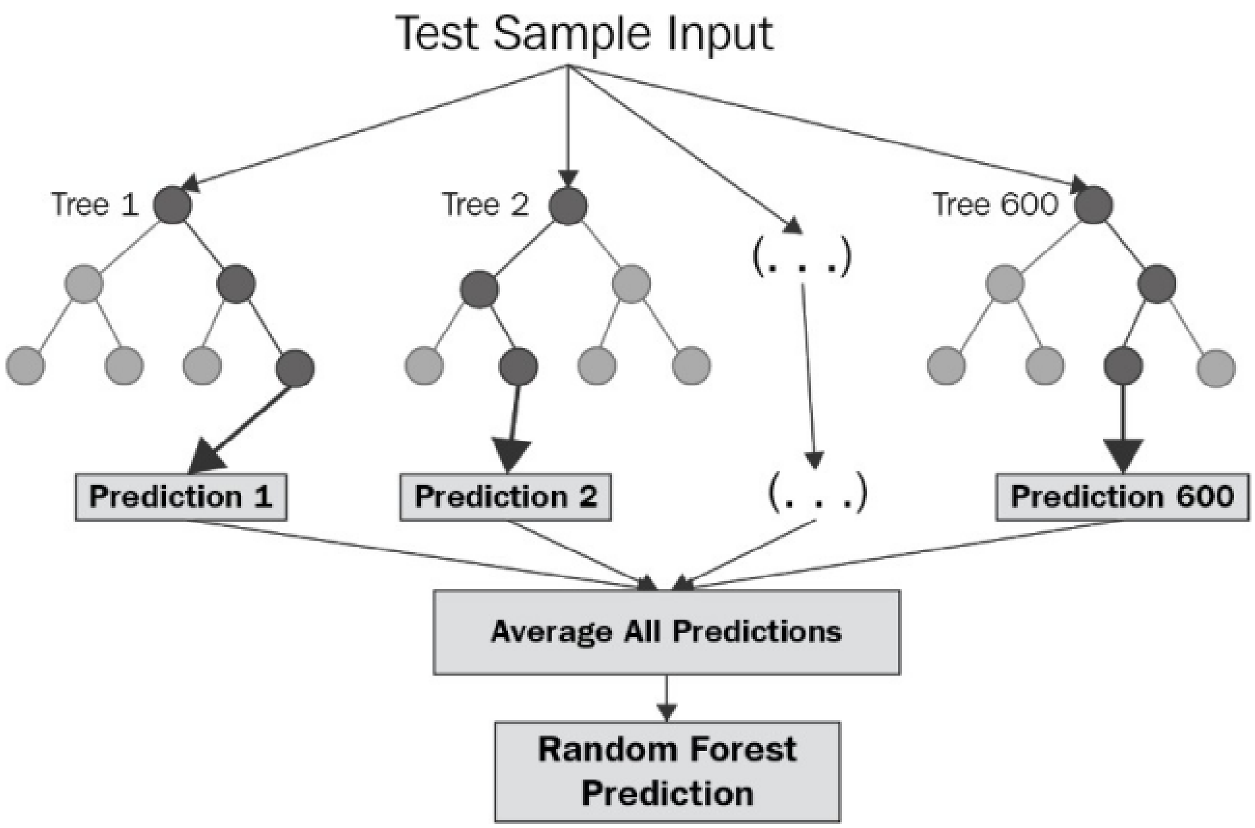
In this paper , the dataset we used is derived from https://archive.ics.uci.edu/ml/machine-learning-databases/concrete/compressive/ It contains more than 1030 instances with 9 attributes.There are 8 input variables and 1 output variable.Those attributes were shown below in the screenshot of the data set we used.



