Concrete compressive strength prediction

Abstract

Motivation:

The compressive strength of the concrete is attributed to the proportion of different ingredients. Thus, the strength of concrete might vary sometime. The compressive strength is the primary criterion for selecting concrete for a particular application. However, the characteristic strength of concrete is defined as the compressive strength of a sample that has been **cured for 28 days**. Therefore, we need to **predict the strength of concrete based on the early strength data** (Chopra et al., 2014).

Requirement of compressive strength of various application is different:

E.g., slabs, girders etc. ⟶ **require a higher compressive strength**

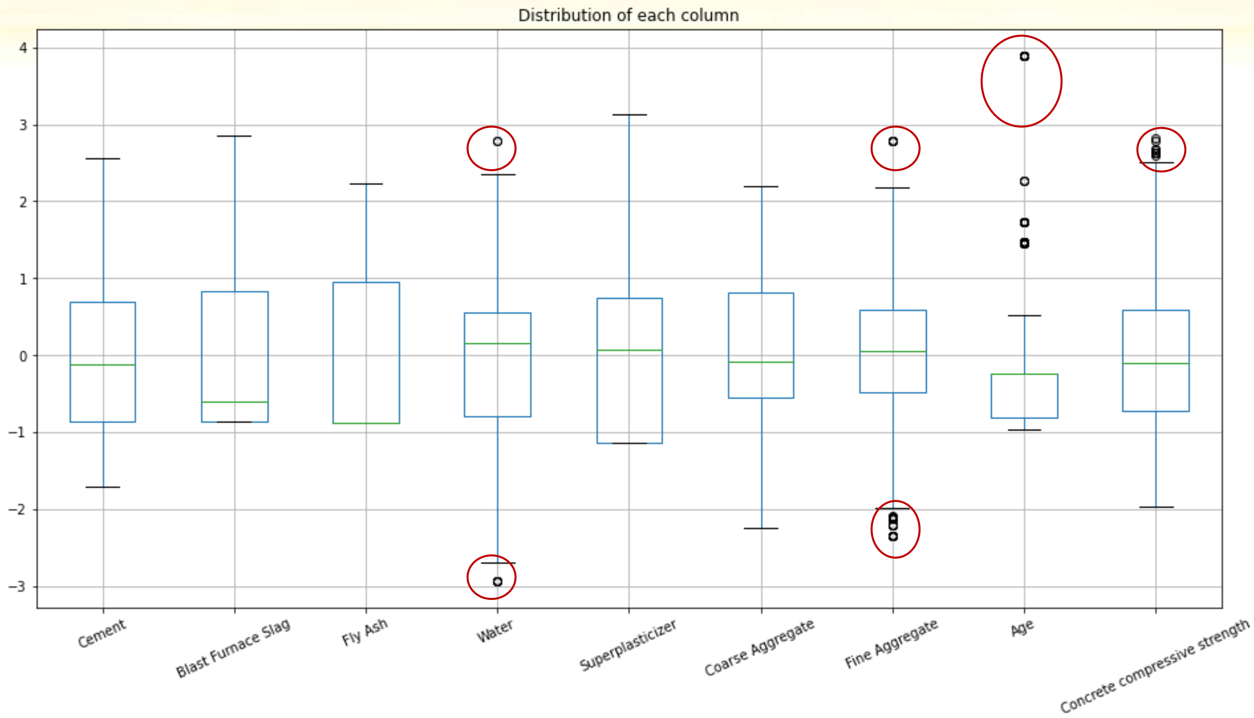
The **stronger** compressive strength ⟶ the **more durability**

⟶ the **lower workability**

Benefits of predicting strength of concrete based on early strength data:

- make the necessary adjustments to mix proportions
- increase the **efficiency** of construction
- balance the **workability** and **durability** of concrete

