```
import pandas as pd
import numpy as np
```

```
data=pd.read_csv("concrete_data.csv")
```

```
data.sample(5)
```

	Cement	Blast Furnace Slag	Fly Ash	Water	Superplasticizer	Coarse Aggregate	Fine Aggregate	Age	Strengt
632	325.0	0.0	0.0	184.0	0.0	1063.0	783.0	28	30.5
392	333.0	17.5	163.0	167.0	17.9	996.0	652.0	28	47.2
946	149.5	236.0	0.0	175.8	12.6	846.8	892.7	28	32.9
965	330.5	169.6	0.0	194.9	8.1	811.0	802.3	28	56.6

```
data.isnull().sum()
```

```
Cement              0
Blast Furnace Slag  0
Fly Ash             0
Water               0
Superplasticizer    0
Coarse Aggregate    0
Fine Aggregate      0
Age                 0
Strength            0
dtype: int64
```

```
data['Water'].corr(data['Strength'])
```

```
-0.2896333849853048
```

```
data.corr()
```

	Cement	Blast Furnace Slag	Fly Ash	Water	Superplasticizer	Coarse Aggregate	Fine Aggregate	Age	Strength
Cement	1.000000	-0.275216	-0.397467	-0.081587	0.092386	-0.109349	-0.222718	0.081946	0.497832
Blast Furnace Slag	-0.275216	1.000000	-0.323580	0.107252	0.043270	-0.283999	-0.281603	-0.044246	0.134829
Fly Ash	-0.397467	-0.323580	1.000000	-0.256984	0.377503	-0.009961	0.079108	-0.154371	-0.105755
Water	-0.081587	0.107252	-0.256984	1.000000	-0.657533	-0.182294	-0.450661	0.277618	-0.289633
Superplasticizer	0.092386	0.043270	0.377503	-0.657533	1.000000	-0.265999	0.222691	-0.192700	0.366079
Coarse Aggregate	-0.109349	-0.283999	-0.009961	-0.182294	-0.265999	1.000000	-0.178481	-0.003016	-0.164935
Fine Aggregate	-0.222718	-0.281603	0.079108	-0.450661	0.222691	-0.178481	1.000000	-0.156095	-0.167241
Age	0.081946	-0.044246	-0.154371	0.277618	-0.192700	-0.003016	-0.156095	1.000000	0.328873
Strength	0.497832	0.134829	-0.105755	-0.289633	0.366079	-0.164935	-0.167241	0.328873	1.000000

```
X=data.drop(columns='Strength').values
y=data['Strength'].values.reshape(-1,1)
```

```
print(np.shape(X))#sklearn needs data in 2d list
```

```
(1030, 8)
```

```
print(np.shape(y))
```

```
(1030, 1)
```

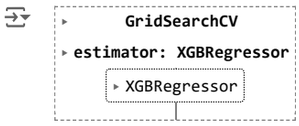
```
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import Lasso
from sklearn.model_selection import cross_val_score
lg=LinearRegression()
lasso=Lasso()
scoring=cross_val_score(lasso,X,y,scoring='r2',cv=5)
print(scoring.mean())
```

```
0.4653339617210076
```

```
from xgboost import XGBRegressor
from sklearn.model_selection import GridSearchCV
xgb=XGBRegressor()
parameters={
    "n_estimators":[100,200,500],
    "max_depth":[3,6,9],
    "gamma":[0.01,0.1],
    "learning_rate":[0.001,0.01,0.1,1]
}
clf=GridSearchCV(xgb,parameters,cv=5,scoring='r2')
```

```
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=42)
```

```
clf.fit(X_train,y_train)
```




```
clf.best_params_
```

```
{'gamma': 0.01, 'learning_rate': 0.1, 'max_depth': 3, 'n_estimators': 500}
```

```
clf.best_score_
```

```
0.9261406696848734
```

```
clf.best_estimator_
```

```
XGBRegressor(base_score=None, booster=None, callbacks=None,
             colsample_bylevel=None, colsample_bynode=None,
             colsample_bytree=None, device=None, early_stopping_rounds=None,
             enable_categorical=False, eval_metric=None, feature_types=None,
             gamma=0.01, grow_policy=None, importance_type=None,
             interaction_constraints=None, learning_rate=0.1, max_bin=None,
             max_cat_threshold=None, max_cat_to_onehot=None,
             max_delta_step=None, max_depth=3, max_leaves=None,
             min_child_weight=None, missing=nan, monotone_constraints=None,
             multi_strategy=None, n_estimators=500, n_jobs=None,
             num_parallel_tree=None, random_state=None, ...)
```

```
best_model=clf.best_estimator_
test_score=best_model.score(X_test,y_test)
print(test_score)
```

```
0.9226101072781454
```

```
best_model.predict([[325.0, 0.0,    0.0 ,184.0, 0.0,    1063.0, 783.0   ,28 ]])
```

```
array([31.081495], dtype=float32)
```

```
Start coding or generate with AI.
```