```
import time

from sklearn import preprocessing \
    as preproc
from sklearn import linear_model \
    as linmod
import pandas as pd
from sklearn.linear_model \
    import RidgeCV
from sklearn import \
    model_selection
from sklearn import metrics
import numpy as np

def showStats(W, X, Yact, Ypred):
    print("R2 = %f, MSE = %f" % (
        mlr.score(X, Yact),
        metrics.mean_squared_error(
            Yact, Ypred)))


df = pd.read_hdf("DatasetPart2.h5", key='ML')

x = df[['NEXTDAYPRECIPAMT2', 'TMIN2', 'TMIN',
        'WT162', 'TMAX', 'PRECIPFLAG2', 'TMAX2',
        'WT16', 'WT182', 'PRCP2', 'NEXTDAYPRECIPAMT',
        'PRECIPAMT2', 'PRECIPFLAG', 'WT18', 'PRCP',
        'TAVG2', 'TAVG', 'PRECIPAMT']]
y = df['NEXTDAYPRECIPAMT3']

scaler = preproc.MinMaxScaler()
(x_train, x_test, y_train, y_test) = \
    model_selection.train_test_split(
        x, y, test_size=1/3, random_state=0)

scaler.fit(x_train)
scaler.fit(x_test)
x_train = scaler.transform(x_train)
x_test = scaler.transform(x_test)

print()
t1 = time.time()
reg = RidgeCV().fit(x_train, y_train)
print(time.time() - t1)

y_pred = reg.predict(x_test)

print("MSE Without Polynomial Test: ",
      metrics.mean_squared_error(y_test, y_pred))
print("R2 Without Polynomial Test Score: ",
      reg.score(x_test, y_test))

print("R2 Without Polynomial Training Score: ",
      reg.score(x_train, y_train))

poly = preproc.PolynomialFeatures(1)
bigTrainX = poly.fit_transform(x_train)
bigTestX = poly.fit_transform(x_test)
mlr = linmod.RidgeCV()
```

```
print()
t = time.time()
mlr.fit(bigTestX, y_test)
print("Test Fit Time", time.time() - t)


t = time.time()
mlr.fit(bigTrainX, y_train)
print("Training Fit Time", time.time() - t)


showStats(np.append(
    np.array(mlr.intercept_), mlr.coef_),
    bigTestX, y_test, mlr.predict(
        bigTestX))

showStats(np.append(
    np.array(mlr.intercept_), mlr.coef_),
    bigTrainX, y_train, mlr.predict(
        bigTrainX))
```