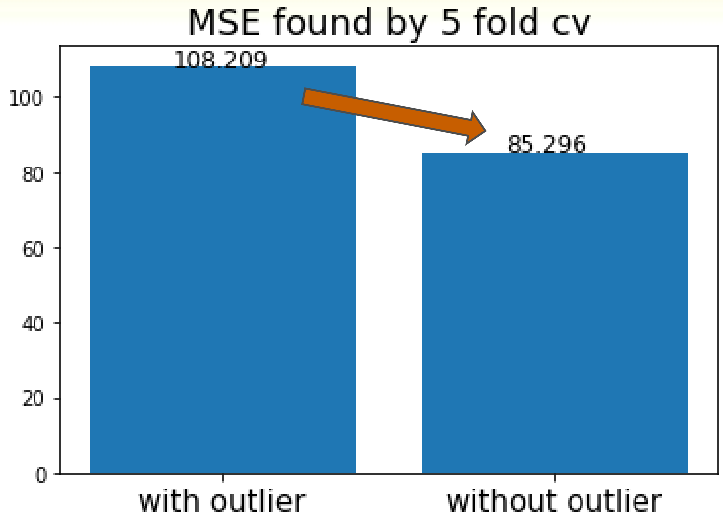
Data pre-processing



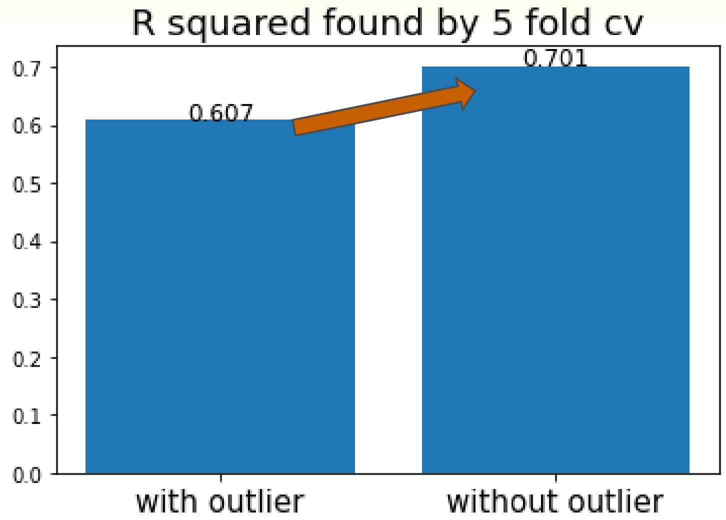
There are **some outliers exist in the dataset**

In order to **determine whether the outliers should be kept or not,** 5-fold cross validation is used to compare the prediction performance on **data set with outliers** and **data set without outliers.**
The results are shown in the next page

Cross validation on training dataset



Reduction in MSE



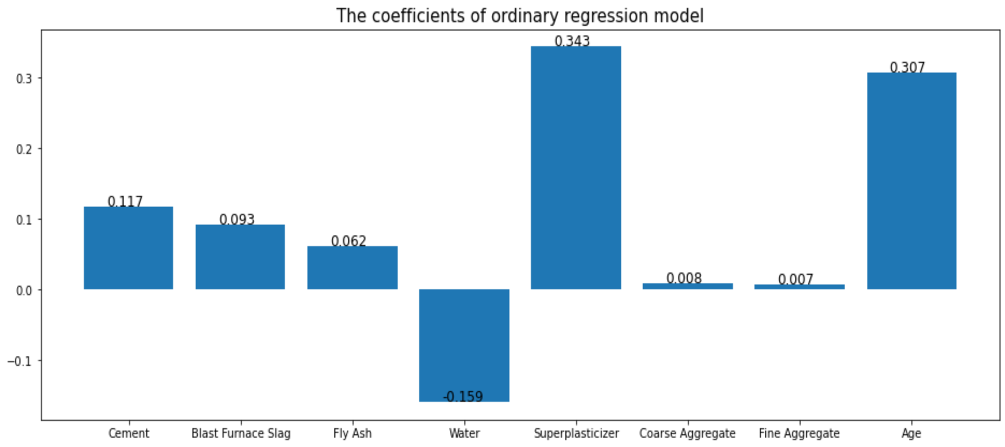
Increment in R squared

From the graphs above, after removing the outliers in the data set, the **MSE** is **reduced** and **R squared** is **improved**. It indicates **the outliers should be dropped from the data set** to improve the prediction accuracy.

