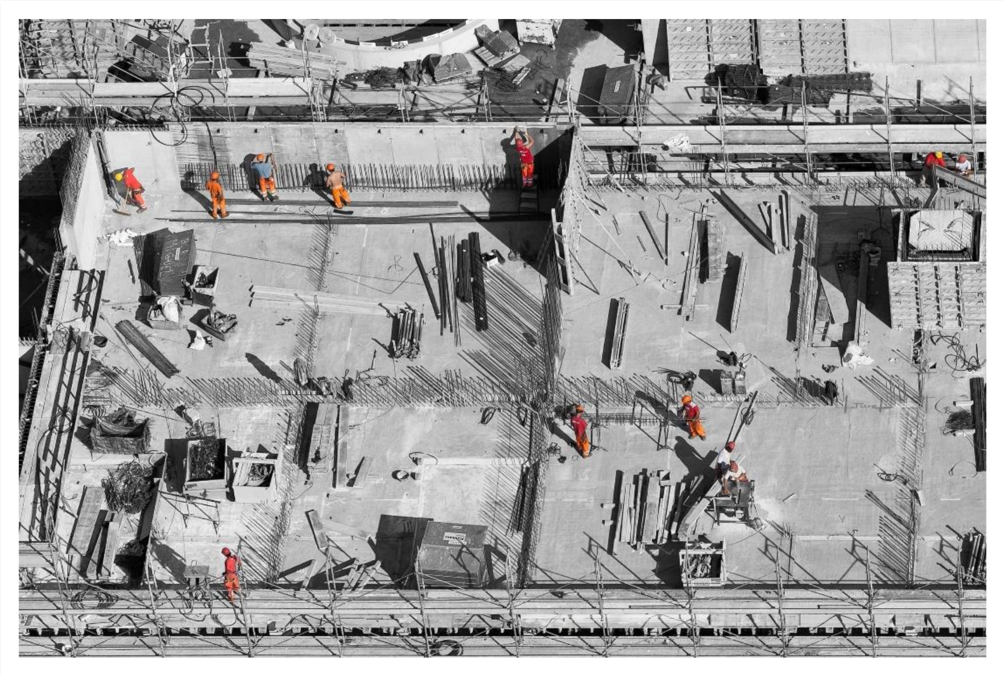
PREDICTING COMPRESSIVE STRENGTH OF CONCRETE

1. Introduction

1.1 overview:



Every data scientist has to build domain knowledge in every field because we have to tackle each problem that is probably faced by the world. If you don’t know what domain knowledge is? it is the knowledge of a specific field or having specialization in any field. we can also say that it is a piece of general knowledge, so as a data scientist we probably solve the real-world problems which were based on machine learning with the domain knowledge of the specific problem.

As a data scientist, we generally faced many real-life problems, like-some social issue, construction, etc.. we have to solve this type of problem using machine learning techniques, let’s we take an example of construction, what will you refer the word construction? it is the art and science to form objects, systems, or organizations. what will you imagine from the word construction is that mega buildings, machines, material, etc… but you know that what is used to build these mega buildings, for construction we use material, cement, iron rods, etc.. where the material is a most important part of building making.

1.2 Purpose

” Use the best possible materials, and reveal the quality of those materials and the craftsmanship of their assembly “.

The meaning of these beautiful words is that if we use the best quality material then you probably be the quality constructor. We can relate this quote with our example, we discuss above that certain things are used for making mega buildings but the material is most important.

Our aim from the project is to make use of pandas, matplotlib, numpy, & seaborn libraries from python to extract the libraries for machine learning for Predicting the compressive strength of concrete.
So, we are going to analyze the Concrete Compressive Strength dataset and build a Machine Learning model to predict the quality.

2. LITERATURE SURVEY

2.1 Existing problem

How we know that this concrete is quality proof or not, that we generally check the strength of concrete. In easy words the Compressive Strength of Concrete determines the quality of Concrete, we check it by standard crushing test on a concrete cylinder. Concrete strength is also considered a key factor in obtaining the desired durability. For testing strength it will take 28 days this is a large time.

