

# Concrete Compressive

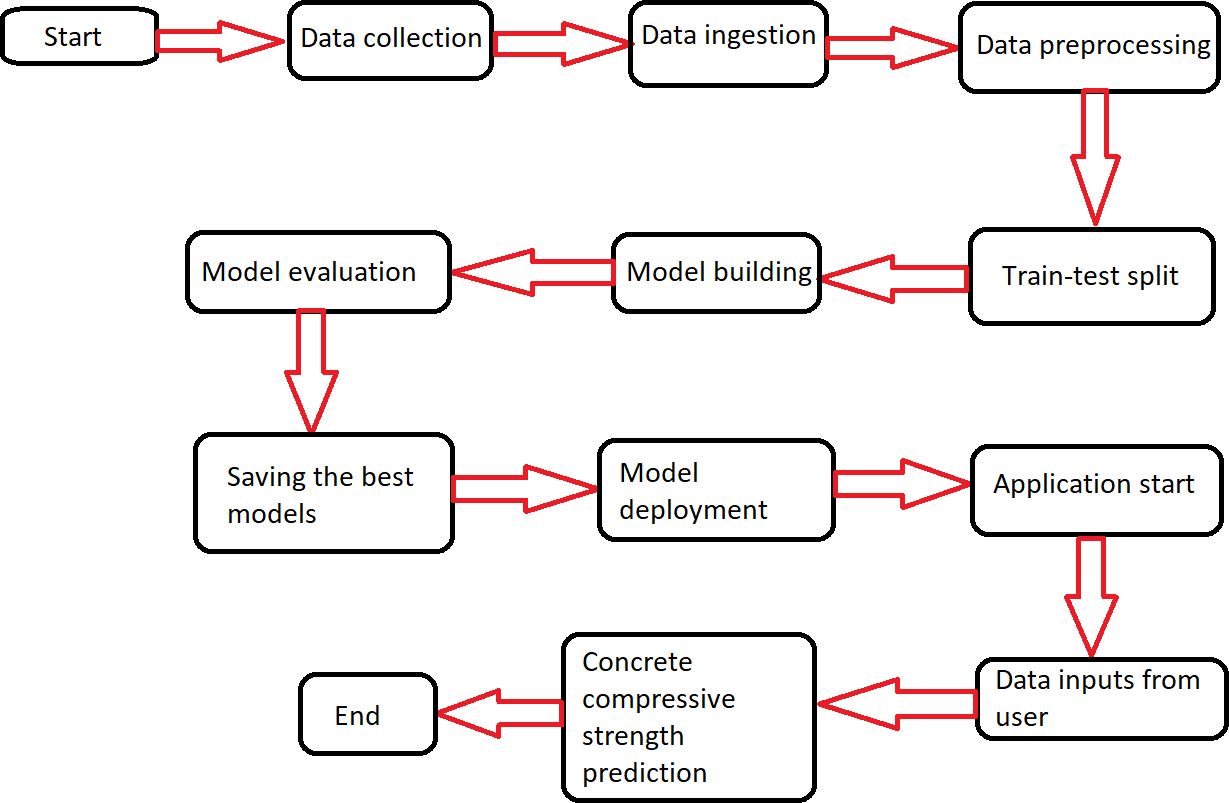
Strength Prediction

BY SREE HARSHA K

## Objective

Being one of the most frequently used building materials, the quality of concrete is determined by its compressive strength, which is measured by crushing a concrete cube or a cylinder until it starts cracking and crushed. The pressure at which the concrete cube or a cylinder starts cracking and eventually crushes is called the Concrete compressive strength and is measured in megapascals (MPa). It takes a long period of 28 days to test like this. With the help of Data science and the Machine learning technology, I developed an application, which allows an engineer to determine the strength of a concrete in just a few seconds of time.

Architecture



Data set

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Measurement | Description |
| **Cement** | Quantitative | kg in a m3 mixture | Input variable |
| **Blast Furnace Slag** | Quantitative | kg in a m3 mixture | Input variable |
| **Fly Ash** | Quantitative | kg in a m3 mixture | Input variable |
| **Water** | Quantitative | kg in a m3 mixture | Input variable |
| **Superplasticizer** | Quantitative | kg in a m3 mixture | Input variable |
| **Coarse Aggregate** | Quantitative | kg in a m3 mixture | Input variable |
| **Fine Aggregate** | Quantitative | kg in a m3 mixture | Input variable |
| **Age** | Quantitative | Days (1~365) | Input variable |
| **Concrete Compressive Strength** | Quantitative | megapascals (MPa) | Output variable |

For training and testing the model, we used the public data set available in Kaggle, “Concrete Compressive Strength Data Set” by Ahiale Darlington.

