

Linear Machine Learning model to predict Concrete Strength

Team 7:

Mohamed Eraky, Mahmoud Nassar, Anish Khilani

Contents

- Problem Description & Data analysis.
- Handling outliers.
- Machine learning model.
- PCA analysis.

Team collaboration (Live share on VS code)

The image illustrates the VS Code Live Share interface for team collaboration. It consists of several interconnected components:

- Live Share Sidebar:** Located on the left, it shows the 'LIVE SHARE' session. Under 'Participants (1)', 'Anish Khilani' is listed. Other sections include 'Shared Terminals', 'Shared Servers', 'Comments', and 'Session chat (1 new)'.
- Code Editor:** The main workspace displays a Jupyter Notebook file named 'preFinal.ipynb'. The code includes:

```
X = standardDataFrame.drop(columns = ['Strength', 'Blast Furnace Slag', 'Course Aggregate', 'Fine Aggregate'])
Y = standardDataFrame.drop(columns = ['Comment', 'Blast Furnace Slag', 'Superplasticizer', 'Course Aggregate', 'Fine Aggregate', 'Age', 'Water'])
X_train, x_test, Y_train, y_test = train_test_split(X, Y, test_size = 0.2, random_state = 42)

## Define the Linear Regression Model
model = linear_model.LinearRegression()
model.fit(X_train, Y_train)
y_pred = model.predict(x_test)
```

A red circle highlights the line `model = linear_model.LinearRegression()`, with a callout box containing the name 'Anish Khilani'.
- Terminal:** The bottom panel shows the execution output of the code, including:

```
## (Based on Standardization) Calculating the RMS Error, Model Coefficient, and
print('The model RMS =', np.sqrt(mean_squared_error(y_test, y_pred)))
print('The Model interception =', model.intercept_)
print('The Model accuracy =', r2_score(y_test, y_pred))
print('The Model coefficients =', model.coef_)
print('-----')
## Plot the predicted results vs. testing result
plt.figure(figsize = (8, 8))
plt.scatter(y_pred, y_test, s = 50, c = 'r', edgecolors = 'k')
plt.xlabel("Predicted Value", fontsize = 20, family = 'serif')
plt.ylabel("Tested Value", fontsize = 20, family = 'serif')
plt.show()
```

A red circle highlights the plotting code, with a callout box containing the name 'AgyleMotion'.
- VS Code Interface:** The top bar shows 'Visual Studio Live Share (Workspace)'. The bottom status bar indicates 'Jupyter Notebook'.

Problem description

Target : Concrete Strength

Number of Features : 8

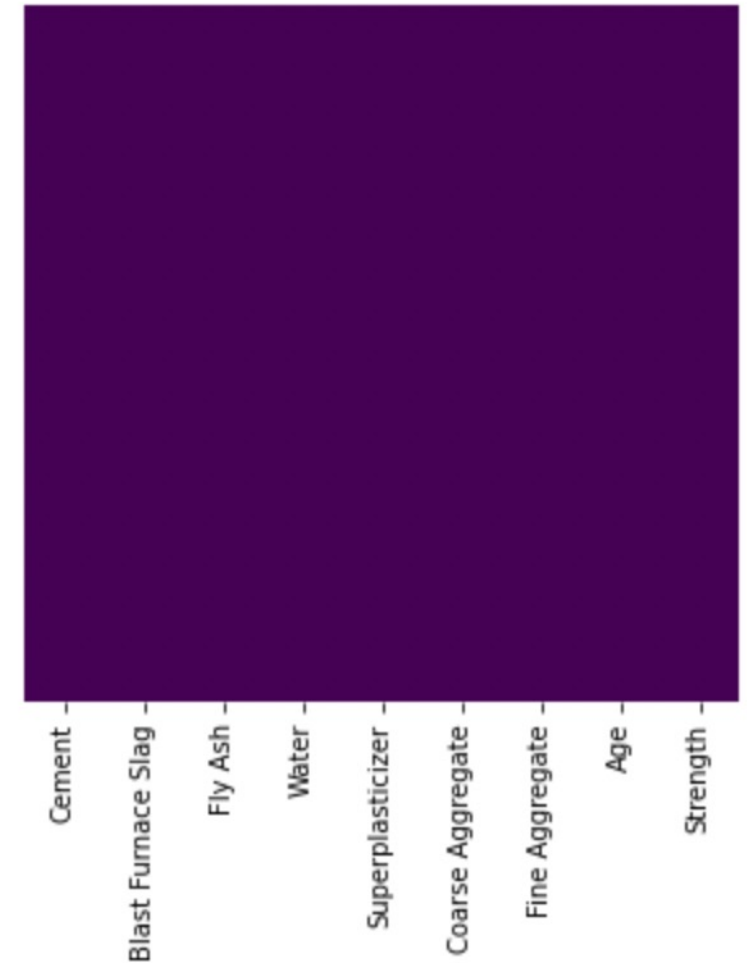
Least-square –Machine learning model , PCA