49 rows are dropped from the dataset

The dataset is split into training dataset and testing dataset

Subset selection



	coef	std err	t	P>|t|	[0.025	0.975]
const	-4.6985	23.887	-0.197	0.844	-51.592	42.195
Cement	0.1172	0.008	15.164	0.000	0.102	0.132
Blast Furnace Slag	0.0926	0.009	9.801	0.000	0.074	0.111
Fly Ash	0.0620	0.012	5.293	0.000	0.039	0.085
Water	-0.1589	0.035	-4.494	0.000	-0.228	-0.089
Superplasticizer	0.3434	0.093	3.712	0.000	0.162	0.525
Coarse Aggregate	0.0084	0.009	0.979	0.328	-0.008	0.025
Fine Aggregate	0.0069	0.010	0.705	0.481	-0.012	0.026
Age	0.3072	0.011	29.157	0.000	0.286	0.328

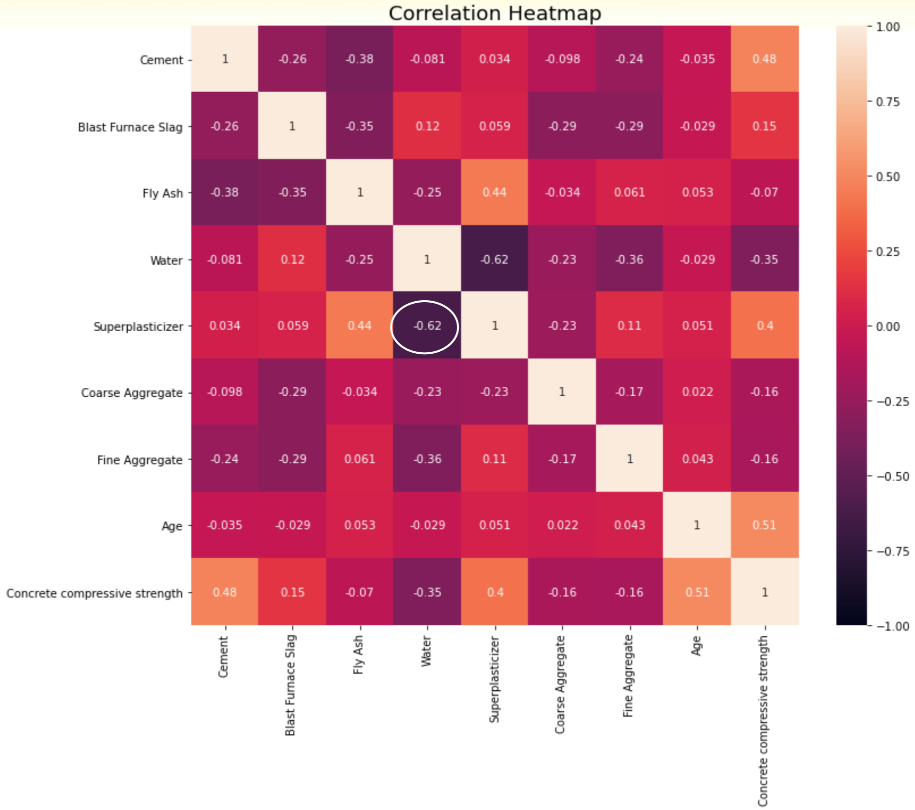
P-value of all variables <0.05 except the constant term

The result of ordinary regression is used to **check whether subset selection is needed.**
P-value of all columns are less than 0.05 → **no variable need to be dropped.**

From previous page, the training MSE of OLS is 85.296 and the R squared is 0.701.
The performance of OLS is not satisfactory.
Possible explanation:

- Multicollinearity might exist among the variables
- More flexibility of model might need to be offered

Correlation and multicollinearity



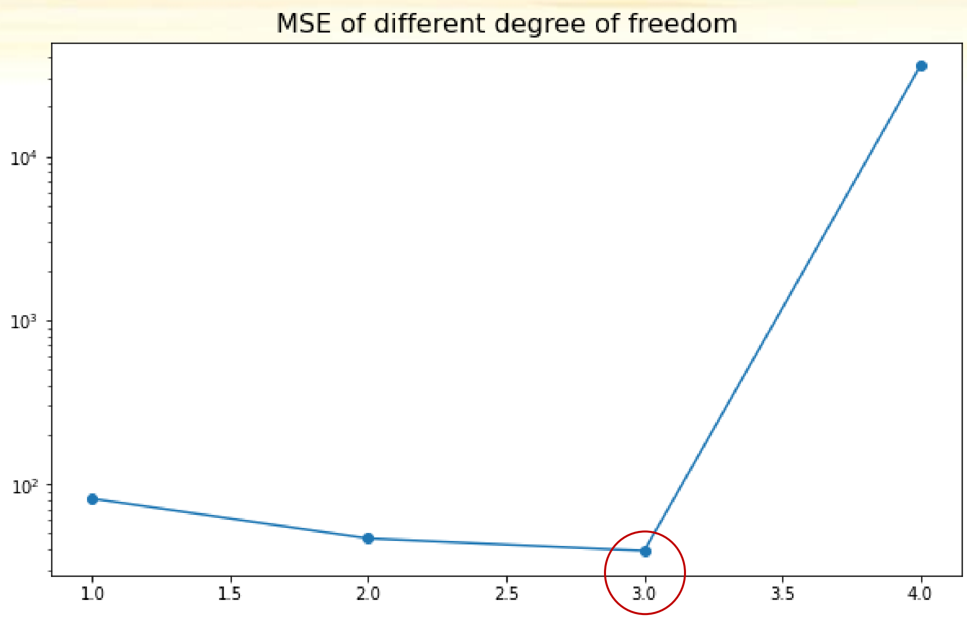
There is only **moderate correlation** between ‘Superplasticizer’ and ‘Water’

