International Journal of Science and Research (IJSR)
ISSN: 2319-7064
ResearchGate Impact Factor (2018): 0.28 | SJIF (2019): 7.583

Impact of COVID-19 on Concrete Construction & Innovative Ways to make it Sustainable & Economical

Ankit Kumar Tripathi

Department of Civil Engineering, MNNIT Allahabad, Prayagraj, India

Abstract: ***There are numerous challenges in the post-COVID construction world. Right from the beginning stage of production of concrete to the completion of project accumulating curing stage, consequences will be of various forms like a deficit of manpower & time, need of properly planned site arrangements, altering ingredients type & proportion in basic construction materials, adoption of increased usage of special materials in an innovative way, introducing disinfectants & relying much on well trained, robust & generalized software models & algorithms for prediction tasks. This paper presents a brief insight into these baffling issues, with suggested feasible, economical & adaptable ways to cater to unfavourable outcomes. To minimize labour, cost, time, machines & power consumed in casting cubes, hardening, testing & then transporting to the site, a large labelled dataset fed machine learning algorithm is introduced & its comparison is made with 5 layers ANN based on prediction accuracy obtained by both on the validation dataset. The last section is oriented about adopting the best suitable type of curing process advanced with sensors technology.***

Keywords: Artificial Neural Network (ANN), Compressive Strength, Exploratory Data Analysis, Machine learning, Sensors Technology

1. Introduction

Post COVID world will have innumerable impacts on construction projects which leads to take a step in direction of doing something new to our plans of completing a project. This paper is in this direction of giving new ideas to perform construction works at the same pace as it was in pre COVID world. This paper emphasizes on different recommendations and ideas to be implemented on the project which gives an economic, sustainable, durable and safe structure. Word 'safe' here represents the safety of workers against any viral disease, reduce the chances of community spread of such viral disease in the future. Some ideas are explained in this paper including the use of statistical modeling & sensor technology to avoid the cost, time and human contact with one another.

2. Prediction of Compressive Strength Using Artificial Neural Network (ANN) & Machine Learning

Since, construction process involves concrete mix design & proper evaluation of cube testing, which involves lots of labour & wastage of materials & time to be matured. These three major things may not available in the post-COVID-19 world. A Concrete mix design requires extensive knowledge of codal provisions. Obtaining concrete with an appropriate strength, and optimized workability are the ultimate aim of concrete production stage. The process of concrete hardening and hydration are irreversible. Timely knowledge of concrete strength helps to schedule operations such as pre-stressing and formwork design. Here are nominal mix present already available in literature but for an urgent project and for getting a specified strength of the concrete we have to perform trial mixing which demands hundreds of cubes to be cast then checked for their strength and find the standard deviation of result which is very long and material consuming process. At the time of maturing, a tobermorite gel is formed, along with some secondary compounds that acts as binder between the fine and coarse aggregate. Water required for complete hydration ranges between 20% to 25% by weight, excluding water trapped in the pores. Hardening of concrete does not end in 28 days, but it's strength gaining is continued for a long period (in years) and in 28 days approx. 99 %strength is gained, which is taken as a characteristic strength of that mix. This methodology has many disadvantages and is labour- intensive to use. Overall, it takes a lot of time, experience, calculations, manpower & lots of wastage of materials in testing, which will already be in shortage in the post COVID world. It will also help to reduce machinery cost, power required, multistage inspection required & transportation cost of labour & material. Designing a concrete mix based on a machine learning algorithm is being introduced here. To develop a predictive model is our aim that could estimate compressive strength as an approximate function of mix proportion & inherent composition of materials that would be useful in reducing the empirical, labour-intensive nature of the ‘Trial-Batching’ approach, that are the basis of industrial practice nowadays in the pre-COVID world. The developed algorithm will evaluate the compressive strength of concrete based on eight input variables, the amount of cement, fly ash, blast furnace slag, superplasticizer, water, fine, and coarse aggregate & age of testing. The output equation has it's limitations and not present a perfect solution, yet that much vulnerability is also there in conventional methods as described in terms of standard deviation in IS-456 . However, it is a step on the way to the introduction of machine learning techniques for concrete mix design. In its present form, it can be a tool for a rough estimation of the concrete class.

The work done on the same is covered under the following points:-

International Journal of Science and Research (IJSR)
ISSN: 2319-7064
ResearchGate Impact Factor (2018): 0.28 | SJIF (2019): 7.583

2.1 Importing of Dataset & Statistical Analysis of Data

The dataset, obtained from various RMCs & already available by Cheng dataset containing 1030 rows (number of observations taken) & 9 columns out of which 8 are input variables, & the response variable is compressive strength in MPa.

The peek on dataset & it's statistical analysis from output window of code, has been shown:-

(In all the plots, abbreviations used are ; 'cmt' for cement, 'fly' for fly ash, 'plas' for superplasticizer, 'slg' for blast furnace slag, 'wat' for water, 'fa' for fine aggregate, 'ca' for coarse aggregate, 'age' for the age of testing & 'cst' for tested compressive strength.)

```
[1030 rows x 9 columns]
35.817961165048544
(1030, 9)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1030 entries, 0 to 1029
Data columns (total 9 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   cmt     1030 non-null   float64
 1   slg     1030 non-null   float64
 2   fly     1030 non-null   float64
 3   wat     1030 non-null   float64
 4   plas    1030 non-null   float64
 5   ca      1030 non-null   float64
 6   fa      1030 non-null   float64
 7   age     1030 non-null   int64
 8   cst     1030 non-null   float64
dtypes: float64(8), int64(1)
memory usage: 72.5 KB
None
     cmt    slg  fly    wat  plas      ca     fa  age    cst
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
5  266.0  114.0  0.0  228.0   0.0   932.0  670.0   90  47.03
6  380.0   95.0  0.0  228.0   0.0   932.0  594.0  365  43.70
7  380.0   95.0  0.0  228.0   0.0   932.0  594.0   28  36.45
8  266.0  114.0  0.0  228.0   0.0   932.0  670.0   28  45.85
9  475.0    0.0  0.0  228.0   0.0   932.0  594.0   28  39.29
```

```
10  198.6  132.4  0.0  192.0  0.0   978.4  825.5   90  38.07
11  198.6  132.4  0.0  192.0  0.0   978.4  825.5   28  28.02
12  427.5   47.5  0.0  228.0  0.0   932.0  594.0  270  43.01
13  190.0  190.0  0.0  228.0  0.0   932.0  670.0   90  42.33
14  304.0   76.0  0.0  228.0  0.0   932.0  670.0   28  47.81
15  380.0    0.0  0.0  228.0  0.0   932.0  670.0   90  52.91
16  139.6  209.4  0.0  192.0  0.0  1047.0  806.9   90  39.36
17  342.0   38.0  0.0  228.0  0.0   932.0  670.0  365  56.14
18  380.0   95.0  0.0  228.0  0.0   932.0  594.0   90  40.56
19  475.0    0.0  0.0  228.0  0.0   932.0  594.0  180  42.62
               cmt          slg  ...          age          cst
count  1030.000000  1030.000000  ...  1030.000000  1030.000000
mean    281.167864    73.895825  ...    45.662136    35.817961
std     104.506364    86.279342  ...    63.169912    16.705742
min     102.000000     0.000000  ...     1.000000     2.330000
25%     192.375000     0.000000  ...     7.000000    23.710000
50%     272.900000    22.000000  ...    28.000000    34.445000
75%     350.000000   142.950000  ...    56.000000    46.135000
max     540.000000   359.400000  ...   365.000000    82.600000
```

Figure 1: Statistical Analysis of Dataset

2.2 Importing all necessary libraries & modules & performing Exploratory Graphical Data Analysis

Exploratory data analysis (EDA) on the imported dataset is performed & deep insight to data from various plots, plotted using matplotlib & seaborn with their adequate syntax is found.