Concrete_Strength

Cement	Blast Furnace Slag	Fly Ash	Water	Superplasticizer	Coarse Aggregate	Fine Aggregate	Age	Strength
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

Data Analysis : Checking missing data

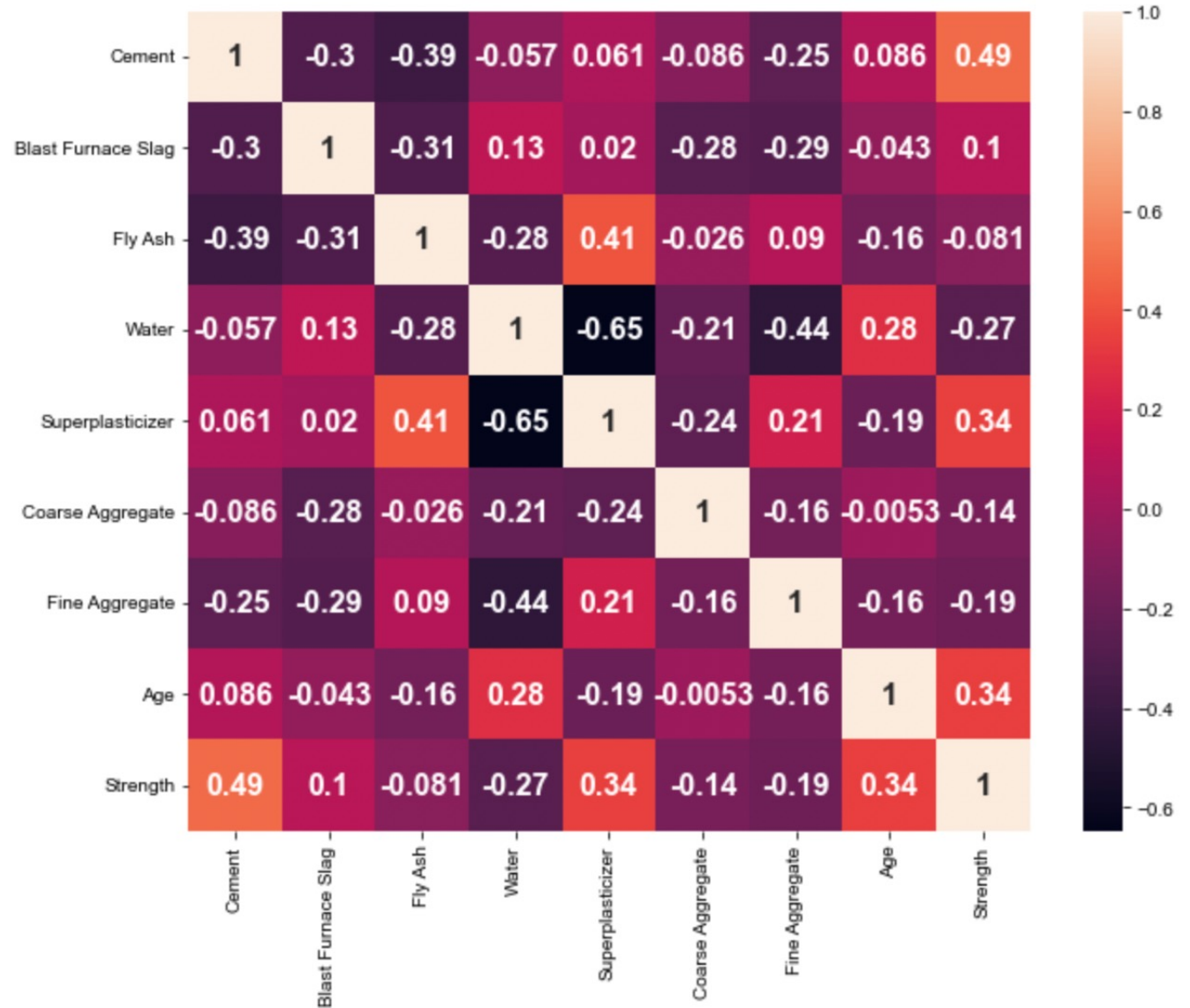
- Data frame shape (raw): 1030 x 9
- Data frame after removing Duplicates :1005 x 9
- Find if any missing data using heatmap.
- Used library by seaborn, and PyPlot to visualize the heatmap.



