```python
# -*- coding: utf-8 -*-
"""
Created on Mon Mar 12 20:21:42 2018

@author: cml
"""

import numpy as np
from sklearn import linear_model
from sklearn.metrics import mean_squared_error, r2_score
import pandas as pd
from sklearn.model_selection import KFold # import KFold

df = pd.read_csv('Auto.csv', usecols=range(1,10))

X = df["horsepower"].values.reshape(-1,1) # our independent variable
y = df["mpg"].values.reshape(-1,1) # our dependent variable

kf = KFold(n_splits=10) # Define the split into 2 folds
print('Splits: ', kf.get_n_splits(X))

#Arrays to store test data and predictions for each run
ytests = []
ypreds = []

#for each KFold split in X, fit a model using current x_train
#and y_train value. Save the array
for train_index, test_index in kf.split(X):
    print("TRAIN: ", train_index, "TEST: ", test_index)
    X_train, X_test = X[train_index], X[test_index]
    y_train, y_test = y[train_index], y[test_index]

    model = linear_model.LinearRegression()
    model.fit(X = X_train, y = y_train)
    y_pred = model.predict(X_test)

    ytests += list(y_test)
    ypreds += list(y_pred)

rr = r2_score(ytests, ypreds)
ms_error = mean_squared_error(ytests, ypreds)

print("KFOLD results:")
print("R^2: {:.5f}%, MSE: {:.5f}".format(rr*100, ms_error))

#ms_error -> ~24 when n_splits increases.
#We use k-1 subsets to train our data.
#Large k means less bias towards overestimating the true expected error.

```