Concrete strenght prediction using Lasso regretion

In [50]:
```
#importing libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import Lasso
from sklearn. metrics import r2_score, mean_absolute_error, mean_squared_error
import warnings
warnings.filterwarnings('ignore')
```

In [10]:
```
df = pd.read_csv('data/concrete_data.csv')

df.head()
```

Out[10]:

	cement	blast_furnace_slag	fly_ash	water	superplasticizer	coarse_aggregate	fine_aggregate	age	concrete_cc
0	540.0	0.0	0.0	162.0	2.5	1040.0	676.0	28	
1	540.0	0.0	0.0	162.0	2.5	1055.0	676.0	28	
2	332.5	142.5	0.0	228.0	0.0	932.0	594.0	270	
3	332.5	142.5	0.0	228.0	0.0	932.0	594.0	365	
4	198.6	132.4	0.0	192.0	0.0	978.4	825.5	360	

In [11]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1030 entries, 0 to 1029
Data columns (total 9 columns):
 #   Column                         Non-Null Count  Dtype
---  ------                         --------------  -----
 0   cement                         1030 non-null   float64
 1   blast_furnace_slag             1030 non-null   float64
 2   fly_ash                        1030 non-null   float64
 3   water                          1030 non-null   float64
 4   superplasticizer               1030 non-null   float64
 5   coarse_aggregate               1030 non-null   float64
 6   fine_aggregate                 1030 non-null   float64
 7   age                            1030 non-null   int64
 8   concrete_compressive_strength  1030 non-null   float64
dtypes: float64(8), int64(1)
memory usage: 72.5 KB
```

In [12]:
```
df.shape
```

Out[12]:
```
(1030, 9)
```

Data Preprocessing

In [13]:
```
#checking for null values
df.isnull().sum()
```

Out[13]:
```
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
```

In [14]:
```
#checking for duplicate rows
df.duplicated().sum()
```

Out[14]:
```
25
```

In [21]:
```
df.loc[df.duplicated(), :]
```

Out[21]:

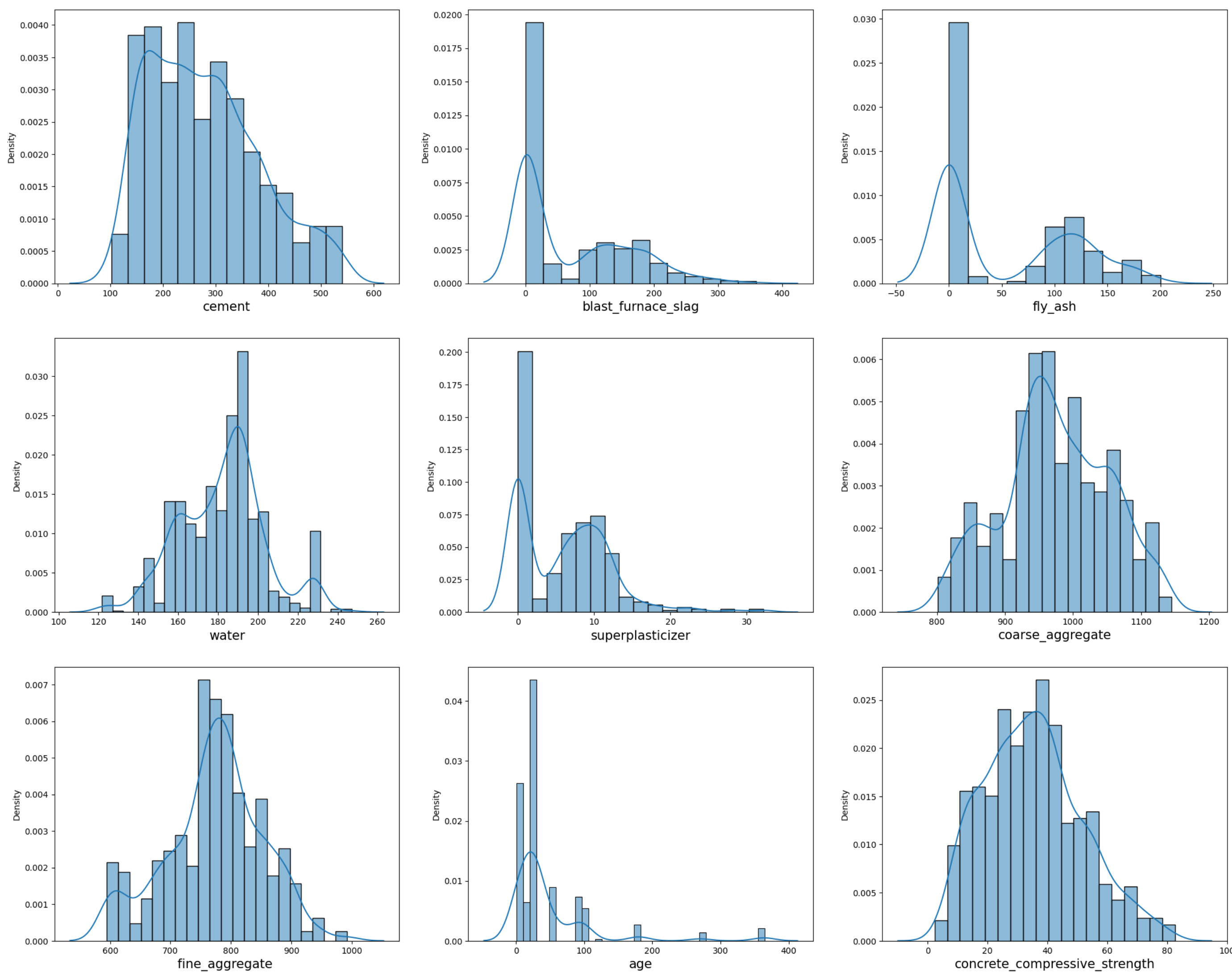
	cement	blast_furnace_slag	fly_ash	water	superplasticizer	coarse_aggregate	fine_aggregate	age	concrete
77	425.0	106.3	0.0	153.5	16.5	852.1	887.1	3	
80	425.0	106.3	0.0	153.5	16.5	852.1	887.1	3	
86	362.6	189.0	0.0	164.9	11.6	944.7	755.8	3	
88	362.6	189.0	0.0	164.9	11.6	944.7	755.8	3	
91	362.6	189.0	0.0	164.9	11.6	944.7	755.8	3	