The view of some of them with explanation from particular plots:-

Box & Whisker plots showing median, 25th, 75th percentile & outliers in the dataset of values of different feature variables as well as response variable.

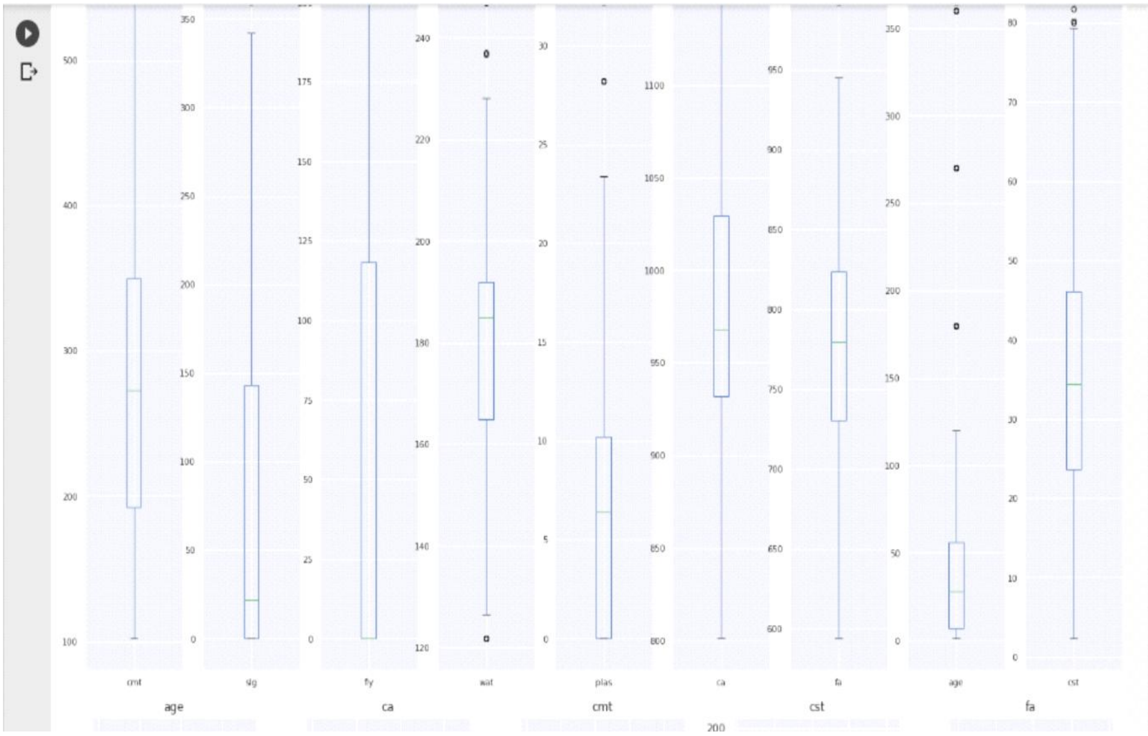
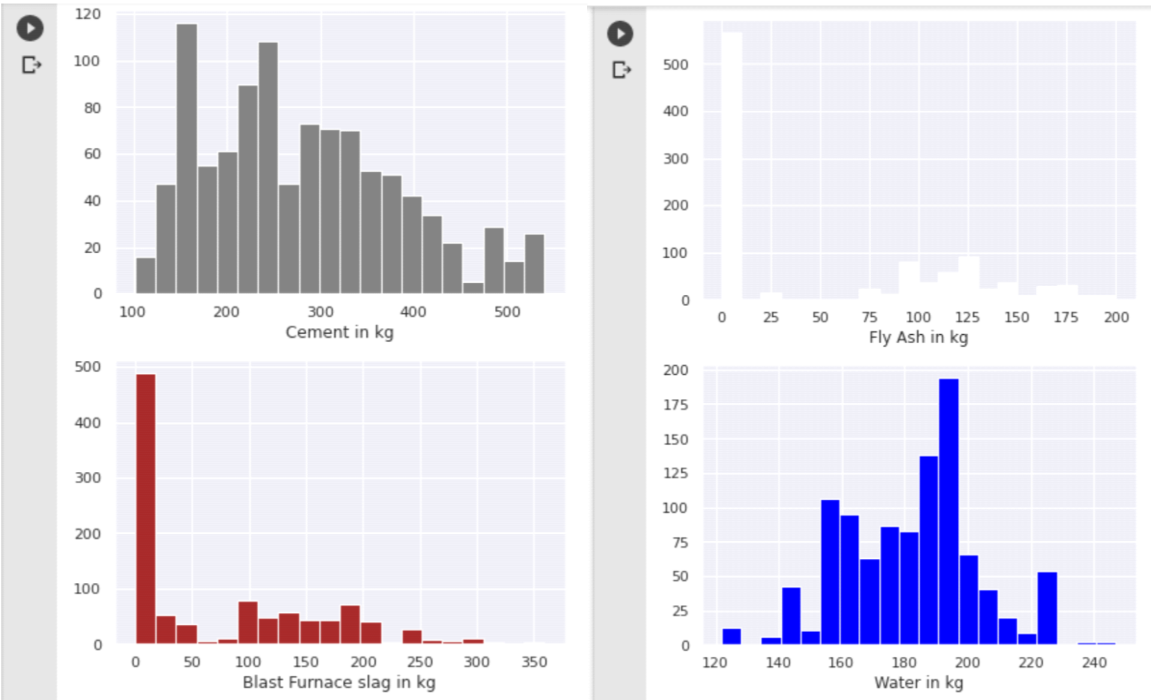


Figure 2: Box & Whisker Plots

Histograms showing variation of data taken for each variable on 20 numbers of bins at x scale. It mainly replicate what particular measure is in abundance & for which data takes lesser number of samples.