Cement (c	Blast Furn	Fly Ash (co	Water (co	Superplast	Coarse Ag	Fine Aggre	Age (day)	Concrete compressive strength(MPa, megapascals)
540	0	0	162	2.5	1040	676	28	79.99
540	0	0	162	2.5	1055	676	28	61.89
332.5	142.5	0	228	0	932	594	270	40.27
332.5	142.5	0	228	0	932	594	365	41.05
198.6	132.4	0	192	0	978.4	825.5	360	44.3
266	114	0	228	0	932	670	90	47.03
380	95	0	228	0	932	594	365	43.7
380	95	0	228	0	932	594	28	36.45
266	114	0	228	0	932	670	28	45.85
475	0	0	228	0	932	594	28	39.29
198.6	132.4	0	192	0	978.4	825.5	90	38.07
198.6	132.4	0	192	0	978.4	825.5	28	28.02
427.5	47.5	0	228	0	932	594	270	43.01
190	190	0	228	0	932	670	90	42.33
304	76	0	228	0	932	670	28	47.81
380	0	0	228	0	932	670	90	52.91
139.6	209.4	0	192	0	1047	806.9	90	39.36
342	38	0	228	0	932	670	365	56.14
380	95	0	228	0	932	594	90	40.56
475	0	0	228	0	932	594	180	42.62
427.5	47.5	0	228	0	932	594	180	41.84
139.6	209.4	0	192	0	1047	806.9	28	28.24
139.6	209.4	0	192	0	1047	806.9	3	8.06
139.6	209.4	0	192	0	1047	806.9	180	44.21
380	0	0	228	0	932	670	365	52.52
380	0	0	228	0	932	670	270	53.3
380	95	0	228	0	932	594	270	41.15
342	38	0	228	0	932	670	180	52.12

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.

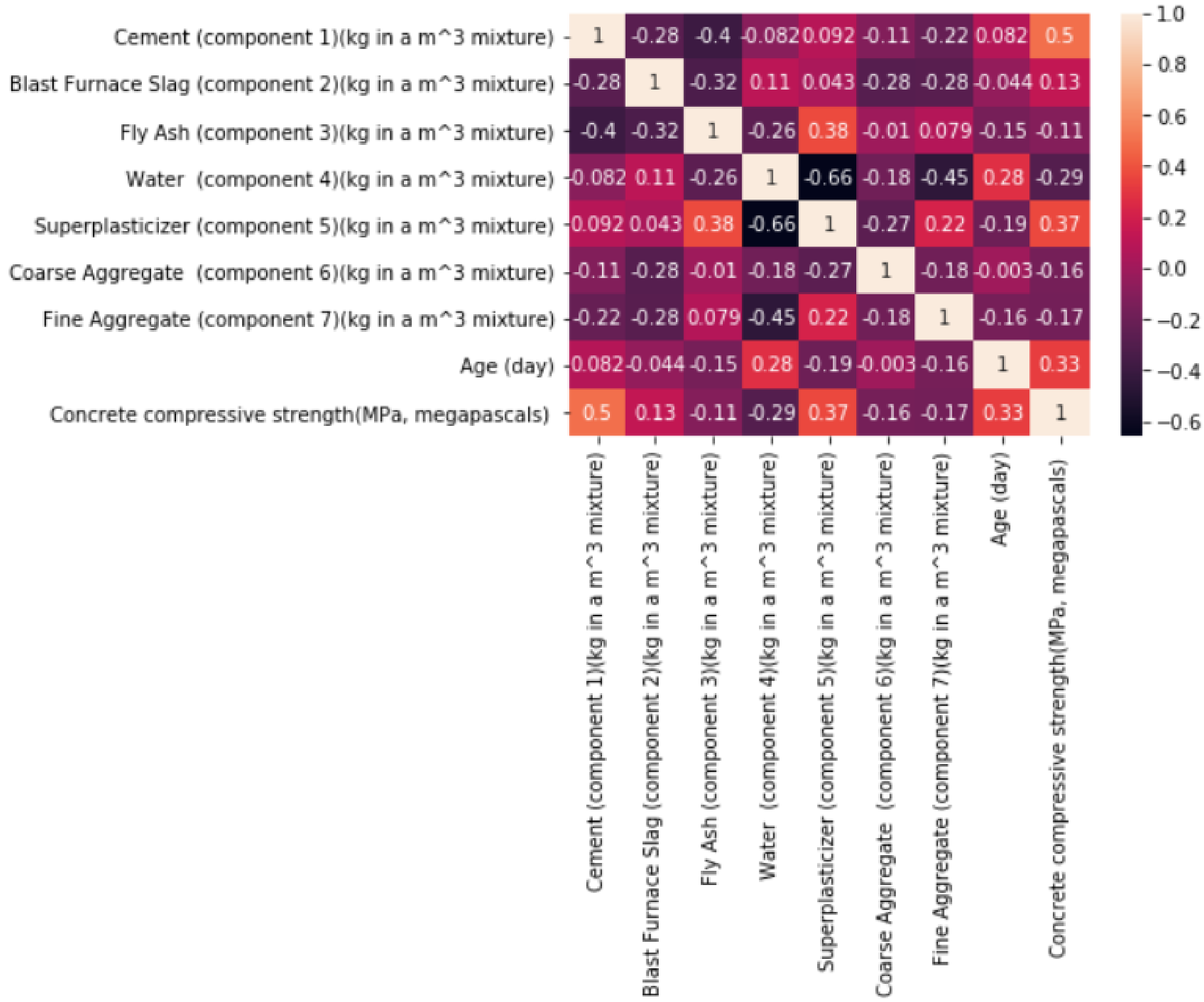
5. Flowchart:



6. Result:

In this paper, the Random Forest Regression algorithm is used to predict its performance, and compared with another three machine learning methods namely the Multi regression, decision tree, and the ANN Algorithm. The obtained results are displayed in Table below . The results show that, the performance of Random forest have comparable performance than that of multi regression,ANN and decision tree, so the Random still performs the best, with an accuracy of 92%, higher than remaining algorithms.

Algorithm used	Accuracy
multi regression	63%
descision tree	78%
Random Forest	92%
ANN	82%



7. ADVANTAGES AND DISADVANTAGES

Advantages:

- EasyandsimpleUserInterfacefor the people who is going to evaluate the concrete strength.
- Random forest give the accurate result of the prediction upto 92% which is the algorithm we used for prediction.
- It is widely used for prediction of compressive strength of concrete.
- It is composed using the HTML and Python for the web usage in real time.
- It can work in real time and predict as soon as the necessary details for prediction are given to the model

Disadvantages:

- Gives only 92% accuracy for the concrete strength prediction.
- Although random forest can be used for both classification and regression tasks, it is not more suitable for Regression tasks.
- Needs more than a single value for the prediction.

8. Applications:

- it is widely used for predicting the compressive strength of concrete and mainly used in constructions.
- while preparing concrete it is hard to take how much water and ash etc., to prepare it and finding it's compressive strength.
- The prediction results of concrete compressive strength based on machine learning methods are seriously influenced by input variables and model parameters.

9. Conclusion:

We have analysed the Compressive Strength Data and used Machine Learning alorithm to Predict the Compressive Strength of Concrete. We have used Multi Linear Regression and its variations, Decision Trees and Random Forests to make predictions and compared their performance. Random Forest Regressor has the lowest RMSE,higest accuracy 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.

10. FUTURE SCOPE :

In future the RandomForest algorithm can be applied on other data sets available for predicting compressive strength of concrete to further investigate its accuracy. A rigorous analysis of other machine learning algorithms other than these can also be done in future to investigate the power of machine learning algorithms for concrete strength prediction. In further study, we will try to conduct experiments on larger data sets or try to tune the model so as to achieve the state -of-art performance of the model and a great UI support system making it complete web application model.

11. BIBLIOGRAPHY

1. https://www.kaggle.com/pavanraj159/concrete-compressive-strength-data-set
2. 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.
3. Zain M.F.M., Suhad M. Abd, Hamid R. and Jamil M., “Potential for Utilizing Concrete Mix Properties to Predict Strength at Different Ages”. Journal of Applied sciences,Vol.10(22),2010,pp.2831-2838.

4. Hasan M.M. and Kabir A., “Prediction of Compressive Strength of Concrete from Early Age Test Result”. Proceedings of 4th Annual Paper Meet and 1st Civil Engineering Congress, Dhaka,Bangladesh,December 22-24,2011,pp.1-7.
5. P.H. Bischoff, S. Perry**Compressive behaviour of concrete at high strain rates** Mater. Struct., 24 (6) (1991), pp. 425-450
6. M. Lessard, O. Challal, P.-C. Aticin**Testing high-strength concrete compressive strength** ACI Mater. J., 90 (4) (1993), pp. 303-307