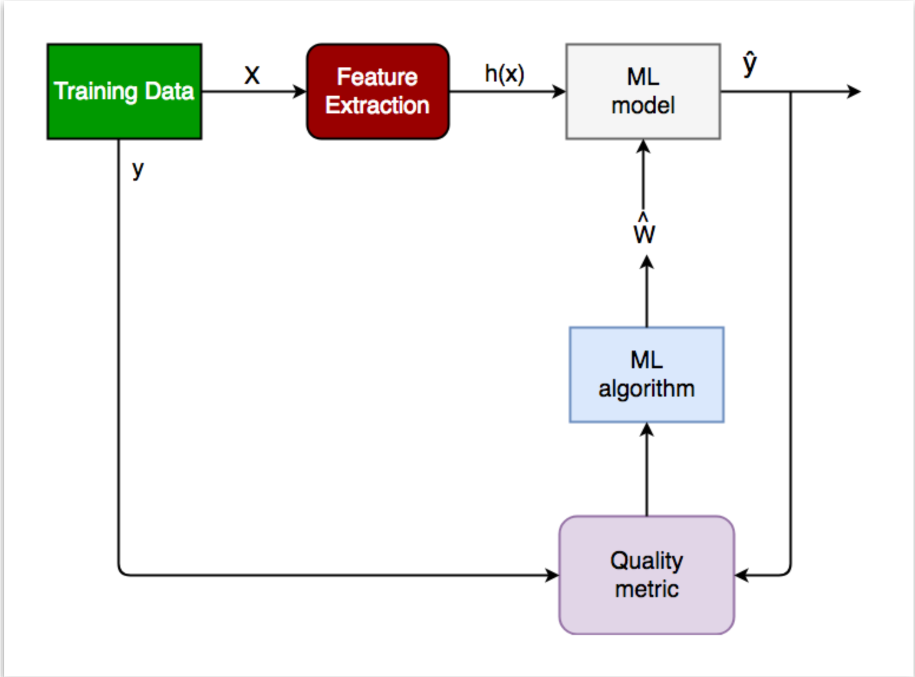
2.2 Proposed solution

So what we will do now? By use of Data Science, we reduce this lot's of effort we will predict that in how much quantity we have to use which raw material for good compressive strength.

So, we are going to analyse the Concrete Compressive Strength dataset and build a Machine Learning model to predict the quality.

3. THEORITICAL ANALYSIS

3.1 Block diagram



3.2 Hardware / Software designing

3.2.1 Hardware

There are a few high end (and expectedly heavy) laptops like Nvidia GTX 1080 (8 GB VRAM), which can train an average of ~14k examples/second. In addition, you can build your own PC with a reasonable **CPU** and a powerful GPU, but keep in mind that the CPU must not bottleneck the GPU.

3.2.2 Software

Software requirements

- CUDA Deep Neural Network (cuDNN) 7.5 library.
- NVIDIA CUDA 10.1.
- NVIDIA GPU driver 418.40.
- NVIDIA NCCL2 2.4.
- Anaconda 2018.12.

4. EXPERIMENTAL INVESTIGATIONS

Dataset:

We will use a concrete compressive strength dataset which was retrieved from the Kaggle.

Dataset knowledge: you see that several features affect the quality of concrete. So we discuss brief of each feature:

cement: a substance used for construction that hardens to other materials to bind them together.

slag: Mixture of metal oxides and silicon dioxide.

Fly ash: coal combustion product that is composed of the particulates that are driven out of coal-fired boilers together with the flue gases.

Water: It is used to form a thick paste.

Superplasticizer: used in making high-strength concrete.

Coaresaggregate: prices of rocks obtain from ground deposits.

fine aggregate: the size of aggregate small than 4.75mm.

age: Rate of gain of strength is faster to start with and the rate gets reduced with age.

csMPa: Measurement unit of concrete strength.