ISSN: 2319-7064
ResearchGate Impact Factor (2018): 0.28 | SJIF (2019): 7.583

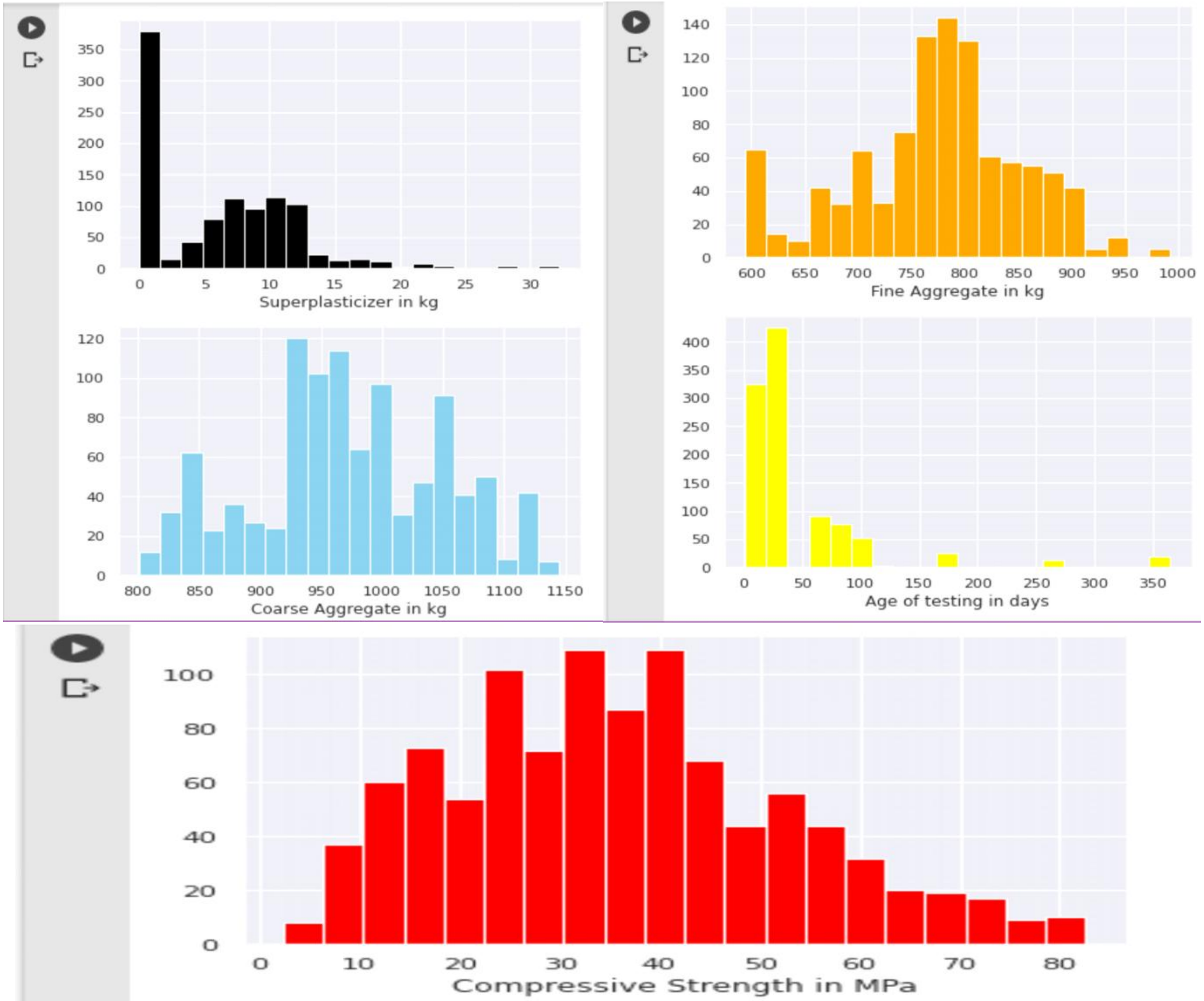


Figure 3: Histograms of feature &response variables

- Empirical Cumulative Distribution Function Curve to show cumulative frequency distribution for each variable, a perfect S shaped curve is obtained for compressive strength. It's like grain size distribution analysis curve which replicate that data is well distributed and helps to make sure that the model getting prepared will be fulfilling each range criteria & there is no abundance of one dominant range, hence no pipeline or data preparation is required for moving further in analysis.

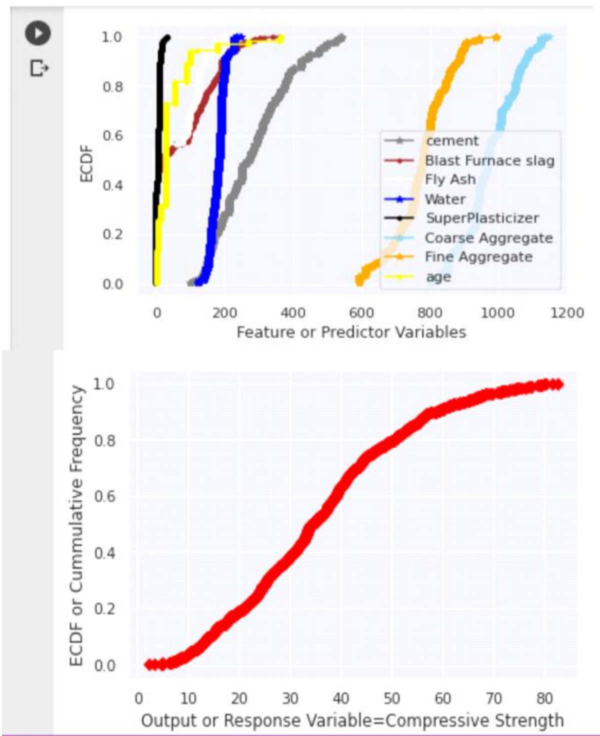


Figure 4: ECDF for feature &response variables

- Bee-Sworm plot of age variable showing that most samples are tested at 28 days but some are also tested at 3

International Journal of Science and Research (IJSR)
ISSN: 2319-7064
ResearchGate Impact Factor (2018): 0.28 | SJIF (2019): 7.583

days, 7, 156 & 365 days. It is also showing that high strength can be achieved at young age with adequate plasticizer & more cement but relatively less for most of the datapoints as concrete matured about 99% in 28 days but approx. 90% of it is achieved in 7 days only. **So age parameter is most critical one for a linear regression model & for this it has to be properly scaled by a machine learning point of view and should be given fewer weights associated with it by deep learning aspect**. Bee-Sworm plot of water v/s plasticizer shows that more the plasticizer is used less is the water required to achieve the desired degree of workability.

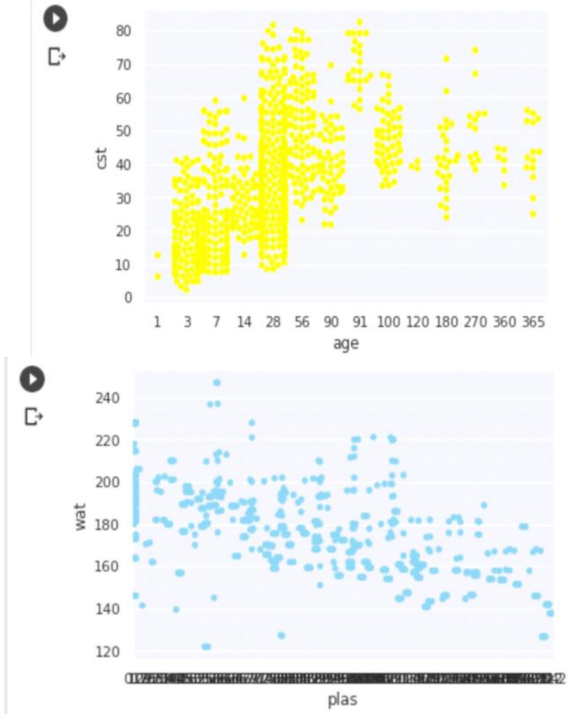


Figure 5: Bee Sworm plots

- 3-D variate relplots, first out of them shows that on using the higher amount of water then less will be the strength, but points with less the water used & higher the cement paste obtain maximum compressive strength, thus compressive strength depends largely on water-cement ratio, as per replication of this plot & second 3-D relplot pasted on the right side, with coarse aggregate on X-axis, compressive strength on Y axis & amount of water added is represented by coloured hue, it shows clearly that when coarse aggregate is very less more cement is needed & strength is comparatively less when it is very much also due to bad gradation cement is more needed but high compressive strength is achieved. These both are the well-validated point in the available literature as well.