	VIF Factor	features
0	6377.5	const
1	7.1	Cement
2	7.5	Blast Furnace Slag
3	6.4	Fly Ash
4	5.2	Water
5	2.7	Superplasticizer
6	4.9	Coarse Aggregate
7	5.9	Fine Aggregate
8	1.0	Age

VIF of all variables <10
except the constant term

Moderate multicollinearity exists among the variables, it might affect the prediction accuracy of ordinary regression

Flexibility of model

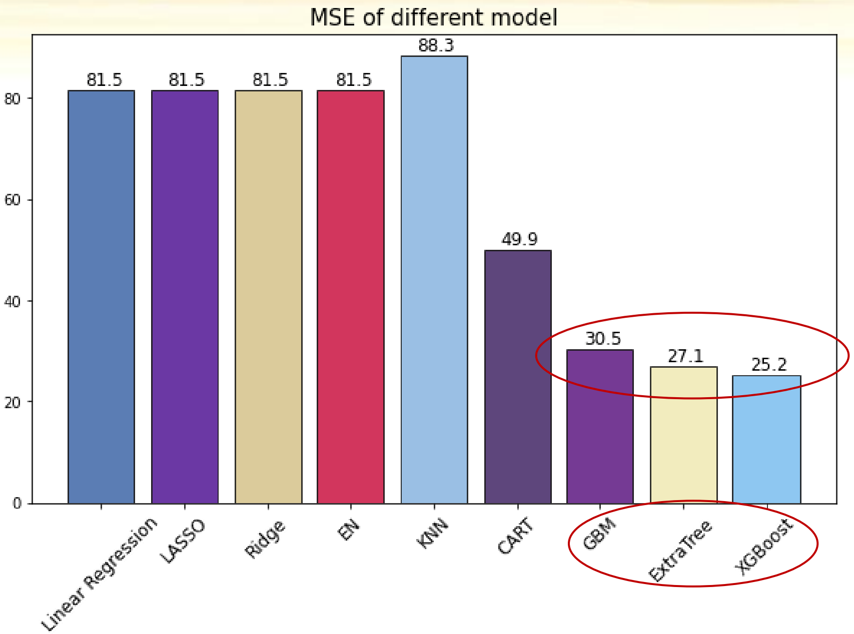


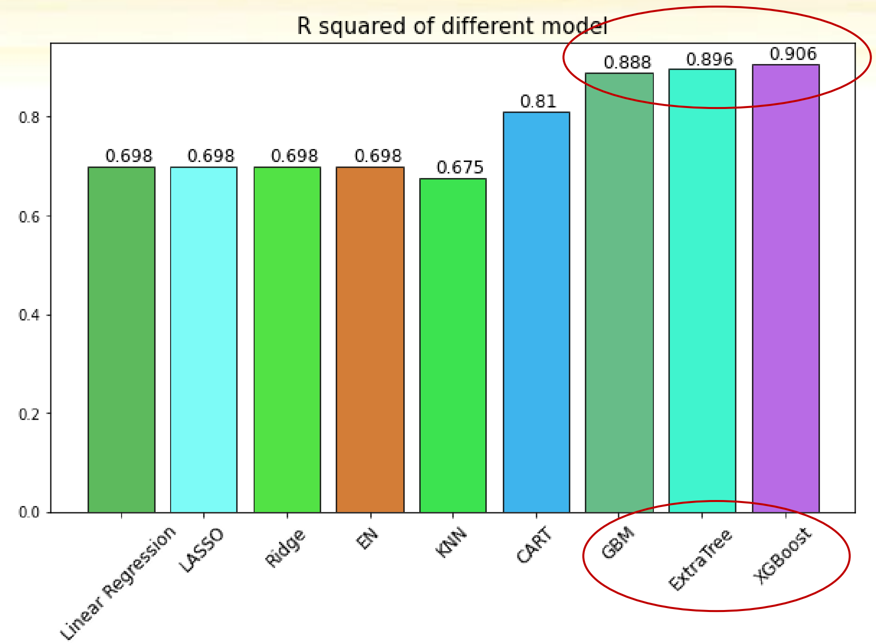

```
Degree 1: MSE=82.712149
Degree 2: MSE=48.467721
Degree 3: MSE=47.434586
Degree 4: MSE=12191112.073471
```

