```python
# -*- coding: utf-8 -*-
"""
Created on Fri Mar  9 10:28:55 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

#get the data
df = pd.read_csv('Auto.csv', usecols=range(1,10))
#print(df.corr())

#initial slicing
train, test = np.split(df.sample(frac=1), [int(0.5*len(df))])

#split the data by formula
formula = 'mpg~horsepower'
y_train, x_train = dmatrices(formula, train, return_type='dataframe')
y_test, x_test = dmatrices(formula, test, return_type='dataframe')

lm = linear_model.LinearRegression()
lm.fit(x_train.iloc[:,1:3], y_train)

predictions = lm.predict(x_train.iloc[:,1:3])
print('MSE (linear): ', sum(predictions)/len(predictions))

#With poly fit 2
z = np.polyfit(x_train.loc[:,'horsepower'], y_train, 2)
print('MSE (poly2): ', sum(z)/len(z))

#With poly fit 3
z = np.polyfit(x_train.loc[:,'horsepower'], y_train, 3)
print('MSE (poly3): ', sum(z)/len(z))

#get the predictions for the trainoing set
#model = sm.OLS(y, X).fit()
#predictions = model.predict(X) # make the predictions by the model

```