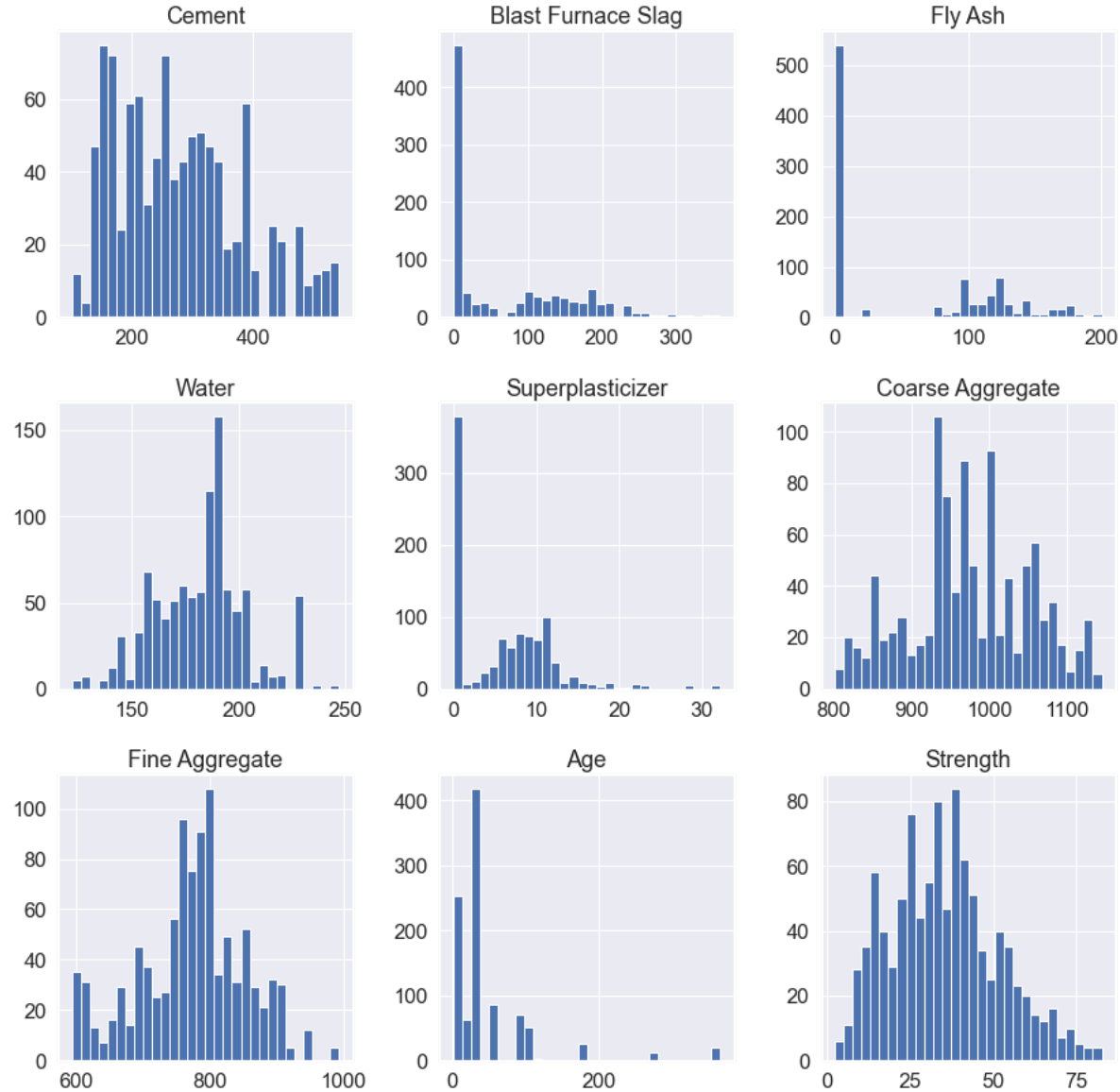
Heat Map

Finding correlation Matrix for Data frame

- Features : Cement, Superplasticizer, Age have the highest correlation with strength.
- Water has highest –Ve correlation with strength.
- Fly Ash has the lowest correlation with Strength.