For sake of checking the flexibility of model needed, **polynomial regression** and **5-fold cross validation** are used to **find the suitable degree of freedom of model** in term of training MSE.

From the plot above, the **training MSE is minimized at degree 3**.
It indicates that **more flexible model should be used** for approximation instead of OLS.

Model selection





After 5-fold cross validation is used in model selection, **Gradient Boosting Regressor**, **Extra Trees Regressor**, and **XGBooster regressor** have the best prediction performance in terms of training MSE and R squared.

These 3 models would be used in prediction after hyperparameter tuning.

Hyperparameter tuning

Gradient boosting regressor

```
from sklearn.model_selection import GridSearchCV
GBR=GradientBoostingRegressor()
grid_params = {
    'n_estimators': [90, 100, 120, 180, 200],
    'learning_rate' : [0.01, 0.1, 0.05, 0.5, 1],
    'loss' : ['ls', 'lad', 'huber', 'quantile']
}

grid_search = GridSearchCV(GBR, grid_params, cv = 5, n_jobs = -1, verbose = 1)
grid_search.fit(X_train, y_train)
print(grid_search.best_params_)
print(grid_search.best_score_)
```

```
Fitting 5 folds for each of 100 candidates, totalling 500 fits
```

```
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done  34 tasks      | elapsed:    2.2s
[Parallel(n_jobs=-1)]: Done 184 tasks      | elapsed:   15.1s
[Parallel(n_jobs=-1)]: Done 434 tasks      | elapsed:   32.0s
[Parallel(n_jobs=-1)]: Done 500 out of 500 | elapsed:   38.5s finished
```

```
{'learning_rate': 0.1, 'loss': 'huber', 'n_estimators': 200}
0.9205091917834783
```

0.04 increment in R squared
0.88 → 0.92

Extra Tree regressor

```
model = ExtraTreesRegressor()
grid_params = {
     'n_estimators': [10,50,100],'criterion': ['mse'],'max_depth': [2,8,16,32,50],'min_samples_split': [2,4,6,8
    'bootstrap': [True, False],'warm_start': [True, False],       }
grid_search = GridSearchCV(model, grid_params, cv = 5, n_jobs = -1, verbose = 1)
grid_search.fit(X_train, y_train)
print(grid_search.best_params_)
print(grid_search.best_score_)
```

```
Fitting 5 folds for each of 1200 candidates, totalling 6000 fits
```

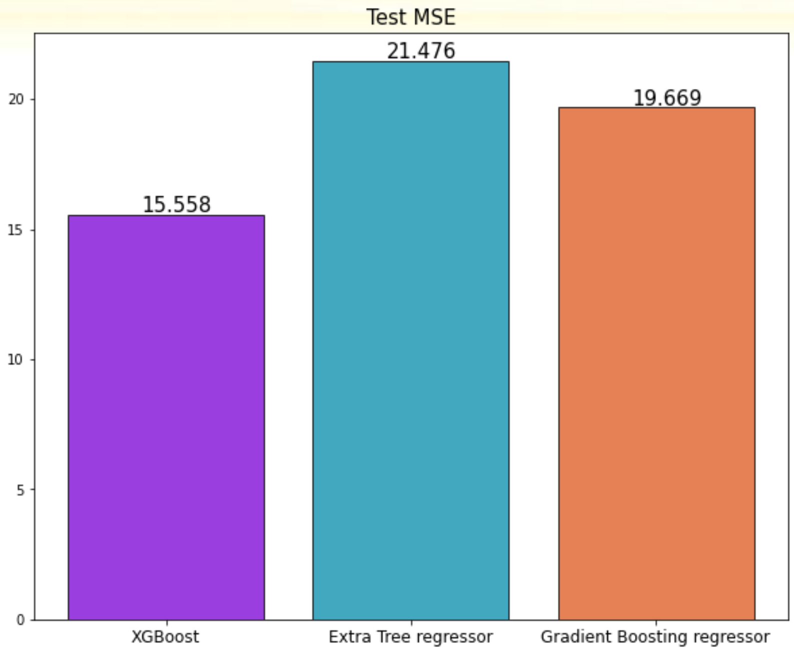
```
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done  52 tasks      | elapsed:    1.5s
[Parallel(n_jobs=-1)]: Done 352 tasks      | elapsed:    9.9s
[Parallel(n_jobs=-1)]: Done 852 tasks      | elapsed:   25.3s
[Parallel(n_jobs=-1)]: Done 1552 tasks      | elapsed:   47.6s
[Parallel(n_jobs=-1)]: Done 2452 tasks      | elapsed:  1.3min
[Parallel(n_jobs=-1)]: Done 3552 tasks      | elapsed:  1.7min
[Parallel(n_jobs=-1)]: Done 4852 tasks      | elapsed:  2.3min
[Parallel(n_jobs=-1)]: Done 6000 out of 6000 | elapsed:  2.8min finished
```