100	425.0	106.3	0.0	153.5	16.5	852.1	887.1	7	
103	425.0	106.3	0.0	153.5	16.5	852.1	887.1	7	
109	362.6	189.0	0.0	164.9	11.6	944.7	755.8	7	
111	362.6	189.0	0.0	164.9	11.6	944.7	755.8	7	
123	425.0	106.3	0.0	153.5	16.5	852.1	887.1	28	
126	425.0	106.3	0.0	153.5	16.5	852.1	887.1	28	
132	362.6	189.0	0.0	164.9	11.6	944.7	755.8	28	
134	362.6	189.0	0.0	164.9	11.6	944.7	755.8	28	
137	362.6	189.0	0.0	164.9	11.6	944.7	755.8	28	
146	425.0	106.3	0.0	153.5	16.5	852.1	887.1	56	
149	425.0	106.3	0.0	153.5	16.5	852.1	887.1	56	
155	362.6	189.0	0.0	164.9	11.6	944.7	755.8	56	
157	362.6	189.0	0.0	164.9	11.6	944.7	755.8	56	
160	362.6	189.0	0.0	164.9	11.6	944.7	755.8	56	
169	425.0	106.3	0.0	153.5	16.5	852.1	887.1	91	
172	425.0	106.3	0.0	153.5	16.5	852.1	887.1	91	
177	362.6	189.0	0.0	164.9	11.6	944.7	755.8	91	
179	362.6	189.0	0.0	164.9	11.6	944.7	755.8	91	
182	362.6	189.0	0.0	164.9	11.6	944.7	755.8	91	
809	252.0	0.0	0.0	185.0	0.0	1111.0	784.0	28	

In [26]:
```
#dropping duplicate rows
df.drop_duplicates(keep='first', inplace=True)
df.shape
```

Out[26]: (1005, 9)

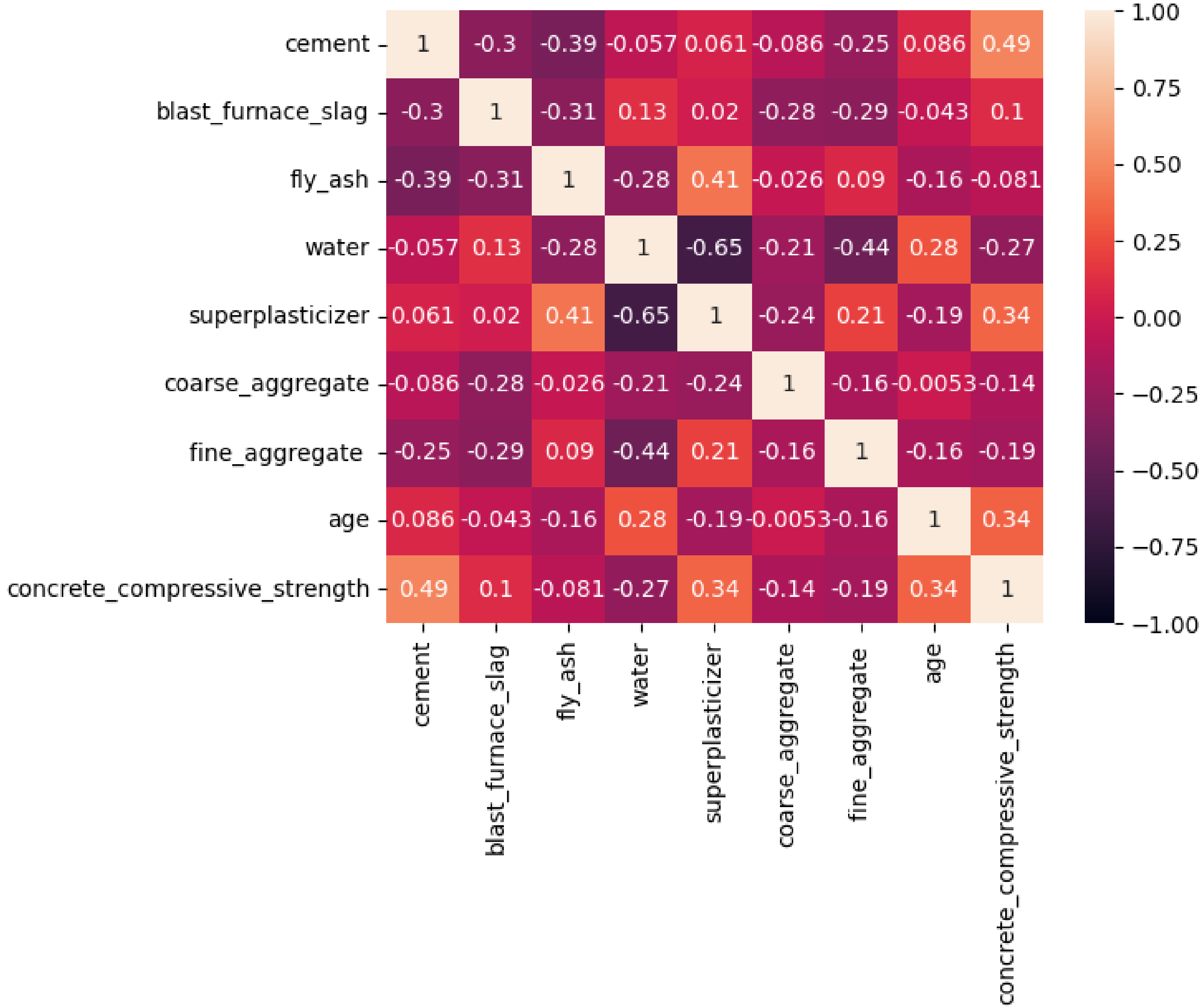
EDA

In [28]:
```
#plotting distributions of features