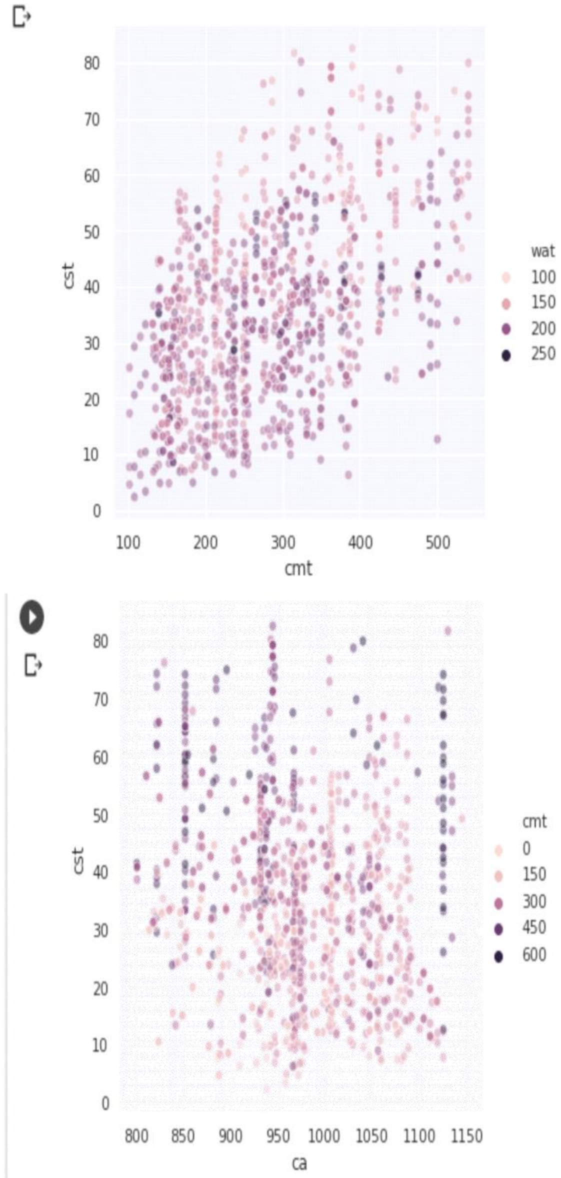


Figure 6: Relplots

Last plot is scattered plot matrix with 'kde' plots as diagonal elements of the matrix. It is the multivariate plot which provides visual insight of each variable's relation with itself & other variables. Diagonal elements of this scattering matrix are the one which shows the frequency of particular feature as histogram does,kde is the plot joining the peaks of all histogram bins, a better way of representation of frequencies.

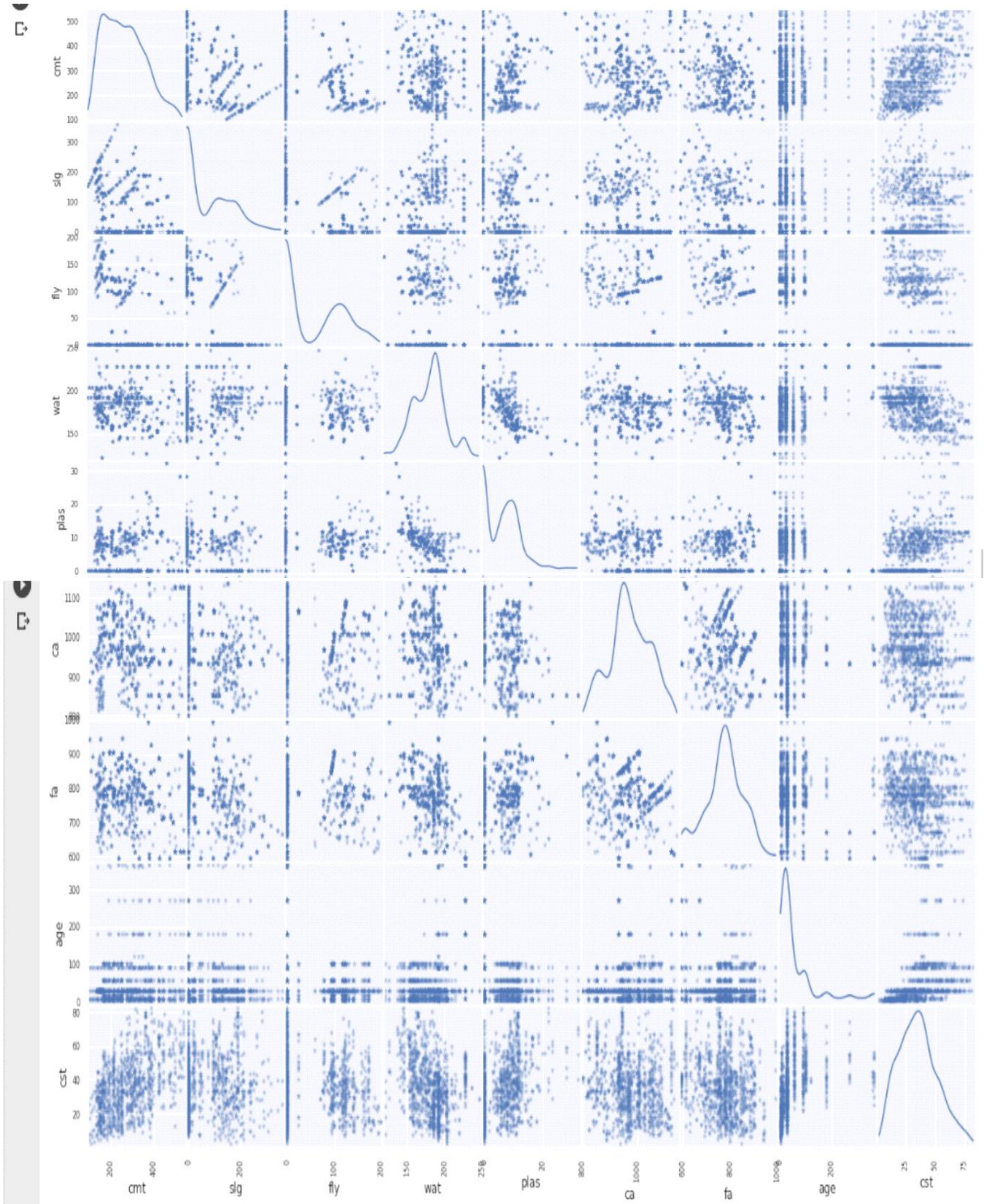


Figure 7: Scatter matrix

3.3 Application of Various Regression Techniques

The whole dataset is spitted into a training & testing dataset. Training data will test for accuracy score on testing dataset in terms of R^2 score. Results of Linear Regression, Ridge regression, Lasso Regression, Elastic Net Regression with & without hyperparameter tuning are applied & compared. ***It was found that results with hyperparameter tuning are more close to actual results in the prediction dataset***. The R squared is defined as R^2=(1-S/S'), where R is the residual sum of squares of predictions with respect to actual values & S' is the total sum of square of whole prediction dataset around it's mean.

The R^2 values with the name of model & tuning status are listed below as obtained from the console of code, which shows how beautifully machine learning has been successful in the estimation.

```
R2 of LinearRegression is: 0.6291213241527397
R2 of Ridge without hyperparameter tuning is: 0.5676566035458499
R2 of Lasso without hyperparameter tuning is: 0.6275785542129153
The best value of hyperparameter alpha is: {'alpha': 7000} R2 of Ridge after hyperparameter tuning is: 0.585021538554741
The best value of hyperparameter alpha is: {'alpha': -0.001} R2 of Lasso after hyperparameter tuning is: 0.5816183181768892
Tuned ElasticNet Alpha: {'elasticnet__l1_ratio': 1.0}
Tuned ElasticNet R squared: 0.5935519283016987
```

Figure 8: R^2 Score On Various Linear Machine Learning Models

This R^2 score can be further improved by using feature scaling and pipelines to imported data. Using more advanced models can further improve R^2 score, so using next advanced algorithms as *XGBoost regression, Decision Trees Algorithm, Nevian-Bayes regressor, Generalized linear regression & support vector machines* to move on to better decision boundary rather than a linear one, as obtained till now in Linear, Lasso, ridge & Elastic Net Regression.