Importing Modules

```
import pandas as pd
import numpy as np
import matplotlib.pylot as plt
import seaborn as sb
```

So, we import **pandas** for data analysis, **NumPy** for calculating N-dimensional array, **seaborn**, and **matplotlib** to visualize the data.

Reading data

Generally, we use a dataset in the form of a CSV file, for reading this CSV file we will use the **panda's** library

Study Dataset

After reading the dataset we have to extract information from the data

Handling Null values

we handle the null values that are present in the dataset for better accuracy

Data Visualization

it is an approach of analyzing datasets to summarize their main characteristics.

Pair plot:

It plots a pairwise relationship in the dataset, it will create a grid of axis where the y-axis belongs to row and the x-axis belongs to columns

Scatter Plot

This plot displays the relationship between any two sets of data

Correlation plot

The correlation plot shows the correlation coefficient between variables. This plot contains the correlation matrix-like table.

Box plot

we plot the outlier that is present inside the dataset.

Splitting the data

we use the scikit-learn module train_test_split, which is used for splitting the training and testing parts

Dividing Dependent And Independent Variables

1. **Independent** variables contain a list of those variables in which concrete quality is dependent.
2. The **dependent** variable is that variable that is dependent on other variables' values.

1. Independent variables are **cement, flyash, water, superplasticizer, coaseseaggregate, fineaggregate, age**.
2. dependent variable is the only **csMPa**

Feature Scaling

We do scaling of data for balancing the data points.

Applying model

Machine learning consists of algorithms that can automate analytical **model building**. Using algorithms that iteratively learn from data. In this step, we applying several machine learning algorithms to training data.
In this project we will use Linear Regression, Lasso Regression, Ridge Regression, RandomForestRegressor for model building.
The accuracy score of RandomForestRegressor is highest among linear, lasso, and ridge regression, so we use the RandomForestRegressor model, Here the highest accuracy means it predicts the quality of concert by using training, which contains independent variables, and also it gives less error rate.
We use RandomForestRegressor for Predicting compressive strength of concrete.