plt.figure(figsize = (25, 20))
plotnumber = 1

for col in df.columns:
    if plotnumber <= 9:
        ax = plt.subplot(3, 3, plotnumber)
        sns.histplot(df[col], kde=True, stat='density', kde_kws=dict(cut=3))
        plt.xlabel(col, fontsize = 15)

    plotnumber += 1
```



In [31]:
```
#correlatin matrix
sns.heatmap(df.corr(), annot=True, vmin=-1, vmax=1)
```

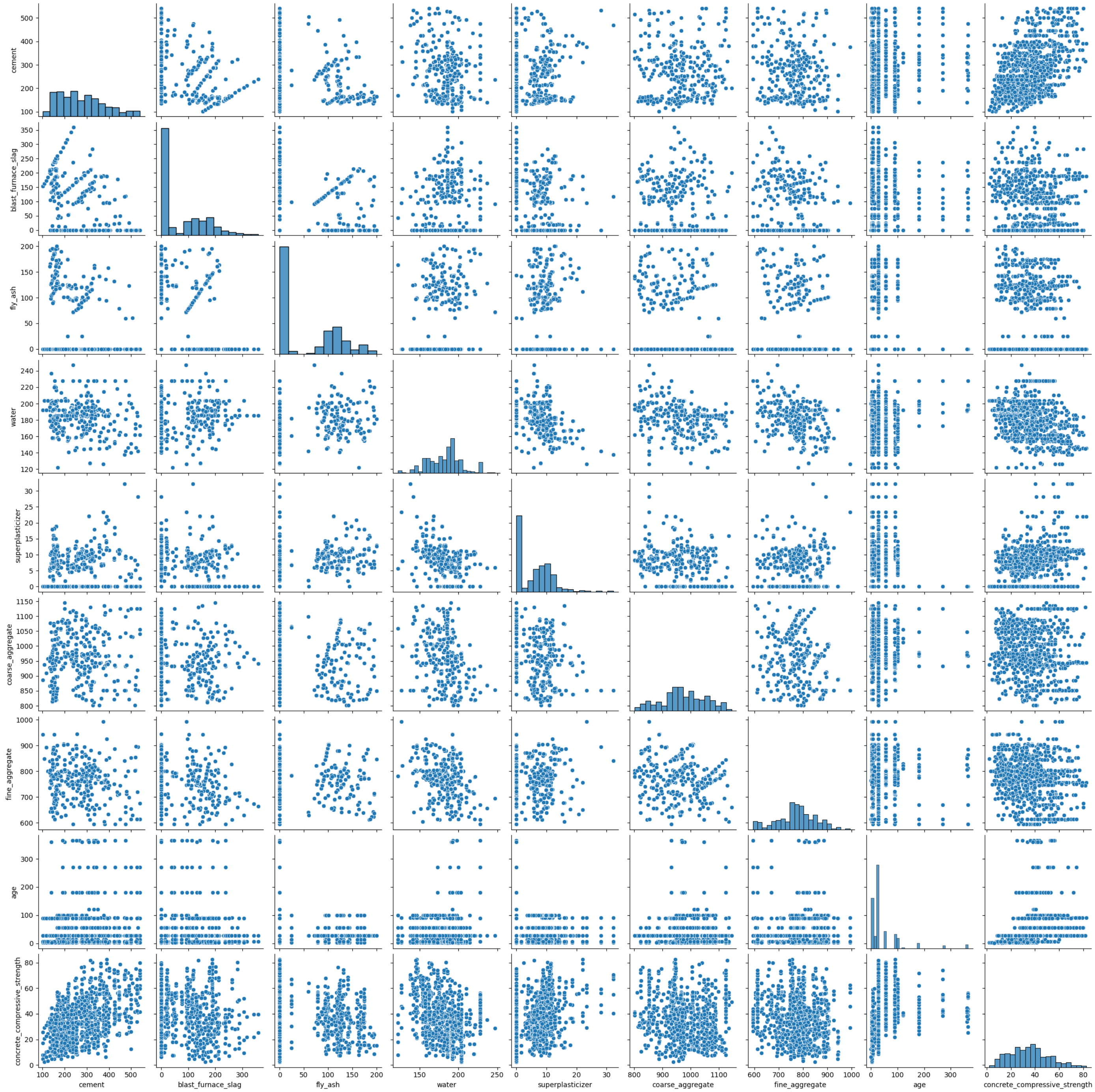
Out[31]: <AxesSubplot: >



- Seems there are no multicollinearity between features

In [32]:
```
#pairplot
sns.pairplot(df)
```

Out[32]:
```
<seaborn.axisgrid.PairGrid at 0x1e9f8957310>
```



- Cement and strength has linear relationship

Seperating features and target

In [33]:
```
X = df.iloc[:,:-1]
y = df.iloc[:,-1]
```

In [35]:
```
#splitting dataset into train and test sets
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=1)
```

Feature scaling

In [36]:
```
from sklearn.preprocessing import StandardScaler
```

