

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

df = pd.read_excel('ML_prediction1.xlsx')
df1 = pd.read_excel('ML_prediction2.xlsx')

X = df[['x1','x2','x3','x4','x5','x6','x7','x8','x9','x10','x11']]
X1 = df[['x1','x2','x3','x4','x5','x6','x7','x8','x9','x10','x11']]
y = df['y']

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=101)

from sklearn import svm
svr = svm.SVR(C=60,gamma=0.075,epsilon=0.08) #change the parameters
svr.fit(X_train,y_train)
df['prediction1'] = svr.predict(X)
Y1 = svr.predict(X1)
svr.score(X_test,y_test)

from sklearn import neighbors
```

```
knn =  
neighbors.KNeighborsRegressor(algorithm='auto',leaf_size=30,n_neighbors=3,p=1)#change the  
parameters  
  
knn.fit(X_train,y_train)  
  
df['prediction2'] = knn.predict(X)  
  
df2['prediction2'] = knn.predict(X2)  
  
knn.score(X_test,y_test)
```

```
from sklearn import ensemble  
  
rf = ensemble.RandomForestRegressor(n_estimators=50,random_state=10)#change the  
parameters  
  
rf.fit(X_train,y_train)  
  
df['prediction3'] = rf.predict(X)  
  
rf.score(X_test,y_test)
```

```
from sklearn import ensemble  
  
gbdt = ensemble.GradientBoostingRegressor(n_estimators=30,random_state=10)#change the  
parameters  
  
gbdt.fit(X_train,y_train)  
  
df['prediction4'] = gbdt.predict(X)  
  
gbdt.score(X_test,y_test)
```

```
# efficiency  
  
import pandas as pd  
  
import numpy as np  
  
import matplotlib.pyplot as plt  
  
%matplotlib inline
```

```
df = pd.read_excel('efficiency1.xlsx')

df2 = pd.read_excel('efficiency2.xlsx')

X = df[['x1','x2','x3','x4','x5','x6','x7','x8','x9','x10','x11','x12','x13']]

X2 = df2[['x1','x2','x3','x4','x5','x6','x7','x8','x9','x10','x11','x12','x13']]

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=101)

from sklearn import svm

svr = svm.SVR(C=25,gamma=0.0002,epsilon=0.1) #change the parameters

svr.fit(X_train,y_train)

df['prediction1'] = svr.predict(X)

df2['prediction1'] = svr.predict(X2)

svr.score(X_test,y_test)

from sklearn import neighbors

knn =

neighbors.KNeighborsRegressor(algorithm='auto',leaf_size=30,n_neighbors=2,p=4)#change the parameters

knn.fit(X_train,y_train)

df['prediction2'] = knn.predict(X)

df2['prediction2'] = knn.predict(X2)

knn.score(X_test,y_test)
```

```
from sklearn import ensemble

rf = ensemble.RandomForestRegressor(n_estimators=8,random_state=10)#change the
parameters

rf.fit(X_train,y_train)

df['prediction3'] = rf.predict(X)

df2['prediction3'] = rf.predict(X2)

rf.score(X_test,y_test)
```

```
from sklearn import ensemble

gbrt = ensemble.GradientBoostingRegressor(n_estimators=20,random_state=10)#change the
parameters

gbrt.fit(X_train,y_train)

df['prediction4'] = gbdt.predict(X)

df2['prediction4'] = gbdt.predict(X2)

gbdt.score(X_test,y_test)
```