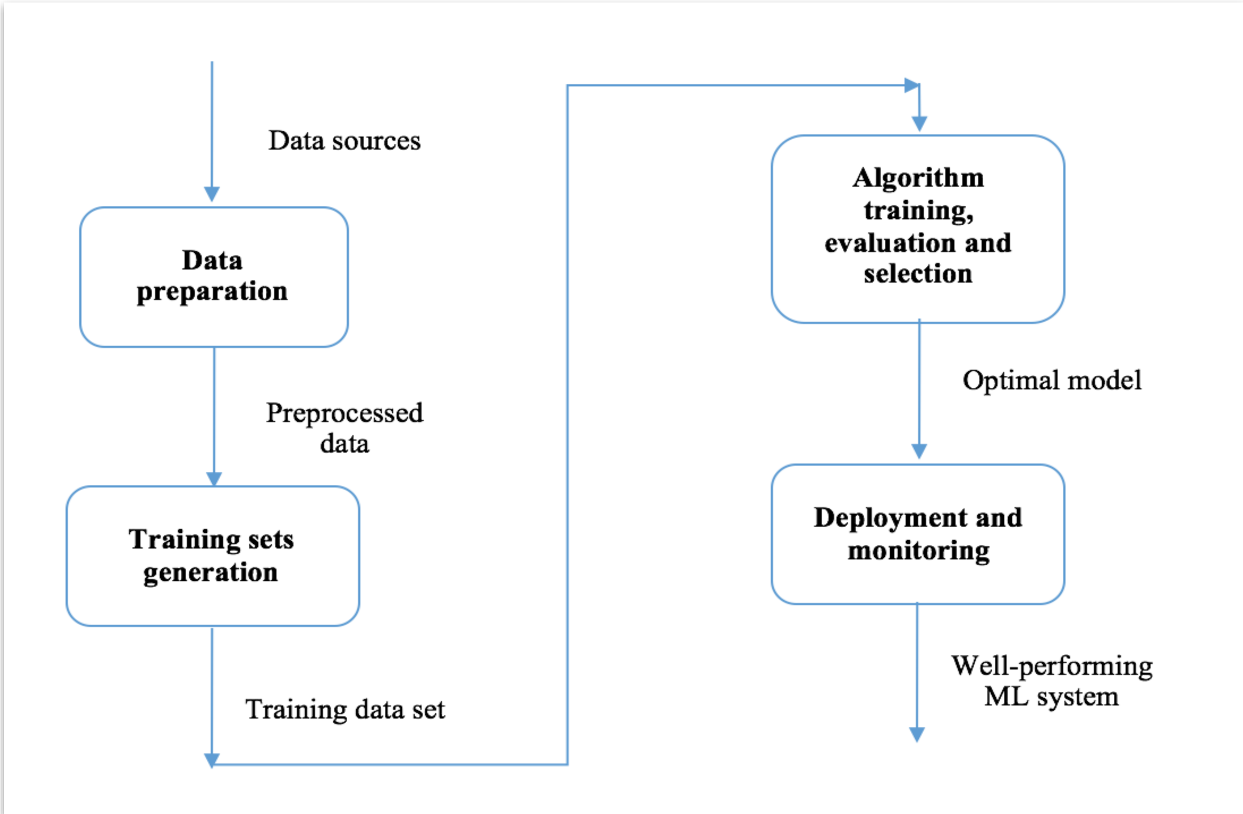
Saving the Model

we save our machine learning model using pickle

Deployment using flask

In the end, we want our model to be available for the end-users so that they can make use of it. Model Deployment is one of the last stages of any machine learning project and can be a little tricky. How do you get your machine learning model to your client/stakeholder? What are the different things you need to take care of when putting your model into production? And how can you even begin to deploy a model?
Here comes the role of Flask.

5. FLOWCHART


Data sources
Data preparation
Preprocessed data
Training sets generation
Training data set
Algorithm training, evaluation and selection
Optimal model
Deployment and monitoring
Well-performing ML system

6. RESULT

In [28]:
```
#Reading dataset
df = pd.read_csv("concrete_data.csv")

df
```

Out[28]:

	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
...	...	...	...	...	...	...	...	...	...
1025	276.4	116.0	90.3	179.6	8.9	870.1	768.3	28	44.28
1026	322.2	0.0	115.6	196.0	10.4	817.9	813.4	28	31.18
1027	148.5	139.4	108.6	192.7	6.1	892.4	780.0	28	23.70
1028	159.1	186.7	0.0	175.6	11.3	989.6	788.9	28	32.77
1029	260.9	100.5	78.3	200.6	8.6	864.5	761.5	28	32.40

1030 rows × 9 columns

Handling Null Values

we handle the null values that are present in the dataset for better accuracy

```
df.isnull().any() #checking for missing values
```

```
cement                           False
blast_furnace_slag               False
fly_ash                          False
water                            False
superplasticizer                 False
coarse_aggregate                 False
fine_aggregate                   False
age                              False
concrete_compressive_strength    False
dtype: bool
```

```
df.isnull().sum() #caluculate the null values in a dataset
```

```
cement                           0
blast_furnace_slag               0
fly_ash                          0
water                            0
superplasticizer                 0
coarse_aggregate                 0
fine_aggregate                   0
age                              0
concrete_compressive_strength    0
dtype: int64
```