```
scaler = StandardScaler()
scalerfit = scaler.fit(X_train)
X_train_scl = scalerfit.transform(X_train)
X_test_scl = scalerfit.transform(X_test)
```

Model training

- ### Linear regression

In [56]:
```
lr = LinearRegression()
lr.fit(X_train_scl, y_train)

for i, col in enumerate(X_train.columns):
    print('The coefficent for {} is {}'.format(col, lr.coef_[i]))
```

```
The coefficent for cement is 11.4564804619059
The coefficent for blast_furnace_slag is 7.635182506579105
The coefficent for fly_ash is 5.121438893949539
The coefficent for water is -4.380549966886574
The coefficent for superplasticizer is 1.301366307573757
The coefficent for coarse_aggregate is 0.4455926699073369
The coefficent for fine_aggregate  is 0.11771690867432172
The coefficent for age is 7.125608300916846
```

- ### Model evaluation

In [74]:
```
y_pred = lr.predict(X_test_scl)

r2 = r2_score(y_test, y_pred)
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)

print('R2 score: ', r2)
print('Mean absolute error: ', mae)
print('Mean squared error: ', mse)
print('Root mean squared error: ', rmse)
```

```
R2 score:  0.5499699178064184
Mean absolute error:  8.917839873008186
Mean squared error:  125.0541420367501
Root mean squared error:  11.182760930859162
```

- ### Lasso regression

In [57]:
```
#finding best value for alpha
from sklearn.model_selection import GridSearchCV

lasso = Lasso()
parameters = {'alpha':[1e-15,1e-10,1e-8,1e-3,1e-2,5e-2,1,5,10,20,30,35,40,45,50,55,100]}
lasso_regressor = GridSearchCV(lasso,parameters,scoring='neg_mean_squared_error',cv=5)
lasso_regressor.fit(X_train_scl,y_train)


lasso_regressor.best_params_
```

Out[57]:
```
{'alpha': 0.05}
```

In [58]:
```
lasso_regressor.best_score_
```

Out[58]: -102.87920928665832

In [71]:
```
#training the model
lasso = Lasso(alpha=0.05)
lasso.fit(X_train_scl, y_train)

for i, col in enumerate(X_train.columns):
    print('The coefficent for {} is {}'.format(col, lasso.coef_[i]))
```

```
The coefficent for cement is 10.8096250006607
The coefficent for blast_furnace_slag is 6.982465219334084
The coefficent for fly_ash is 4.546702730768315
The coefficent for water is -4.747505308943211
The coefficent for superplasticizer is 1.2699794919238228
The coefficent for coarse_aggregate is 0.0
The coefficent for fine_aggregate  is -0.372093231809397
The coefficent for age is 7.034176988616852
```

- ### Model evaluation

In [73]:
```
y_pred = lasso.predict(X_test_scl)

r2 = r2_score(y_test, y_pred)
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)

print('R2 score: ', r2)
print('Mean absolute error: ', mae)
print('Mean squared error: ', mse)
print('Root mean squared error: ', rmse)
```

```
R2 score:  0.5492034596064339
Mean absolute error:  8.944123667075363
Mean squared error:  125.26712507143723
Root mean squared error:  11.192279708416745
```

In [69]:
```
#coeffients shrinking to 0 when aplha value is high
lasso = Lasso(alpha=3)
lasso.fit(X_train_scl, y_train)

for i, col in enumerate(X_train.columns):
    print('The coefficent for {} is {}'.format(col, lasso.coef_[i]))
```

```
The coefficent for cement is 4.423900784941909
The coefficent for blast_furnace_slag is 0.0
The coefficent for fly_ash is 0.0
The coefficent for water is -0.490293514540079
The coefficent for superplasticizer is 2.6714813724364235
The coefficent for coarse_aggregate is -0.0
The coefficent for fine_aggregate  is -0.0
The coefficent for age is 2.78021828050698
```

In [ ]:
```
!jupyter nbconvert --to webpdf --allow-chromium-download appliance_energy_prediction.ipy
```