- Data Analysis : Data Distribution

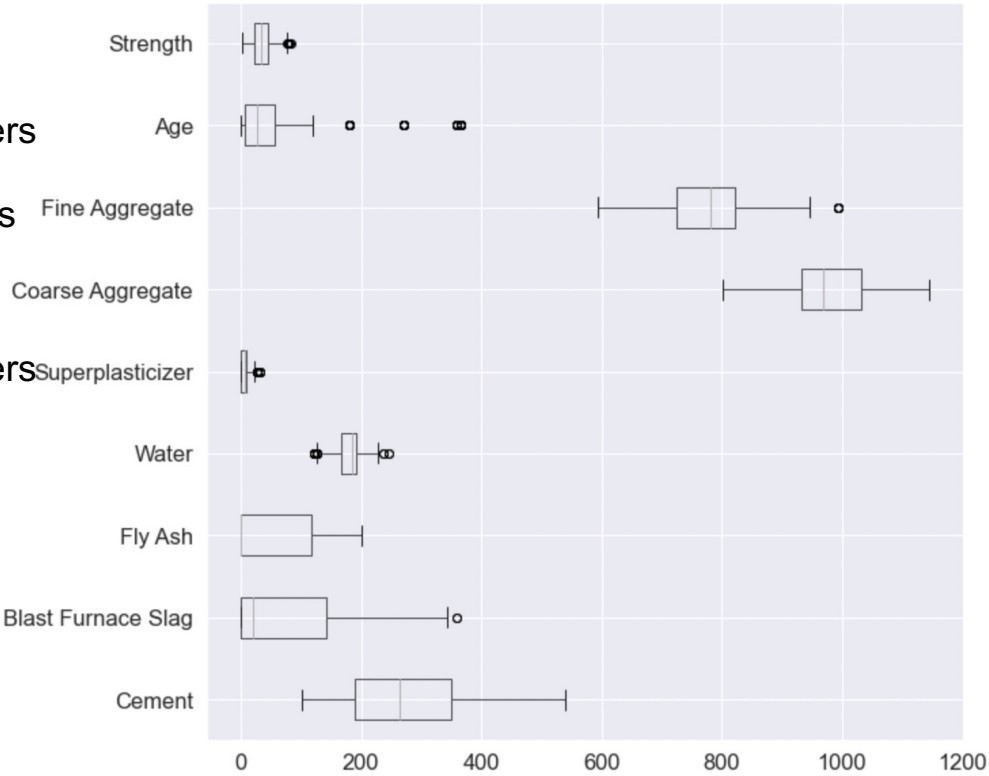


• Data Analysis : Plotting data distribution and box plot

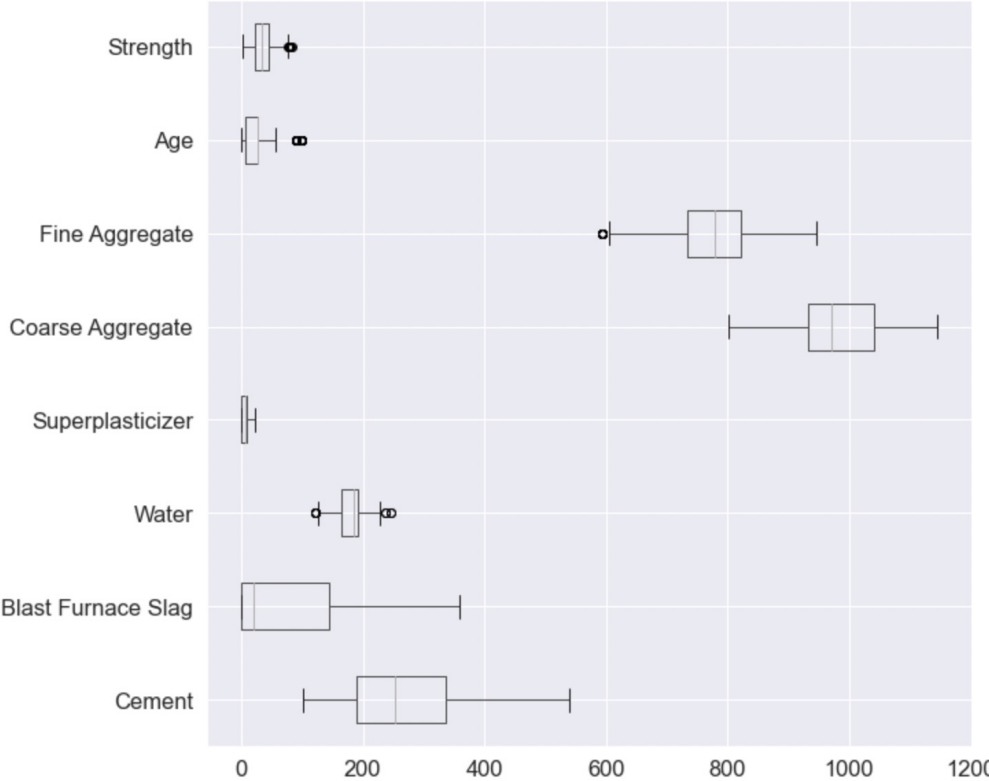
IQR, 62 outliers

IQR, 5 outliers

IQR, 10 outliers



	Cement	Blast Furnace Slag	Fly Ash	Water	Superplasticizer	Coarse Aggregate	Fine Aggregate	Age	Strength
count	1005.000000	1005.000000	1005.000000	1005.000000	1005.000000	1005.000000	1005.000000	1005.000000	1005.000000
mean	278.631343	72.043483	55.536318	182.075323	6.033234	974.376816	772.688259	45.856716	35.250378
std	104.344261	86.170807	64.207969	21.339334	5.919967	77.579667	80.340435	63.734692	16.284815
min	102.000000	0.000000	0.000000	121.800000	0.000000	801.000000	594.000000	1.000000	2.330000
25%	190.700000	0.000000	0.000000	166.600000	0.000000	932.000000	724.300000	7.000000	23.520000
50%	265.000000	20.000000	0.000000	185.700000	6.100000	968.000000	780.000000	28.000000	33.800000
75%	349.000000	142.500000	118.300000	192.900000	10.000000	1031.000000	822.200000	56.000000	44.870000
max	540.000000	359.400000	200.100000	247.000000	32.200000	1145.000000	992.600000	365.000000	82.600000



	Cement	Blast Furnace Slag	Water	Superplasticizer	Coarse Aggregate	Fine Aggregate	Age	Strength
count	928.000000	928.000000	928.000000	928.000000	928.000000	928.000000	928.000000	928.000000
mean	272.163901	73.107328	180.971659	6.082328	976.316164	774.576401	32.016164	34.357187
std	101.738846	87.165004	19.552706	5.248805	77.672976	75.277924	28.017038	16.313298