APPENDIX

Html:

```
<!DOCTYPE html>
<html >
<!--From https://codepen.io/frytyler/pen/EGdtg-->
<head>
  <meta charset="UTF-8">
  <title>Machine Learning</title>
  <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>
<link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">

<style>
.login{
top: 20%;
}
</style>
</head>
<body>
 <div class="login">
  <h1>Compressive Strength of Concrete Prediction</h1>

    <!-- Main Input For Receiving Query to our ML -->
   <form action="{{ url_for('y_predict')}}"method="post">
     <input type="text" name="cement" placeholder="Cement (kg in a m^3 mixture)" required="required" />
       <input type="text" name="blast_furnace_slag" placeholder="Blast Furnace Slag (kg in a m^3 mixture)" required="required" />
   <input type="text" name="fly_ash" placeholder="Fly Ash (kg in a m^3 mixture)" required="required" />
       <input type="text" name="water" placeholder="Water (kg in a m^3 mixture)" required="required" />
       <input type="text" name="superplasticizer" placeholder="Superplasticizer (kg in a m^3 mixture)" required="required" />
```

```html
        <input type="text" name="coarse_aggregate" placeholder="Coarse Aggregate (kg in a m^3 mixture)" required="required" />
        <input type="text" name="fine_aggregate" placeholder="Fine Aggregate (kg in a m^3 mixture) " required="required" />
        <input type="text" name="age" placeholder="Age (day)" required="required" />
        <button type="submit" class="btn btn-primary btn-block btn-large">Find Out Strength!     </button>

    </form>
   <br>
   <br>
   {{ prediction_text }}
 </div>
</body>
</html>
```