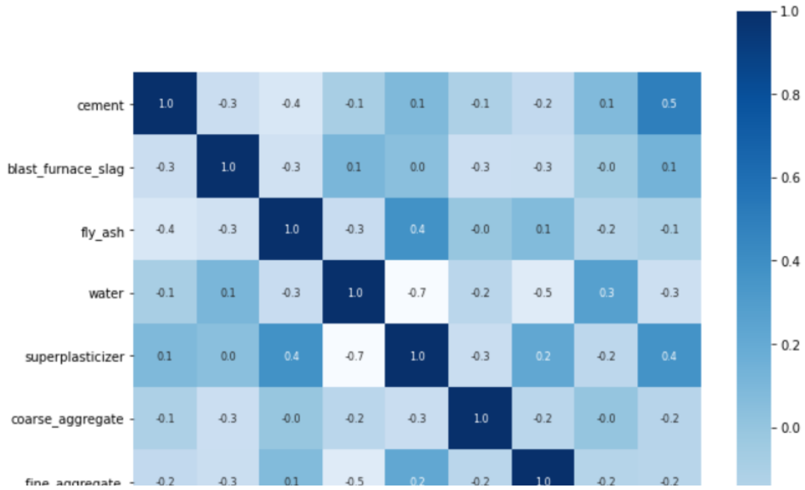
Correlation plot

The correlation plot shows the correlation coefficient between variables. This plot contains the correlation matrix-like table.

```
correlation = df.corr()
```

```
plt.figure(figsize=(10,10))
sb.heatmap(correlation,cbar=True,square=True, fmt='.1f', annot=True, annot_kws={'size':8}, cmap='Blues')
```

<AxesSubplot:>



Splitting the data into training data and test data

Now we use the scikit-learn module train_test_split, which is used for splitting the training and testing parts.

```
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest = train_test_split(x,y, test_size=0.3, random_state=42)
```

```
print(x.shape, xtrain.shape, xtest.shape)
```

```
(1030, 8) (721, 8) (309, 8)
```

Feature Scaling

We do scaling of data for balancing the data points.

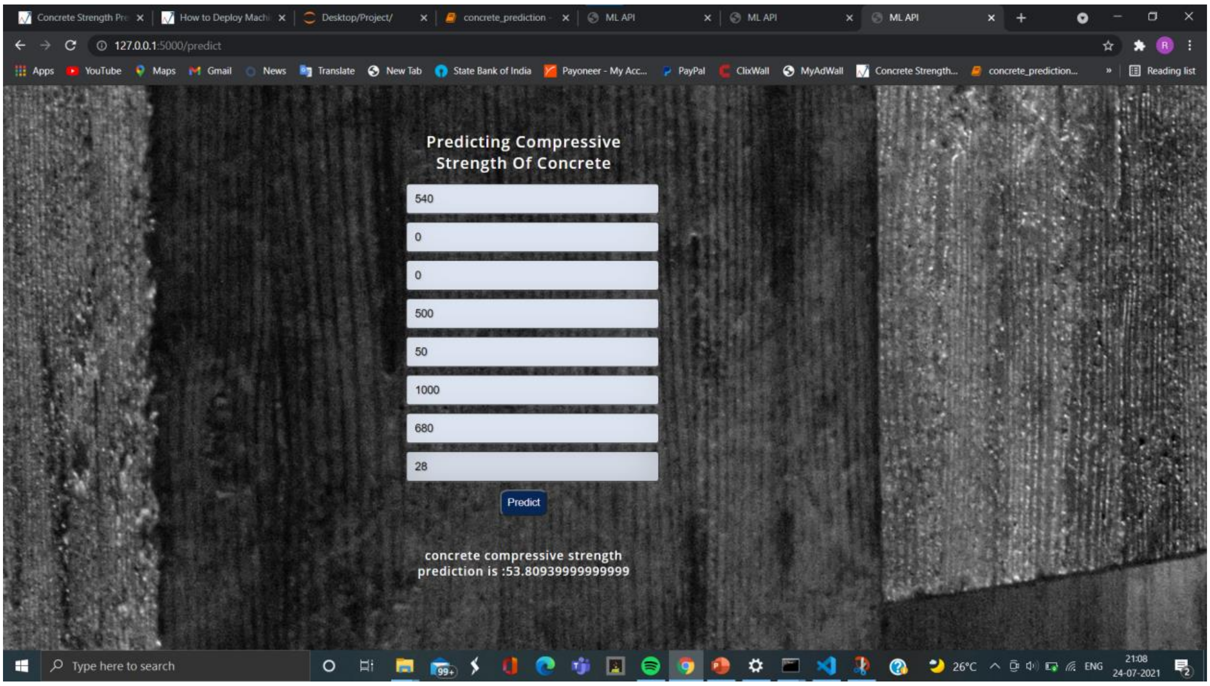
```
from sklearn.preprocessing import StandardScaler
stand = StandardScaler()
Fit = stand.fit(xtrain)
xtrain_scl = Fit.transform(xtrain)
xtest_scl = Fit.transform(xtest)
```