min	102.000000	0.000000	121.800000	0.000000	801.000000	594.000000	1.000000	2.330000
25%	189.050000	0.000000	165.600000	0.000000	932.000000	734.300000	7.000000	22.440000
50%	252.200000	20.000000	184.700000	6.700000	971.800000	779.500000	28.000000	33.085000
75%	336.125000	144.325000	192.000000	10.000000	1040.600000	821.000000	28.000000	44.280000
max	540.000000	359.400000	247.000000	22.100000	1145.000000	945.000000	100.000000	82.600000

Data Standardization & Normalization

- Data Standardization,
 - After standardization, $\mu = 0$

$$z = \frac{x - \mu}{\sigma}$$

	Cement	Blast Furnace Slag	Water	Superplasticizer	Coarse Aggregate	Fine Aggregate	Age	Strength
count	9.280000e+02	9.280000e+02	9.280000e+02	9.280000e+02	9.280000e+02	9.280000e+02	9.280000e+02	9.280000e+02
mean	-4.322004e-16	6.359257e-16	-1.001833e-15	5.535562e-16	-1.198754e-16	-6.713679e-16	7.515540e-16	-1.074631e-16
std	1.000539e+00	1.000539e+00	1.000539e+00	1.000539e+00	1.000539e+00	1.000539e+00	1.000539e+00	1.000539e+00
min	-1.673458e+00	-8.391756e-01	-3.027896e+00	-1.159427e+00	-2.258323e+00	-2.400090e+00	-1.107643e+00	-1.964315e+00
25%	-8.173743e-01	-8.391756e-01	-7.865892e-01	-1.159427e+00	-5.708557e-01	-5.353245e-01	-8.933726e-01	-7.309137e-01
50%	-1.963327e-01	-6.096020e-01	1.907844e-01	1.177421e-01	-5.817466e-02	6.544089e-02	-1.434245e-01	-7.802674e-02
75%	6.290182e-01	8.174849e-01	5.643356e-01	7.467956e-01	8.280679e-01	6.170286e-01	-1.434245e-01	6.085932e-01
max	2.634004e+00	3.286262e+00	3.378762e+00	3.053325e+00	2.172890e+00	2.265146e+00	2.427826e+00	2.958864e+00

