```python
# -*- coding: utf-8 -*-
"""
Created on Mon Mar 12 19:02:49 2018

@author: cml
"""

import matplotlib.pyplot as plt
import numpy as np
from sklearn import datasets, linear_model
from sklearn.metrics import mean_squared_error, r2_score
import numpy as np
import pandas as pd
import statsmodels.api as sm
import statsmodels.formula.api as smf
from patsy import dmatrices
from sklearn.model_selection import LeaveOneOut
from sklearn import metrics

# Courtesy: https://codeburst.io/cross-validation-calculating
#           -r%C2%B2-and-accuracy-scores-after-loocv-5bd1015a50ec

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

loo = LeaveOneOut()
print('Splits: ', loo.get_n_splits(X))

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

#for each LOOCV  split in X, fit a model using current x_train
#and y_train value. Save the array
for train_index, test_index in loo.split(X):
    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 = metrics.r2_score(ytests, ypreds)
ms_error = metrics.mean_squared_error(ytests, ypreds)

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