RandomForestRegressor

]:
```
# import random forest regression
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error
rnd = RandomForestRegressor()
fit_rnd= rnd.fit(xtrain_scl,ytrain)

print('score is:-',rnd.score(xtest_scl,ytest))
print('........................................')
print('mean_sqrd_error is==',mean_squared_error(ytest,rnd.predict(xtest_scl)))
print('root_mean_squared error of is==',np.sqrt(mean_squared_error(ytest,rnd.predict(xtest_scl))))
```

```
score is:- 0.8883530426479788
........................................
mean_sqrd_error is== 30.20879357203278
root_mean_squared error of is== 5.496252684514494
```



7. ADVANTAGES

we reduce this lot's of effort we will predict that in how much quantity we have to use which raw material for good compressive strength.

The Compressive Strength of Concrete determines the quality of Concrete.

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.

8. APPLICATIONS

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.

PREDICTING COMPRESSIVE STRENGTH OF CONCRETE

The compressive strength of concrete is mostly used criterion in producing concrete. However, testing for compressive strength of concrete specimens is a complicated and time-consuming task. More importantly, it is too late to make improvement if the test result does not satisfy the required strength, since the test is usually performed at the 28th day after the placement of concrete at the construction site.

Therefore, strength prediction before the placement of concrete is highly desirable. This study presents the effort in applying the neural network technique for predicting the compressive strength of concrete based on concrete mix proportions. For training and testing, the data sets on the mix proportions of two ready mixed concrete companies were used, and then the required compressive strength was predicted by trial and error.

The predicted compressive strengths were verified by comparing the predicted results with those tested in the laboratory. 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.

9. CONCLUSION

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, Decision Trees and Random Forests to make predictions and compared their performance. Random Forest Regressor has the lowest RMSE 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.

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.

10. FUTURE SCOPE

Several studies independently have shown that concrete strength development is determined not only by the water-to-cement ratio, but that it also is influenced by the content of other concrete ingredients. High-performance concrete is a highly complex material, which makes modelling its behaviour a very difficult task. This paper is aimed at demonstrating the possibilities of adapting artificial neural networks (ANN) to predict the compressive strength of high-performance concrete. A set of trial batches of HPC was produced in the laboratory and demonstrated satisfactory experimental results. This study led to the following conclusions: 1) A strength model based on ANN is more accurate than a model based on regression analysis; and 2) It is convenient and easy to use ANN models for numerical experiments to review the effects of the proportions of each variable on the concrete mix.

11. BIBILOGRAPHY

APPENDIX

1. I-Cheng Yeh, " Modeling of strength of high performance concrete using artificial neural networks," Cement and Concrete Research, Vol. 28, №12, pp. 1797–1808 (1998).

2. Ahsanul Kabir, Md Monjurul Hasan, Khasro Miah, " Strength Prediction Model for Concrete", ACEE Int. J. on Civil and Environmental Engineering, Vol. 2, №1, Aug 2013.

3. https://archive.ics.uci.edu/ml/datasets/Concrete+Compressive+Strength