- Data Normalization

- $x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}}$

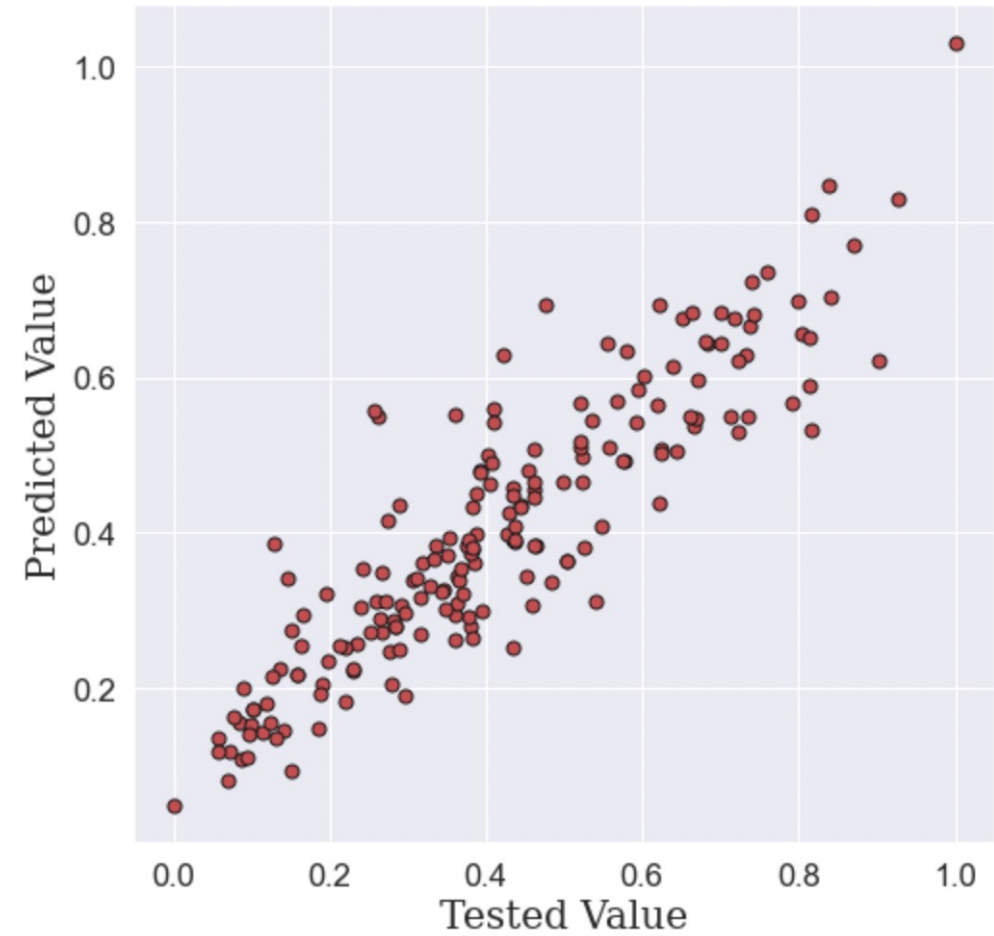
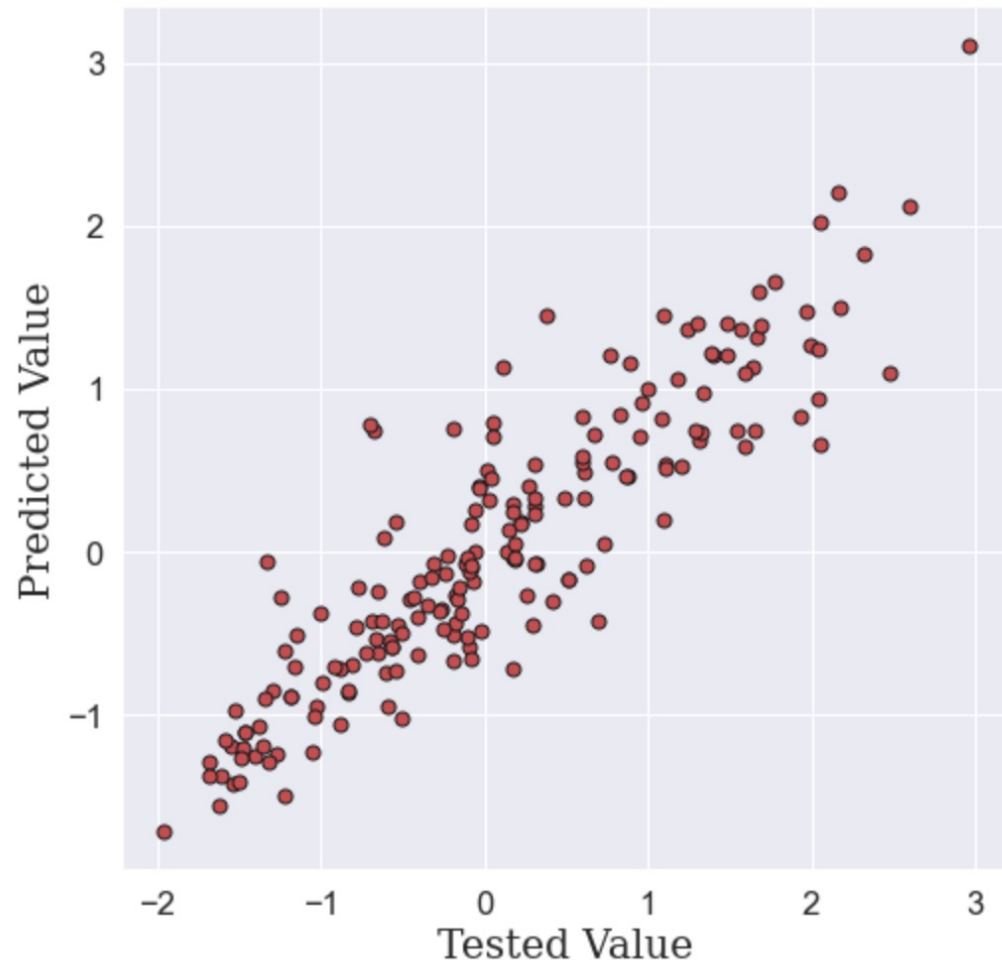
[illegible]