URL -

[https://www.kaggle.com/elikplim/ concrete-compressive-strength- data-set](https://www.kaggle.com/elikplim/concrete-compressive-strength-data-set)

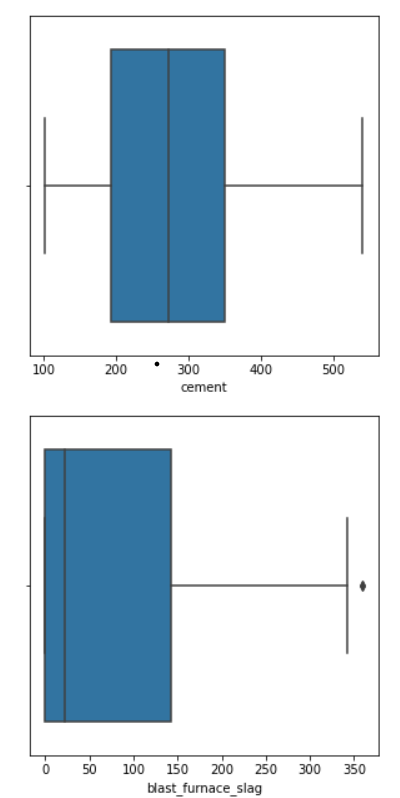
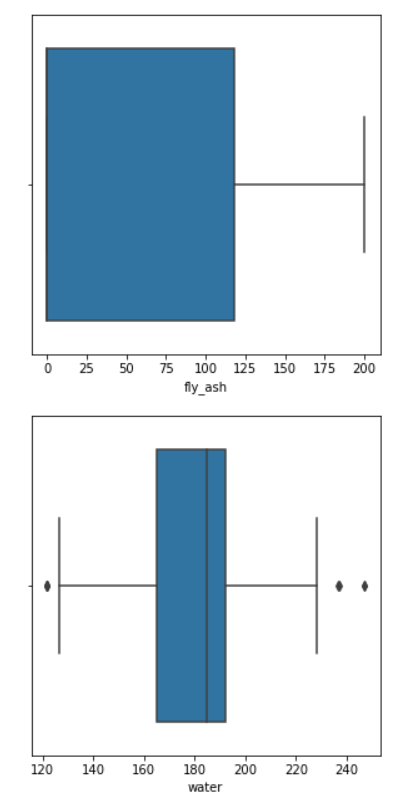
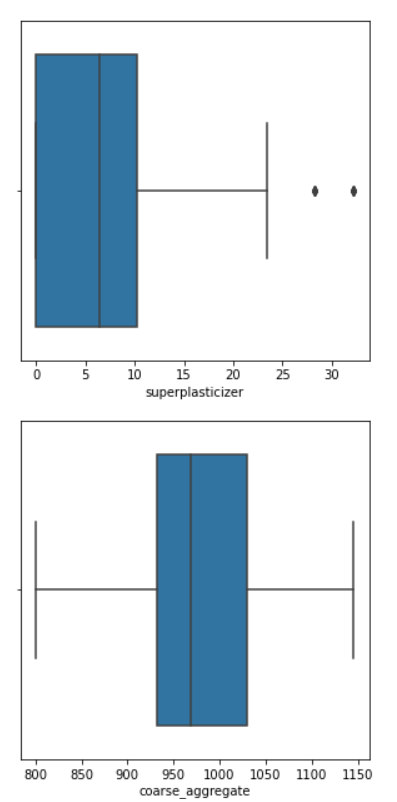
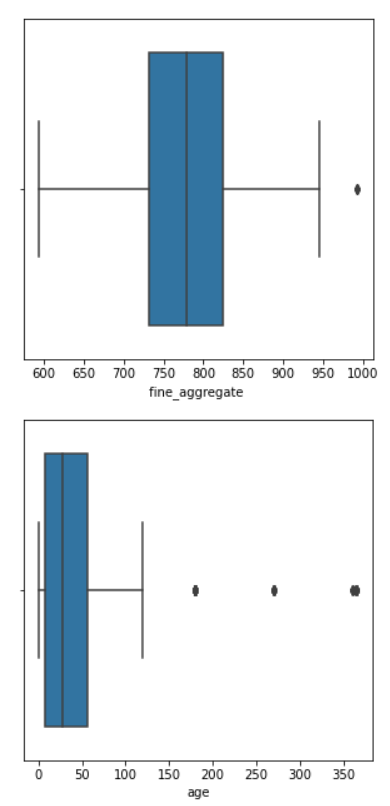
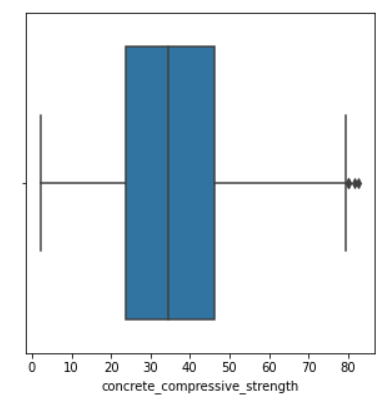
### EDA – Data cleaning

* The given data set has 9 features and each one is quantitative in nature.
* This data set doesn’t have any

missing values.