Using advanced hyperparameter tuning to the model, in order to fit the model with best-suited parameters, informed search algorithms, like Genetic Search algorithm & coarse to fine search, is implied.

Hence, when these modifications will be induced then the model may become up to 90% accurate which is even not the case of manual testing as there is a quality difference in site condition & lab casting, also cube prepared may not homogeneous, the error induced by reading or working of testing mechanism & calculation or conceptual error in the mix design step.

3.4Application of Deep Learning Networks & Neural Networks:-

Work is also done on a deep learning algorithm for the prediction of compressive strength. The extra thing that is added here that deep learning also counters the interactions between various features, which was not there in case of linear machine learning models. Interactions are then allotted some weights based on their effect on the response variable.

Here is the basic architecture of a neural network:
It consists of one input neuron, multiple hidden neurons & finally one output neuron. Input neuron consists of a number of nodes equal to a number of feature variables, which are 8. In contrast, all hidden neurons may contain any number of nodes, which are optimized based on the performance of model & output neuron consist of a number of nodes just equal to a number of response variables, which is 1.

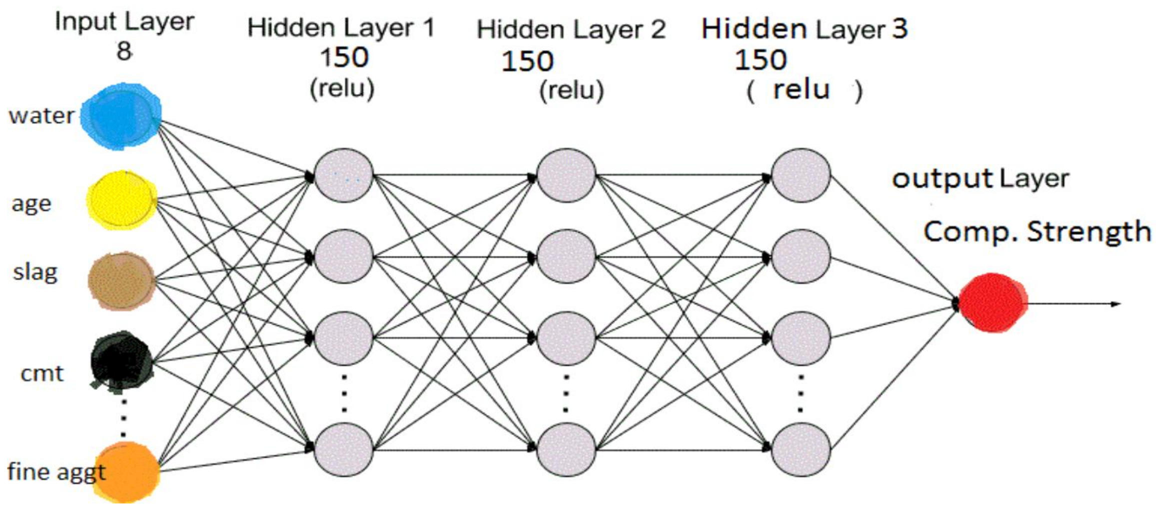


Figure 9: Architecture of artificial neural network adopted

Deep learning works on random weights allocation, the prediction from forwarding propagation passing to the right side from the input layer through multiple hidden layers & finally to output layer & make a prediction based on random weights. An error was calculated by defining loss function as 'mean squared error' and done back propagation to optimize weights using Gradient Descent Algorithm. In this algorithm owing to learning rate, the slope of loss function on the node it is calculated at, the slope of activation function on the node it is deeded into & value at that node, multiplication of these all are then subtracted from respective randomly allocated weights & finally doing this calculus trick for minimizing the value of the slope, optimized weights are obtained, which are then again forward propagated and results into final prediction. The activation function is the mathematical function applied to input coming to a particular node & after application of activation function; it is finally converted into node output, ready to multiply with weights further in forwarding propagation. Sequential is used to add a neural network in our work & dense it's layered with a suitable number of choice of nodes in hidden layers & the specified number of nodes in input & output layers.

The optimized learning rate is chosen as 'adam,' the value of which is 0.01 & tried over all possible activation functions that are available in the Keras library, which works on Tensorflow backend. Thus multiple neural networks are prepared & tested for error matric 'mean absolute error,' it was found that for ReLU (Rectified Linear Activation Function) & SeLU (Secondary Linear activation function), terms out to provide most accurate predictions. Since the model contains only 3 to 4 hidden layers here, so no dead neuron problem occurs here. SeLU activation function is well & good choice since it doesn't return zero as per it's the definition, it contains some small constant values which are multiplied with the input if it's negative, thus resulting in non-zero small negative number as out of that particular node. The number of epochs is specified to increase computational efficiency. Thus the weight optimization algorithm is named as stochastic gradient descent.

After optimizing the learning rate, assigning a particular number of epochs & choosing the best-suited activation function, it is also required to optimize the number of hidden layers & nodes per layer.

A trial on various possible combinations was done again & best combination of nodes per layer & number of hidden layers adopted. Based on a number of epochs forward & backward propagation occur multiple times & weights are kept on optimizing in each epoch owing to working stochastic gradient descent algorithm it is seen that after a certain number of epochs, loss function & error matric again starts to be increasing, that's the point where early stopping is required. It was also defined that 20% of the training dataset will be treated as a validation dataset, which is created by passing argument validation split in the model. Fit step along with the data to fit into & number of epochs.

Here are some trial models with a finalized model, at last, are pasted along with their features to understand the way to the choice of optimum & effect of various parameters like activation function, learning rate, number of layers & no of nodes per layer. The summary of each model has been presented, including the number of parameters it has. Numbers of parameters are the number of weights that are associated with that layer.

Number of weights=(A*B)+C (2)

Here B=C=Number of nodes in the current layer,
C term represent bias,i.e., weights associated with input 1 of the previous layer,
A=number of nodes in the previous layer.

This formula is derived on the basis of basic permutations & combination formulae. As each node is interconnected with each node of it's previous & upcoming layer. In each snapshot of collaboratory console attached in following points, the loss function & error written along with last epoch are that made on validation set which is part of training set itself & the loss function & error reported after model summary are the one, measured on test data, which was created earlier by using train test split method of sci-kit-learn library. All the following models are trained to 500 numbers of epochs with earlystopping of patience=20 & having 15% of the training set as the validation set in common:

• Using 'tan h' activation function, 3 hidden layers of 80 nodes each (not much accuracy).

```
Epoch 292/500
700/700 [==============================] - 0s 51us/step - loss: 70.8864 - mean_absolute_error: 6.7236 - val_loss: 89.5148 - val_mean_absolute_error: 7.0524
Epoch 293/500
700/700 [==============================] - 0s 50us/step - loss: 71.9634 - mean_absolute_error: 6.7861 - val_loss: 98.6365 - val_mean_absolute_error: 7.4195
Epoch 294/500
700/700 [==============================] - 0s 55us/step - loss: 73.3541 - mean_absolute_error: 6.9079 - val_loss: 106.3181 - val_mean_absolute_error: 7.4534
Epoch 295/500
700/700 [==============================] - 0s 56us/step - loss: 72.8900 - mean_absolute_error: 6.8367 - val_loss: 101.0338 - val_mean_absolute_error: 7.4866
Epoch 296/500
700/700 [==============================] - 0s 53us/step - loss: 70.5702 - mean_absolute_error: 6.6817 - val_loss: 113.0402 - val_mean_absolute_error: 7.7000
Epoch 297/500
700/700 [==============================] - 0s 50us/step - loss: 70.9305 - mean_absolute_error: 6.6938 - val_loss: 109.4937 - val_mean_absolute_error: 7.6452
Epoch 298/500
700/700 [==============================] - 0s 51us/step - loss: 78.5326 - mean_absolute_error: 7.0642 - val_loss: 118.6252 - val_mean_absolute_error: 7.8454
Epoch 299/500
700/700 [==============================] - 0s 52us/step - loss: 77.5859 - mean_absolute_error: 6.9911 - val_loss: 110.9704 - val_mean_absolute_error: 7.6313
Model: "sequential_26"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_103 (Dense)            (None, 80)                720
_________________________________________________________________
dense_104 (Dense)            (None, 80)                6480
_________________________________________________________________
dense_105 (Dense)            (None, 80)                6480
_________________________________________________________________
dense_106 (Dense)            (None, 1)                 81
=================================================================
Total params: 13,761
Trainable params: 13,761
Non-trainable params: 0
_________________________________________________________________
ArbANN1 loss on test data as mean squared error using tanh activation function: 104.25748858405548
ArbANN1 Accuracy on test data as mean absolute error using tanh activation function: 7.718321800231934
```

Figure 10: Results Of 'tanh' activation function with 3 layers each having 80 nodes

Using 'ReLU' activation function & 3 hidden layers of 80 nodes each. Accuracy improved; hence ReLU is preferable choice over tan h.

```
700/700 [==============================] - 0s 51us/step - loss: 13.4009 - mean_absolute_error: 2.7372 - val_loss: 27.8480 - val_mean_absolute_error: 3.8137
Epoch 322/500
700/700 [==============================] - 0s 53us/step - loss: 13.3898 - mean_absolute_error: 2.7572 - val_loss: 32.6737 - val_mean_absolute_error: 4.1380
Epoch 323/500
700/700 [==============================] - 0s 50us/step - loss: 14.1393 - mean_absolute_error: 2.7939 - val_loss: 26.9031 - val_mean_absolute_error: 3.7651
Epoch 324/500
700/700 [==============================] - 0s 60us/step - loss: 14.5637 - mean_absolute_error: 2.8743 - val_loss: 34.7830 - val_mean_absolute_error: 4.6722
Epoch 325/500
700/700 [==============================] - 0s 54us/step - loss: 14.7422 - mean_absolute_error: 2.8655 - val_loss: 26.8454 - val_mean_absolute_error: 3.8550
Epoch 326/500
700/700 [==============================] - 0s 59us/step - loss: 12.8946 - mean_absolute_error: 2.6978 - val_loss: 28.2469 - val_mean_absolute_error: 3.7661
Epoch 327/500
700/700 [==============================] - 0s 54us/step - loss: 16.9124 - mean_absolute_error: 3.1986 - val_loss: 29.1033 - val_mean_absolute_error: 3.8933
Epoch 328/500
700/700 [==============================] - 0s 51us/step - loss: 17.5280 - mean_absolute_error: 3.2052 - val_loss: 30.6715 - val_mean_absolute_error: 4.0427
Epoch 329/500
700/700 [==============================] - 0s 51us/step - loss: 31.2232 - mean_absolute_error: 4.4569 - val_loss: 33.0886 - val_mean_absolute_error: 4.4145
Model: "sequential_27"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_107 (Dense)            (None, 80)                720
_________________________________________________________________
dense_108 (Dense)            (None, 80)                6480
_________________________________________________________________
dense_109 (Dense)            (None, 80)                6480
_________________________________________________________________
dense_110 (Dense)            (None, 1)                 81
=================================================================
Total params: 13,761
Trainable params: 13,761
Non-trainable params: 0
_________________________________________________________________
ArbANN2 loss on test data as mean squared error using relu activation function: 41.17619018184329
ArbANN2 Accuracy on test data as mean absolute error using relu activation function: 4.926435470581055
```

Figure 11: Results of 'ReLU' activation function with 3 layers each having 80 Nodes

Using 'Selu' activation function with 3 hidden layers each of 80 nodes. Now error is approx. same as ReLU, hence model is comparable to Relu activation function model, now more trial are needed focused on these two.

```
Epoch 220/500
700/700 [==============================] - 0s 62us/step - loss: 32.7990 - mean_absolute_error: 4.3871 - val_loss: 33.2156 - val_mean_absolute_error: 4.4644
Epoch 221/500
700/700 [==============================] - 0s 61us/step - loss: 33.8626 - mean_absolute_error: 4.3983 - val_loss: 33.5335 - val_mean_absolute_error: 4.3343
Epoch 222/500
700/700 [==============================] - 0s 60us/step - loss: 34.2415 - mean_absolute_error: 4.5600 - val_loss: 54.4907 - val_mean_absolute_error: 5.7984
Epoch 223/500
700/700 [==============================] - 0s 63us/step - loss: 32.3868 - mean_absolute_error: 4.3186 - val_loss: 33.3725 - val_mean_absolute_error: 4.4622
Epoch 224/500
700/700 [==============================] - 0s 62us/step - loss: 29.0536 - mean_absolute_error: 4.0399 - val_loss: 36.1497 - val_mean_absolute_error: 4.5284
Epoch 225/500
700/700 [==============================] - 0s 60us/step - loss: 34.5747 - mean_absolute_error: 4.5102 - val_loss: 30.5388 - val_mean_absolute_error: 4.1931
Epoch 226/500
700/700 [==============================] - 0s 63us/step - loss: 35.7110 - mean_absolute_error: 4.5712 - val_loss: 33.1050 - val_mean_absolute_error: 4.4101
Epoch 227/500
700/700 [==============================] - 0s 63us/step - loss: 35.5007 - mean_absolute_error: 4.4886 - val_loss: 36.8789 - val_mean_absolute_error: 4.8469
Model: "sequential_28"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_111 (Dense)            (None, 80)                720
_________________________________________________________________
dense_112 (Dense)            (None, 80)                6480
_________________________________________________________________
dense_113 (Dense)            (None, 80)                6480
_________________________________________________________________
dense_114 (Dense)            (None, 1)                 81
=================================================================
Total params: 13,761
Trainable params: 13,761
Non-trainable params: 0
_________________________________________________________________
ArbANN8 loss on test data as mean squared error using selu: 40.07452922191435
ArbANN8 Accuracy on test data as mean absolute error using selu: 4.998620510101318
```

Figure 12: Results Of 'SeLU' activation function with 3 layers each having 80 nodes

