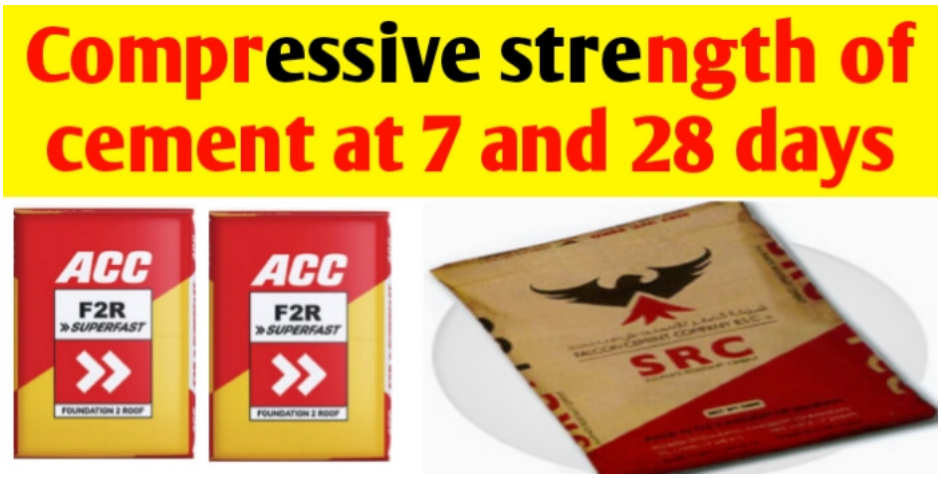
⭐ Regression Model to Predict Cement Compressive Strength




```
# import library
import pandas as pd
import numpy as np
```

```
# import data
cement = pd.read_csv('https://github.com/ybifoundation/Dataset/raw/main/Concrete%20Compressive%20Strength.csv')
```

```
# view data
cement.head()
```

	Cement (kg in a m^3 mixture)	Blast Furnace Slag (kg in a m^3 mixture)	Fly Ash (kg in a m^3 mixture)	Water (kg in a m^3 mixture)	Superplasticizer (kg in a m^3 mixture)	Coarse Aggregate (kg in a m^3 mixture)	Fine Aggregate (kg in a m^3 mixture)	(d
0	540.0	0.0	0.0	162.0	2.5	1040.0	676.0	
1	540.0	0.0	0.0	162.0	2.5	1055.0	676.0	
2	332.5	142.5	0.0	228.0	0.0	932.0	594.0	
3	332.5	142.5	0.0	228.0	0.0	932.0	594.0	
4	198.6	132.4	0.0	192.0	0.0	978.4	825.5	

```
# info of data
cement.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1030 entries, 0 to 1029
Data columns (total 9 columns):
 #   Column                                          Non-Null Count  Dtype  
---  ------                                          --------------  -----  
 0   Cement (kg in a m^3 mixture)                    1030 non-null   float64
 1   Blast Furnace Slag (kg in a m^3 mixture)        1030 non-null   float64
 2   Fly Ash (kg in a m^3 mixture)                   1030 non-null   float64
 3   Water (kg in a m^3 mixture)                     1030 non-null   float64
 4   Superplasticizer (kg in a m^3 mixture)          1030 non-null   float64
 5   Coarse Aggregate (kg in a m^3 mixture)          1030 non-null   float64
 6   Fine Aggregate (kg in a m^3 mixture)            1030 non-null   float64
 7   Age (day)                                       1030 non-null   int64  
 8   Concrete Compressive Strength(MPa, megapascals) 1030 non-null   float64
dtypes: float64(8), int64(1)
memory usage: 72.5 KB
```

```
# summary statistics
cement.describe()
```

	Cement (kg in a m^3 mixture)	Blast Furnace Slag (kg in a m^3 mixture)	Fly Ash (kg in a m^3 mixture)	Water (kg in a m^3 mixture)	Superplasticizer (kg in a m^3 mixture)	Coars Aggregat (kg in m^ mixture
count	1030.000000	1030.000000	1030.000000	1030.000000	1030.000000	1030.0000(
mean	281.165631	73.895485	54.187136	181.566359	6.203112	972.9185(
std	104.507142	86.279104	63.996469	21.355567	5.973492	77.7538
min	102.000000	0.000000	0.000000	121.750000	0.000000	801.0000(
25%	192.375000	0.000000	0.000000	164.900000	0.000000	932.0000(

```
# check for missing value
cement.isna().sum()
```