* The shape of the data set is 1030 rows × 9 column

EDA – Outlier treatment for columns



## Age” column - code

Data pre-processing

* Split the data-frame into the training (70% of the data) and testing (30% of the data) data- frames respectively.
* For building linear regression models, used the scaled data obtained by scaling the features using the Standard scaler. Tree models use the original data i.e., without scaling as they aren’t affected by the feature scaling.
* Then both the training and testing data-frames were further split into x\_train, x\_test, y\_train and y\_test. Here the data-frames with X indicate independent features while those with y indicate the dependent or the target feature.
* Model Building

## Linear Regression-

## r 2\_score in train model- 55%

## r 2\_score in test model-63%

## Random forest Regressor-

## r 2\_score in train model- 87%

## r 2\_score in test model-98%

## Ada Boost Regressor

## r 2\_score in train model- 78%

## r 2\_score in test model-82%

## Xgboost

## r 2\_score in train model- 99.9%

## r 2\_score in test model-89.99%

## Grid Search CV is used to find the best parameters of the Xgboost

## r 2\_score in train model- 99%

## r 2\_score in test model-92%

### Model deployment

* Saved XGBoost regressor modelinto the “models” directory. Then deployed the XGBoost regressor model using the Flask and linked to web application which was designed using HTML .
* Deployed on web using GitHub, Gunicorn and Heroku.
* This application can be accessed [using the URL:-](https://ccs-predictor.herokuapp.com/) https://compressivestrength.herokuapp.com/

Q & A

1. What's the source of data?

[Answer: The data for training the model is taken from Kaggle - https://www.kaggle.com/elikplim/concrete-compressive-strength-data- set.](https://www.kaggle.com/elikplim/concrete-compressive-strength-data-set)

1. What's the complete flow you followed in this project? Answer: Please refer slide 3 for the better understanding.
2. What techniques were you using for data-preprocessing?

Answer: 1. Visualizing the relation ships between the target variable and each predictor variable.

* 1. Visualizing the correlation among all the variables.
  2. Removing outliers using IQR method from the "age" column and increasing the size of the data (slides 7-8)
  3. Scaling the data for building Linear regression models using Standard scaler.

Q & A

1. What's the purpose of increasing the size of data?

Answer: Having less amount of data makes the model to learn completely on it (in addition to learning the underlying patterns, it also learns the noise in that data), leading to overfit. Means, model may not make predictions accurately on the unseen data. So, to avoid model overfit, one of the methods is to add more data points so that the model we are going to train, can learn the underlying patterns in the data efficiently and can make accurate predictions on the unseen datapoints.

1. Why feature scaling is not necessary for the tree-based models ?

Answer: The tree-based models are not sensitive to the scale of the features. If we consider a decision tree algorithm, it splits a node based on a single feature and this is not influenced by the other features, i.e., there won't be any effect of the remaining features if a split is performed based on one single feature.

1. How was prediction done?

Answer: Based on the performances of each model, I chose XGBoost regressor model. I considered only the top 6 features as per the feature importance by the model and rebuilt it.(slide 12). So, when a user inputs the age of the concrete specimen, quantities of cement, water, fly ash, superplasticizer and the blast furnace slag, the model takes these values as an input and makes a prediction of the compressive strength of the given concrete specimen.

Q & A

1. What are the different stages of deployment?

Answer: First, deployed the model locally using Flask (a micro web framework) which works as a backend application. The frontend application is a web page designed using HTML5 with CSS styling. So, when a user enters the data and hit "predict" button, model in the backend flask application makes prediction and it will be displayed in the frontend application which user can take a note. Then, deployed this application on web using Heroku and Gunicorn (a python web server gateway interface HTTP server).

1. How are logs managed?

Answer: The entire project is divided into two stages - the development stage and the deployment stage. The logs recorded during the development stage are stored in the "development\_logs.log" file while the logs recorded during the local deployment are stored in the "deployment\_logs.log" file.

# Thank you