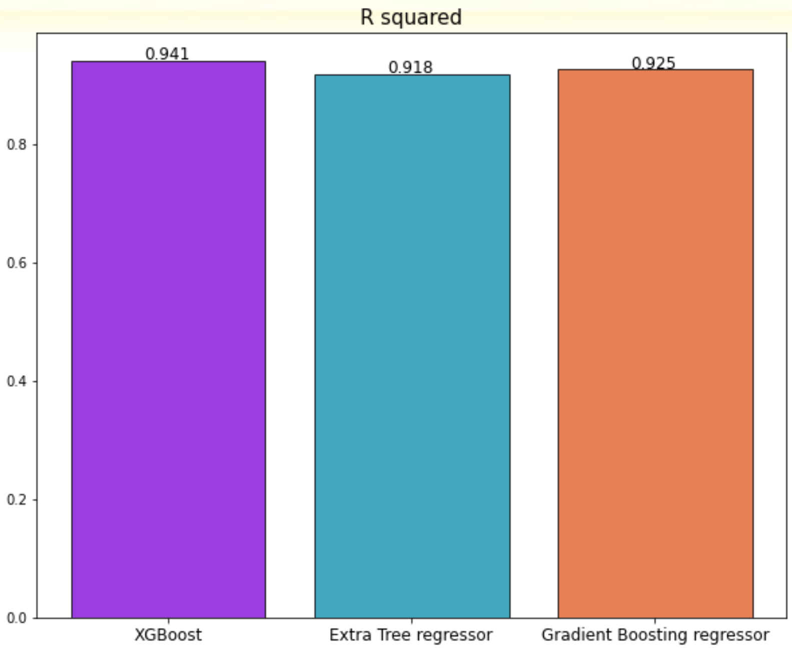
```
{'bootstrap': False, 'criterion': 'mse', 'max_depth': 16, 'max_features': 'auto', 'min_samples_leaf': 1, 'min
_samples_split': 2, 'n_estimators': 100, 'warm_start': False}
0.9193158413847344
```

0.03 increment in R squared
0.89 → 0.92

Models would be used to predict the test dataset

Prediction result





Finally, XGBoost regressor has the best performance on the test dataset after hyperparameter tuning. This model can be used to predict new data.

Summary

Data pre-processing part:

outliers are dropped from the dataset, and the necessary of subset selection is checked by the result of OLS. Moreover, the correlation between variables and VIF are investigated. Polynomial regression is utilized to investigate whether more flexibility of model should be given.

Model selection part:

Gradient Boosting Regressor, Extra Trees Regressor, and XGBooster regressor have the best prediction performance among all the candidates. After hyperparameter tuning, XGBooster regressor has the best prediction performance on test dataset.

Model application:

The final model can be used to predict the strength of concrete based on the early strength data. It enables us to make the necessary adjustments to mix proportions, increase the efficiency of construction, and balance the workability and durability of concrete

Reference

- Chopra, P., Sharma, R. K., & Kumar, M. (2014). PREDICTING COMPRESSIVE STRENGTH OF CONCRETE FOR VARYING WORKABILITY USING REGRESSION MODELS. *International Journal Of Engineering & Applied Sciences*, *6*(4), 10. https://doi.org/10.24107/ijeas.251233
- Concrete Compressive Strength Dataset:https://archive.ics.uci.edu/ml/datasets/Concrete+Compressive+Strength