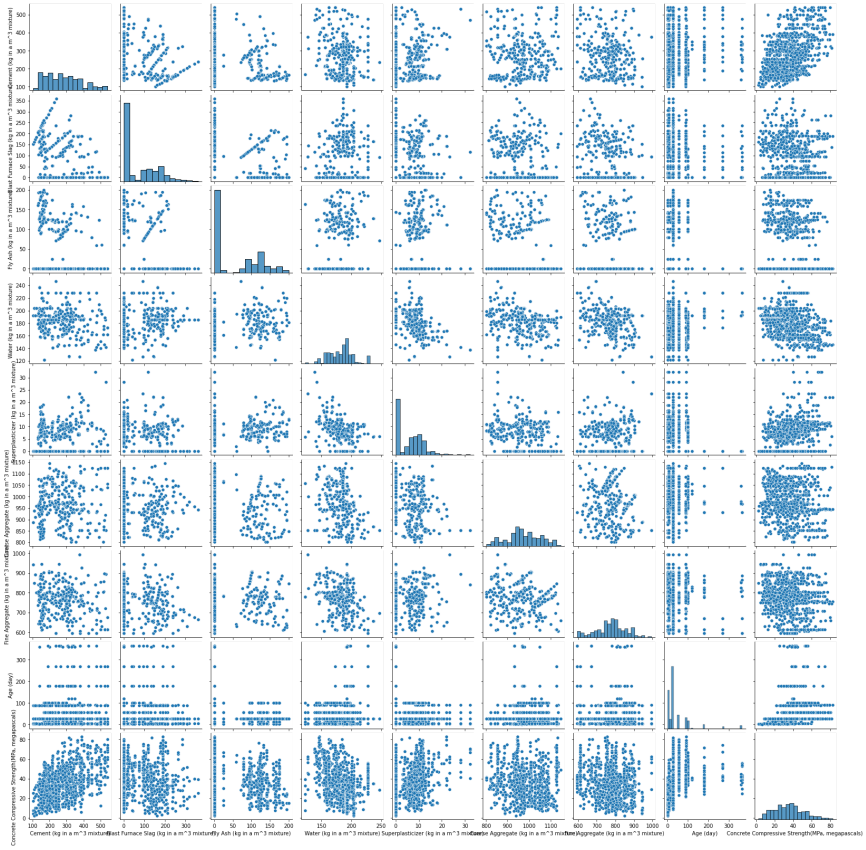
```
Cement (kg in a m^3 mixture)                        0
Blast Furnace Slag (kg in a m^3 mixture)            0
Fly Ash (kg in a m^3 mixture)                       0
Water (kg in a m^3 mixture)                         0
Superplasticizer (kg in a m^3 mixture)              0
Coarse Aggregate (kg in a m^3 mixture)              0
Fine Aggregate (kg in a m^3 mixture)                0
Age (day)                                           0
Concrete Compressive Strength(MPa, megapascals)     0
dtype: int64
```

```
# check for categories
cement.nunique()
```

```
Cement (kg in a m^3 mixture)                      280
Blast Furnace Slag (kg in a m^3 mixture)          187
Fly Ash (kg in a m^3 mixture)                     163
Water (kg in a m^3 mixture)                       205
Superplasticizer (kg in a m^3 mixture)            155
Coarse Aggregate (kg in a m^3 mixture)            284
Fine Aggregate (kg in a m^3 mixture)              304
Age (day)                                          14
Concrete Compressive Strength(MPa, megapascals)   938
dtype: int64
```

```
# visualize pairplot
import seaborn as sns
sns.pairplot(cement)
```

```
<seaborn.axisgrid.PairGrid at 0x7fe0bf032d60>
```



```
# columns name
cement.columns
```

```
Index(['Cement (kg in a m^3 mixture)',
       'Blast Furnace Slag (kg in a m^3 mixture)',
       'Fly Ash (kg in a m^3 mixture)', 'Water (kg in a m^3 mixture)',
       'Superplasticizer (kg in a m^3 mixture)',
       'Coarse Aggregate (kg in a m^3 mixture)',
       'Fine Aggregate (kg in a m^3 mixture)', 'Age (day)',
       'Concrete Compressive Strength(MPa, megapascals) '],
      dtype='object')
```

```
# define y
y=cement['Concrete Compressive Strength(MPa, megapascals) ']
```

```
# define X
X=cement[['Cement (kg in a m^3 mixture)',
'Blast Furnace Slag (kg in a m^3 mixture)',
'Fly Ash (kg in a m^3 mixture)', 'Water (kg in a m^3 mixture)',
'Superplasticizer (kg in a m^3 mixture)',
'Coarse Aggregate (kg in a m^3 mixture)',
'Fine Aggregate (kg in a m^3 mixture)', 'Age (day)']]
```

```
# split data
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,train_size=0.7,random_state=2559)
```

```
# verify shape
X_train.shape,X_test.shape,y_train.shape,y_test.shape
((721, 8), (309, 8), (721,), (309,))
```

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

```
# select model
from sklearn.linear_model import LinearRegression
model=LinearRegression()
```

```
# train model
model.fit(X_train,y_train)
```

```
LinearRegression()
```

```
# predict with model
y_pred=model.predict(X_test)
```

```
# model evaluation
from sklearn.metrics import mean_absolute_error,mean_absolute_percentage_error,mean_squared_error
```

```
# model MAE
mean_absolute_error(y_test,y_pred)
```

```
7.814891951068712
```

```
# model MAPE
mean_absolute_percentage_error(y_test,y_pred)
```

```
0.28040027489426594
```

```
# model MSE
mean_squared_error(y_test,y_pred)
```

```
102.62674212692517
```

```
# future prediction
X.sample()
```

	Cement (kg in a m^3 mixture)	Blast Furnace Slag (kg in a m^3 mixture)	Fly Ash (kg in a m^3 mixture)	Water (kg in a m^3 mixture)	Superplasticizer (kg in a m^3 mixture)	Coarse Aggregate (kg in a m^3 mixture)	Fine Aggregate (kg in a m^3 mixture)	Age (day)
823	322.0	0.0	0.0	203.0	0.0	974.0	800.0	180

```
# define X_new
X_new=X.sample()
X_new
```

	Cement (kg in a m^3 mixture)	Blast Furnace Slag (kg in a m^3 mixture)	Fly Ash (kg in a m^3 mixture)	Water (kg in a m^3 mixture)	Superplasticizer (kg in a m^3 mixture)	Coarse Aggregate (kg in a m^3 mixture)	Fine Aggregate (kg in a m^3 mixture)	Age (day)
836	304.0	140.0	0.0	214.0	6.0	895.0	722.0	28

```
# predict for X_new
model.predict(X_new)
```

```
array([31.10002048])
```