Increasing number of nodes in hidden layers to 150.Error is more minimized on both training & testing dataset.

```
Epoch 399/500
700/700 [==============================] - 0s 85us/step - loss: 11.7398 - mean_absolute_error: 2.5178 - val_loss: 47.4273 - val_mean_absolute_error: 3.8747
Epoch 400/500
700/700 [==============================] - 0s 82us/step - loss: 11.7551 - mean_absolute_error: 2.5303 - val_loss: 50.0626 - val_mean_absolute_error: 3.8618
Epoch 401/500
700/700 [==============================] - 0s 80us/step - loss: 10.4822 - mean_absolute_error: 2.3323 - val_loss: 48.6706 - val_mean_absolute_error: 3.6544
Epoch 402/500
700/700 [==============================] - 0s 84us/step - loss: 8.8854 - mean_absolute_error: 2.1568 - val_loss: 53.1710 - val_mean_absolute_error: 4.1909
Epoch 403/500
700/700 [==============================] - 0s 96us/step - loss: 10.9887 - mean_absolute_error: 2.4499 - val_loss: 53.9929 - val_mean_absolute_error: 4.4651
Epoch 404/500
700/700 [==============================] - 0s 78us/step - loss: 10.5924 - mean_absolute_error: 2.3815 - val_loss: 53.3190 - val_mean_absolute_error: 3.6801
Epoch 405/500
700/700 [==============================] - 0s 79us/step - loss: 8.0679 - mean_absolute_error: 1.9993 - val_loss: 50.0633 - val_mean_absolute_error: 3.5723
Epoch 406/500
700/700 [==============================] - 0s 97us/step - loss: 9.0576 - mean_absolute_error: 2.1580 - val_loss: 52.7724 - val_mean_absolute_error: 3.9803
Model: "sequential_31"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_123 (Dense)            (None, 150)               1350
_________________________________________________________________
dense_124 (Dense)            (None, 150)               22650
_________________________________________________________________
dense_125 (Dense)            (None, 150)               22650
_________________________________________________________________
dense_126 (Dense)            (None, 1)                 151
=================================================================
Total params: 46,801
Trainable params: 46,801
Non-trainable params: 0
_________________________________________________________________
ArbANN11 loss on test data as mean squared error using selu with 150 nodes & 3 hidden layers: 32.458508241523816
ArbANN11 Accuracy on test data as mean absolute error using selu with 150 nodes & 3 hidden layers: 3.7851574420928955
```

Figure 13: Results of 'SeLU' activation function with 3 layers each having 150 nodes & 500 epochs

Increasing number of epochs further to 1000, error minimizes more on both validation & testing dataset, on which model has not been trained.

```
Epoch 839/1000
700/700 [==============================] - 0s 76us/step - loss: 2.6886 - mean_absolute_error: 1.1612 - val_loss: 30.5002 - val_mean_absolute_error: 3.6048
Epoch 840/1000
700/700 [==============================] - 0s 76us/step - loss: 2.9302 - mean_absolute_error: 1.2092 - val_loss: 32.9682 - val_mean_absolute_error: 3.6524
Epoch 841/1000
700/700 [==============================] - 0s 75us/step - loss: 3.4586 - mean_absolute_error: 1.3444 - val_loss: 27.6451 - val_mean_absolute_error: 3.2659
Epoch 842/1000
700/700 [==============================] - 0s 75us/step - loss: 3.7850 - mean_absolute_error: 1.4046 - val_loss: 33.5827 - val_mean_absolute_error: 3.7699
Epoch 843/1000
700/700 [==============================] - 0s 74us/step - loss: 3.7779 - mean_absolute_error: 1.4419 - val_loss: 31.3803 - val_mean_absolute_error: 3.7671
Epoch 844/1000
700/700 [==============================] - 0s 76us/step - loss: 3.2966 - mean_absolute_error: 1.3226 - val_loss: 32.1054 - val_mean_absolute_error: 3.6406
Epoch 845/1000
700/700 [==============================] - 0s 73us/step - loss: 3.3681 - mean_absolute_error: 1.3320 - val_loss: 33.9055 - val_mean_absolute_error: 4.0783
Epoch 846/1000
700/700 [==============================] - 0s 78us/step - loss: 6.6338 - mean_absolute_error: 2.0000 - val_loss: 24.7327 - val_mean_absolute_error: 3.3031
Model: "sequential_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_9 (Dense)              (None, 150)               1350
_________________________________________________________________
dense_10 (Dense)             (None, 150)               22650
_________________________________________________________________
dense_11 (Dense)             (None, 150)               22650
_________________________________________________________________
dense_12 (Dense)             (None, 1)                 151
=================================================================
Total params: 46,801
Trainable params: 46,801
Non-trainable params: 0
_________________________________________________________________
ArbANN11 loss on test data as mean squared error using selu with 150 nodes & 3 hidden layers: 25.204379535415796
ArbANN11 Accuracy on test data as mean absolute error using selu with 150 nodes & 3 hidden layers: 3.3341002464294434
```

Figure 14: Results of 'SeLU' activation function with 3 layers each having 150 nodes & 1000 epochs

Thus deep learning models are more efficient in prediction than conventional machine learning ones as they also involve complex interactions between all feature variables in multiconnected, multi nodes multilayers. Still, as it is about 3 MPa, a considerable amount of mean absolute error is there on validation dataset, the reason is again lack of feature scaling & pre-processing to make dataset free from outliers.

It is an on-going trial from author's side, if this area is worked on extensively by industry professionals & using larger dataset & eliminating as well as additional sources of error; this remaining error will surely be minimized further from 2 or 3 MPa to just less than 1 MPa, which would be way better than conventional, laborious, time taking, experience handled, material wasting multiple steps conventional design & cube test process.

In the post COVID world, one may not get all kinds of materials readily available at the site; there may be lots of reasons for delayed manufacture & transportation of materials. So this model could be utilized with no time wastage, at the site itself, so that best composition can be achieved with readily available resources, strength requirements & meeting economy. For making it more advanced, software-based on PERT & CPM with model can be added to directly get an optimized & economical composition. The second benefit of this model is that while transporting concrete mix from mix plant to site if time is long and setting time of a particular cement is less so one has to either use admixtures or more water to increase setting time so that it gets sufficient workability at site, another way of doing the same is to change the composition of cement by reducing the percentage of C3A.So if just add some more features such as C3A percent, amount of Gypsum & amount of C3S & C2S & admixture then this advanced model can also predict setting time & workability values, which no need to tell, will be how much time, efforts, apparatus & machine requirement & labor saving step,& will be vitally needed in Post COVID world. Last but the most important role of this model is that as the water-cement ratio is an important parameter as strength & workability both depend on this (Abraham law) , it suggests that if water content is increased to increase the workability then strength is reduced. Hence, it is required to maintain a proper balance between workability and strength of concrete by adjusting amount of water and cement which is found based on trial mixing, and repeating that mixing iterative number of times, which results into wastage of lots of calculations, labour, materials & hence cost.

On the other hand, with this model, one can play with varying any combinations & getting strength results within a minute. This becomes manifolds beneficial, mainly when some special kind of concrete, as explained in the following sections of this paper, is required. Such as high workability is needed for pumpable concrete (to reduce no. of labours) without affecting strength of concrete, because automating the process is required by just transporting concrete with the help of power machines so that less number of labour is needed and there is a minimum possibility between them to get contacted with one another, i.e., Social Distancing. Pumpable concrete with high workability & optimized composition (so that there is no compromise with strength) & properly designed automation at the site along with training provided to labours to operate the same, indirectly gives rise to maintain social distancing and less labour requirement at the site which is the need of the hour in post COVID world.

2. Applied Sensor Chip Atcuring Stage

The safety of a person who's in charge of curing must be ensured. This becomes more necessary, in the view of the most recent research published on 2nd June, by Prof. Rajneesh Bhardwaj & Prof. Amit Agrawal of IIT Bombay, who took a sample of respiratory droplets exerted from corona patient & studied survival of the virus in different ambient temperature environment. As a result, it was concluded that ambient temperature & relative humidity play critical roles in the time of survival. So in view of that, the curing period is no more exception. That's why in the conclusion stage, essential efforts to minimize risks are recommended.

Other ways of curing if available should be emphasized rather than water curing, so that there will be less of humidity sustain. Electric curing, with installed rubber sheets, is one of the ways but a bit costly; other methods like steam curing, chemical, or infrared radiation based curing can also work well. Moreover, if water curing is inevitable then an electronic chip on the prepared concrete structure can be installed for this, which work on wireless sensors & through the internet of things & cloud communication it should send signals to a receiver, which convert them to information signal & by proper modelling in receiver system it will be able to convey to the user when structure require water, then the only worker should go there. Also, it will be the time when there is minimum humidity present there, hence reduced chances of labour to get infected.