Machine learning model Results : Test cases

Linear regression model, split the data by 80 (train) – (test) 20%

Feature removed	RMS	Accuracy	Scaling
Fly Ash	0.473/ 0.0961	0.801	Standardized /Normalized
Fly Ash, Water, Blast Furnace Slag, Coarse & Fine Aggregate	0.622/ 0.126	0.657	Standardized /Normalized
Fly Ash, Blast Furnace Slag, Coarse & Fine Aggregate	0.606/0.123	0.674	Standardized /Normalized

Standardized & Normalized



Principal Component Analysis : PCA

Copy data frame and applied standardization.

Split them into features and target.

Calculate covariance Matrix.

Calculate the eigen values and Eigen vectors.

Reorder the Eigen vectors in descending order according to Eigen values and

Compute the features weight.

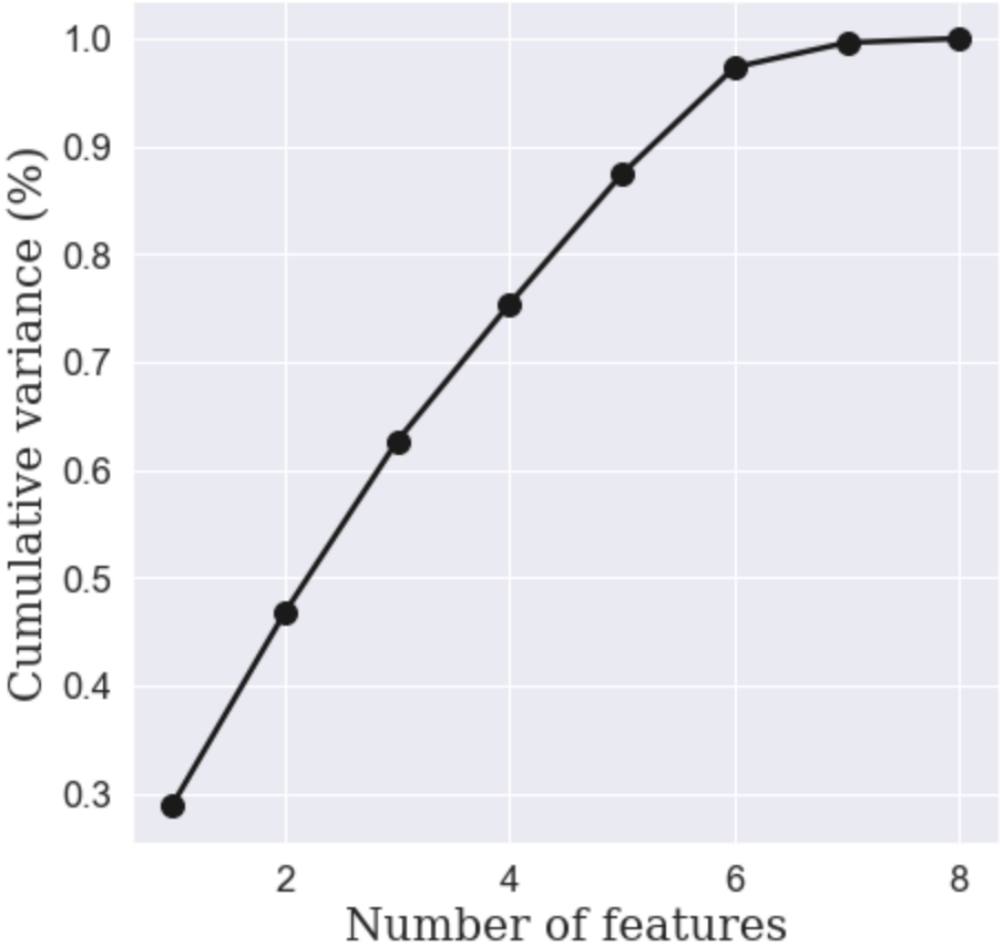
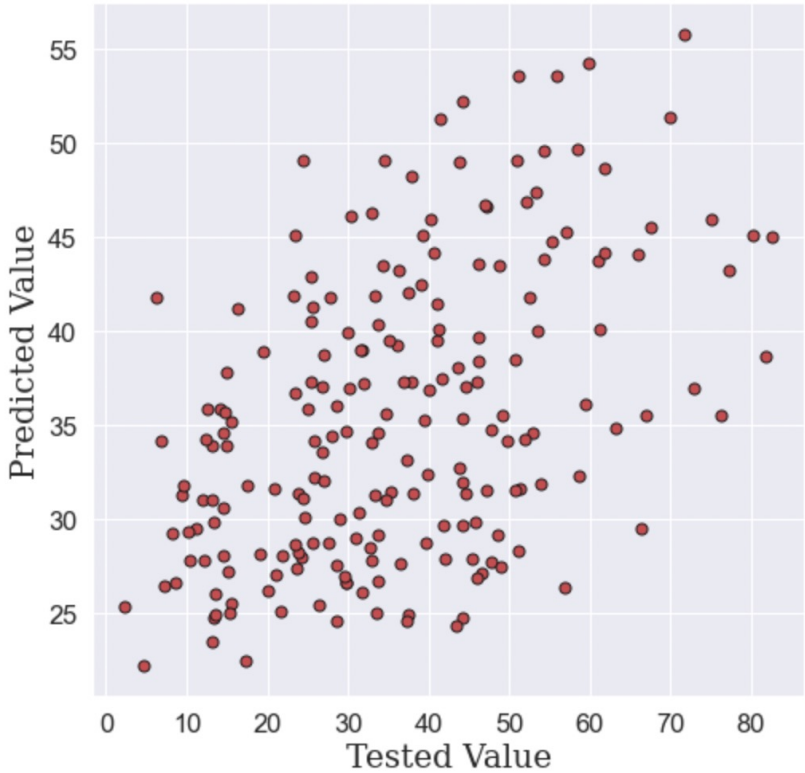
PCA- CONT.

Feature	Eigen Value
Cement	0.289
Water	0.1785
Super Plasticizer	0.159
Age	0.127
Fine Aggregate	0.120
Coarse Aggregate	0.098
Fly ash	0.0229
Blast Furnace slag	0.003

PCA-Results

K	RMS	Accuracy
2	15.247	0.2279
6	8.1176	0.788

K=2



Cumulative sum

Conclusion

- Linear regression model produced 0.801 accuracy upon excluding Fly Ash, excluding features gradually decreases model prediction accuracy.
- Considering only two components in PCA reduced the model prediction accuracy to 22.79 %.
- Considering 6 features (same as standard linear regression model) gave acceptable accuracy.