CSS:

```css
@import url(https://fonts.googleapis.com/css?family=Open+Sans);
.btn { display: inline-block; *display: inline; *zoom: 1; padding: 4px 10px 4px; margin-bottom: 0; font-size: 13px; line-height: 18px; color: #333333; text-align: center;text-shadow: 0 1px 1px rgba(255, 255, 255, 0.75); vertical-align: middle; background-color: #f5f5f5; background-image: -moz-linear-gradient(top, #ffffff, #e6e6e6); background-image: -ms-linear-gradient(top, #ffffff, #e6e6e6); background-image: -webkit-gradient(linear, 0 0, 0 100%, from(#ffffff), to(#e6e6e6)); background-image: -webkit-linear-gradient(top, #ffffff, #e6e6e6); background-image: -o-linear-gradient(top, #ffffff, #e6e6e6); background-image: linear-gradient(top, #ffffff, #e6e6e6); background-repeat: repeat-x; filter: progid:dximagetransform.microsoft.gradient(startColorstr=#ffffff, endColorstr=#e6e6e6, GradientType=0); border-color: #e6e6e6 #e6e6e6 #e6e6e6; border-color: rgba(0, 0, 0, 0.1) rgba(0, 0, 0, 0.1) rgba(0, 0, 0, 0.25); border: 1px solid #e6e6e6; -webkit-border-radius: 4px; -moz-border-radius: 4px; border-radius: 4px; -webkit-box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px 2px rgba(0, 0, 0, 0.05); -moz-box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px 2px rgba(0, 0, 0, 0.05); box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px 2px rgba(0, 0, 0, 0.05); cursor: pointer; *margin-left: .3em; }
.btn:hover, .btn:active, .btn.active, .btn.disabled, .btn[disabled] { background-color: #c97f7f; }
.btn-large { padding: 9px 14px; font-size: 15px; line-height: normal; -webkit-border-radius: 5px; -moz-border-radius: 5px; border-radius: 5px; }
.btn:hover { color: #9b1e1e; text-decoration: none; background-color: #571a1a; background-position: 0 -15px; -webkit-transition: background-position 0.1s linear; -moz-transition: background-position 0.1s linear; -ms-transition: background-position 0.1s linear; -o-transition: background-position 0.1s linear; transition: background-position 0.1s linear; }
.btn-primary, .btn-primary:hover { text-shadow: 0 -1px 0 rgba(0, 0, 0, 0.25); color: #e4dc77; }
.btn-primary.active { color: rgba(51, 143, 51, 0.801); }
.btn-primary { background-color: #69d389; background-image: -moz-linear-gradient(top, #72de6e, #3f9666); background-image: -ms-linear-gradient(top, #79c739, #30dba8); background-image: -webkit-gradient(linear, 0 0, 0 100%, from(#036d2f), to(#6c296e)); background-image: -webkit-linear-gradient(top, #4fa756, #4fd44a); background-image: -o-linear-gradient(top, #22be44, #70b835); background-image: linear-gradient(top, #371a6d, #51165f); background-repeat: repeat-x; filter: progid:dximagetransform.microsoft.gradient(startColorstr=#702544, endColorstr=#58073d, GradientType=0);  border: 1px solid #37bc54; text-shadow: 1px 1px 1px rgba(139, 224, 83, 0.4); box-shadow: inset 0 1px 0 rgba(53, 179, 184, 0.2), 0 1px 2px rgba(29, 112, 145, 0.507); }
.btn-primary:hover, .btn-primary:active, .btn-primary.active, .btn-primary.disabled, .btn-primary[disabled] { filter: none; background-color: #4a77d4; }
.btn-block { width: 100%; display:block; }

* { -webkit-box-sizing:border-box; -moz-box-sizing:border-box; -ms-box-sizing:border-box; -o-box-sizing:border-box; box-sizing:border-box; }
html { width: 100%; height:100%; overflow:hidden; }
body {
    width: 100%;
    height:100%;
    font-family: 'Open Sans', sans-serif;
    background: #88eb37;
    color: rgb(150, 76, 134);
    font-size: 18px;
    text-align:center;
```

```
    letter-spacing:1.2px;
    background: -moz-radial-gradient(0% 100%, ellipse cover, rgba(135, 86, 145, 0.4) 10%,rgba(138, 104, 50, 0) 40%),-moz-linear-gradient(top, rgba(62, 77, 148, 0.25) 0%, rgba(42,60,87,.4) 100%), -moz-linear-gradient(-45deg,  #4d2223 0%, #531849 100%);
    background: -webkit-radial-gradient(0% 100%, ellipse cover, rgba(67, 58, 88, 0.4) 10%,rgba(83, 57, 83, 0) 40%), -webkit-linear-gradient(top, rgba(71, 119, 138, 0.25) 0%,rgba(68, 34, 65, 0.4) 100%), -webkit-linear-gradient(-45deg,  #22635f 0%,#3ca78f 100%);
    background: -o-radial-gradient(0% 100%, ellipse cover, rgba(104,128,138,.4) 10%,rgba(138,114,76,0) 40%), -o-linear-gradient(top,  rgba(36, 27, 43, 0.25) 0%,rgba(81, 163, 159, 0.4) 100%), -o-linear-gradient(-45deg,  #3b2964 0%,#29d46b 100%);
    background: -ms-radial-gradient(0% 100%, ellipse cover, rgba(104,128,138,.4) 10%,rgba(138,114,76,0) 40%), -ms-linear-gradient(top, rgba(66, 136, 84, 0.25) 0%,rgba(179, 224, 142, 0.4) 100%), -ms-linear-gradient(-45deg,  #670d10 0%,#092756 100%);
    background: -webkit-radial-gradient(0% 100%, ellipse cover, rgba(104,128,138,.4) 10%,rgba(138,114,76,0) 40%), linear-gradient(to bottom, rgba(173, 57, 219, 0.473) 0%,rgba(36, 60, 61, 0.4) 100%), linear-gradient(135deg,  #670d10 0%,#092756 100%);
    filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#3E1D6D', endColorstr='#092756',GradientType=1 );
}
.login {
    position: absolute;
    top: 40%;
    left: 50%;
    margin: -150px 0 0 -150px;
    width:400px;
    height:400px;
}
.login h1 { color: #fff; text-shadow: 0 0 10px rgba(0,0,0,0.3); letter-spacing:1px; text-align:center; }
input {
    width: 100%;
    margin-bottom: 10px;
    background: rgba(87, 54, 107, 0.3);
    border: none;
    outline: none;
    padding: 10px;
    font-size: 13px;
    color: rgb(255, 255, 255);
    text-shadow: 1px 1px 1px rgba(119, 61, 61, 0.3);
    border: 1px solid rgba(67, 165, 182, 0.3);
    border-radius: 4px;
    box-shadow: inset 0 -5px 45px rgba(172, 63, 63, 0.2)), 0 1px 1px rgba(255,255,255,0.2);
    -webkit-transition: box-shadow .5s ease;
    -moz-transition: box-shadow .5s ease;
    -o-transition: box-shadow .5s ease;
    -ms-transition: box-shadow .5s ease;
    transition: box-shadow .5s ease;
}
input:focus { box-shadow: inset 0 -5px 45px rgba(145, 69, 69, 0.4), 0 1px 1px rgba(46, 25, 25, 0.2); }
```

App.py:

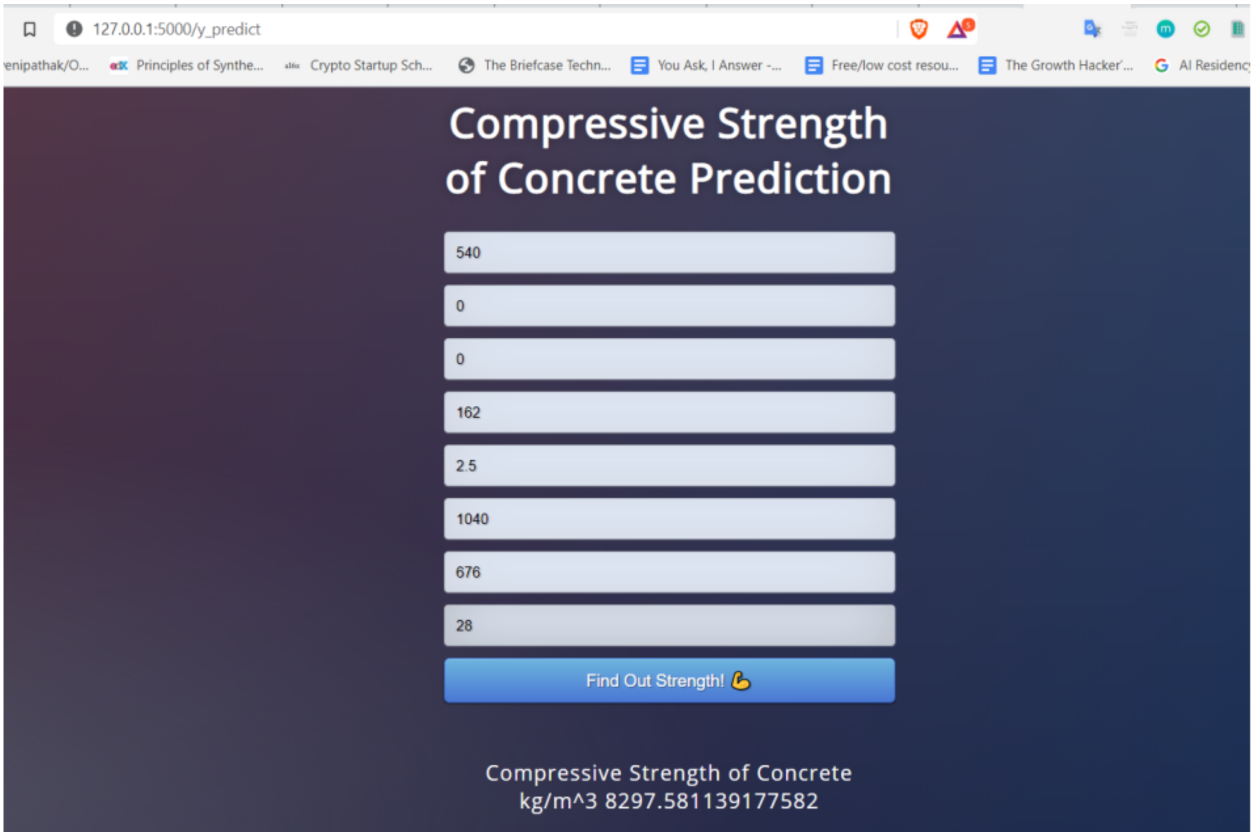
```
import numpy as np
from flask import Flask, request, jsonify, render_template
import pickle

app = Flask(__name__)
model = pickle.load(open('linear.pkl', 'rb'))
@app.route('/')
def home():
    return render_template('index.html')
@app.route('/y_predict',methods=['POST'])
def y_predict():
    '''
```

```
    For rendering results on HTML GUI
    '''
    X_test = [[float(X) for X in request.form.values()]]


    prediction = model.predict(X_test)
    print(prediction)
    output=prediction[0]
    return render_template('index.html', prediction_text='Compressive Strength of Concrete kg/m^3 {}'.format(output))
@app.route('/predict_api',methods=['POST'])
def predict_api():
    '''
    For direct API calls trought request
    '''
    data = request.get_json(force=True)
    prediction = model.y_pred_list([np.array(list(data.values()))])
    output = prediction[0]
    return jsonify(output)
if __name__ == "__main__":
    app.run(debug=True)
```

Output:


127.0.0.1:5000/y_predict
enipathak/O...
Principles of Synthe...
Crypto Startup Sch...
The Briefcase Techn...
You Ask, I Answer -...
Free/low cost resou...
The Growth Hacker'...
AI Residenc
Compressive Strength of Concrete Prediction
540
0
0
162
2.5
1040
676
28
Find Out Strength!
Compressive Strength of Concrete kg/m^3 8297.581139177582