This chip will work on the principles more or less the same as the structural health monitoring system. Signals received from sensors through cloud will be converted to an output that can be interpreted, through adding physics informed modelling to the system. Adding this will enhance expected output rather than data trained modelling alone.

In addition to that, also installation of Jute covers or making soil leves on roof or pavement type flat structures is also a way to effectively utilize given water at one time and hence increases interval of curing between two successive ones.

Thus adopting any of the above stated multiple options as per the availability & economic consideration for the individual site may significantly reduce the spread of the virus if there any at post-construction curing stage in the post-COVID world.

3. Other Recommended Provisions

Making use of properly designed and planned site layout plan in order to ensure social distancing, using infrared viralyzer to sanitize electronic devices & files, a pre scheduled spray of disinfectant sodium hypochlorite solution (1% strength), a well nutritioned diet as per immunity boosting guidelines of Ayush Ministry, provided viro shield to labours, more use of fly ash in cement, using hydraulic disinfectant cement, using self compacted concrete in order to ease in automation are some of the ways to ensure prevent of spread of coronavirus at construction site.

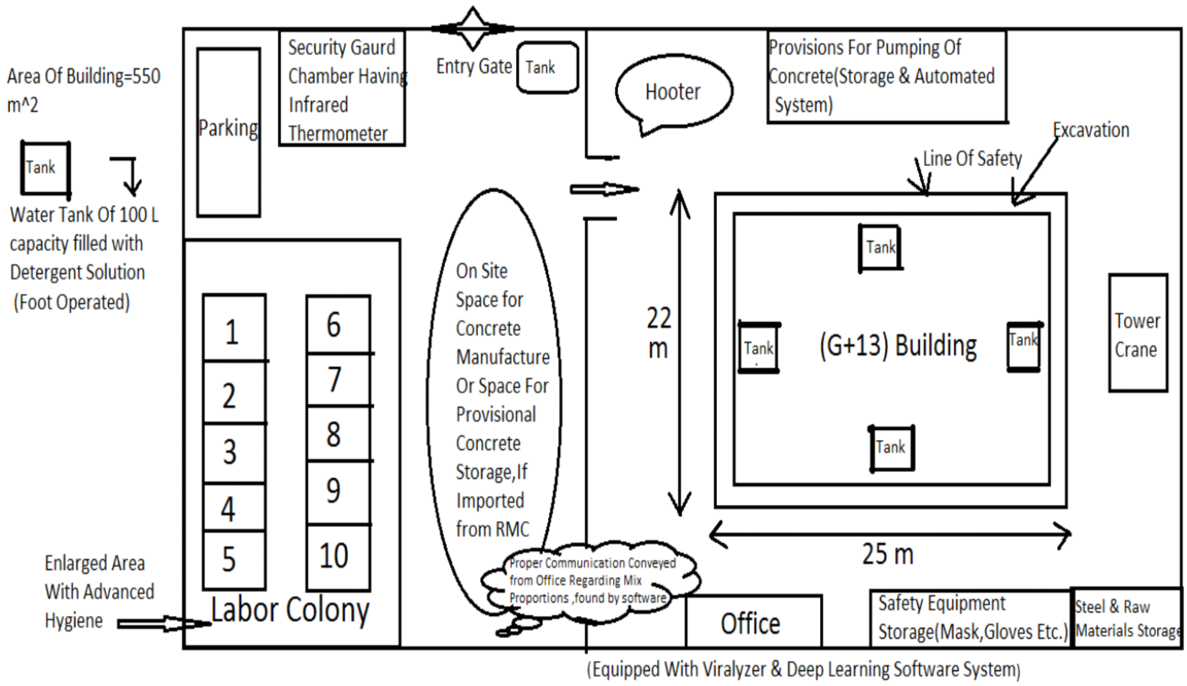


Figure 15: Site layout

4. Conclusion

This paper can be concluded as:-

- At the site, proper provision should be made for the health safety of labours and proper hygiene needs to be maintained.
- More usage of automation should be there in construction process to reduce the labour requirement and this paper recommends to more of modelling techniques in determining concrete mix to reduce wastage of labour and material.
- At the curing stage, an electronic chip is to be used, which indicates when it is needed to do the curing. Thus it provides safety to the person against any virus, who is assigned a job of curing. Also, various alternatives to water curing are emphasized.

The paper can be summarised as to make more use of modelling & prediction techniques & available and as per economical sensor techniques to employ at any stage of any construction project post-COVID wherever needed or appropriate. It is required to give more importance to health by providing hygienic environment and maintaining social distancing by reducing the number of labours and increasing usage of machines, self-compacted concrete, etc. to make construction process economical, durable, efficient and healthier.

References

Strength Prediction through ANN & Machine learning

[1] Andreas C. Muller & Sarah Guido,Introduction to machine learning with python,O'Reilly Publication

[2] Comparison of Machine LearningTechniques for the Prediction of Compressive Strength of Concrete

[3] D. Prayogol, M.-Y. Cheng, J. Widjaja, H. Ongkowijoyo, and H. Prayogol ,Prediction of Concrete Compressive Strength from Early Age Test Result Using an Advanced Metaheuristic-Based Machine Learning Technique ,

[4] Department of Civil Engineering, Petra Christian University, Iendonesia Department of Civil and Construction Engineering, National Taiwan University of Science and Technology, Taiwan

[5] Lee, S. Prediction of concrete strength using artificial neural networks. Eng. Struct. 2003

[6] Palika Chopra ,Rajendra Kumar Sharma,Maneek Kumar ,and Tanuj Chopra ,Department of Computer Science and Engineering, Thapar University, Patiala, India

[7] UCI repository dataset by Prof. I-Cheng Yeh, Department of Information Management, Chung-Hua University, Taiwan

[8] Yeh, I.-C. (1998): Modelling of strength of high performance concrete using artificial neural networks. In: Cement and Concrete Research

Electronic Chip for Curing

[9] "Likelihood of survival of coronavirus in a respiratory droplet deposited on a solid surface," is authored by Rajneesh Bhardwaj and Amit Agrawal. The article will appear in Physics of Fluids on June 9, 2020

[10] Machine learning for structural health monitoring: challenges and opportunities,Fuh-Gwo Yuan, Sakib Ashraf Zargar, Qiuyi Chen, Shaohan Wang, dated On 23 April 2020

[11] Von Rueden, L., Mayer, S., Garcke, J., Bauckhage, C., & Schuecker, J., "Informed machine learning–towards a taxonomy of explicit integration of knowledge into machine learning."

[12] Worden, K., & Manson, G., "The application of machine learning to structural health monitoring" Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences

Other Recommended Provisions

[13] Dr. Sameer Srivastava, Dr. Ambak K. Rai, Dr. Ashutosh Mani, Dr. N.K. Singh and Prof. Shivesh Sharma from Department of Biotechnology, MNNIT Allahabad , Design of Viroshield & Viralyzer 1.0
[14] Immunity boosting guidelines, by ministry of Ayush ,Uttar Pradesh Government.
[15] Notice from chief secretary Amit Mohan Prasad Uttar Pradesh Governence to DMs & CMOs,Medical section 5,Lucknow dated on 23 april 2020,Subject:-Guidelines on disinfection of public places & offices in regard of COVID 19 decease.
[16] U.S. guidelines for disease prevention and control.