Name: Chanyu Choung

Class: CMP414

Homework due date: Mar 15th, 2021 (Monday)

Week 6 Homework

This homework assignment will build three models on the advertising data and evaluate their performance. You can use tools from sklearn to complete this task.

Source of data: https://www.statlearning.com/s/Advertising.csv

1. Use train_test_split to split the data into training set (80%) and test set (20%).
2. Build a multilinear regression model with 'TV', 'Radio', and 'newspaper' as input variables and 'sales' as output variable. Name the model model_lr. Train the model on the training set and obtain model predictions on the test set.
3. Build a degree 2 polynomial regression model with 'TV', 'Radio', and 'newspaper' as input variables and 'sales' as output variable. Name the model model_pr2. Train the model on the training set and obtain model predictions on the test set.
4. Build a degree 10 polynomial regression model with 'TV', 'Radio', and 'newspaper' as input variables and 'sales' as output variable. Name the model model_pr10. Train the model on the training set and obtain model predictions on the test set.
5. Calculate the test MSE of each model using the mean_squared_error function. Which model gives the best MSE?

```
# Importing the libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline

# Importing the data frame
url = "https://www.statlearning.com/s/Advertising.csv"
data = pd.read_csv(url, index_col=0)


# Split the data into 80% training data and 20% test data.
from sklearn.model_selection import train_test_split
training_data, test_data = train_test_split(data, test_size=0.2)
test_data = test_data.copy()


# Initializing values
trainingX = training_data[["TV","radio","newspaper"]]
trainingY = training_data["sales"]
testX = test_data[["TV","radio","newspaper"]]
testY = test_data["sales"]


from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures

def get_Poly(n):
    poly_features = PolynomialFeatures(degree=n, include_bias=False)
    X_poly = poly_features.fit_transform(trainingX)
    return X_poly

def get_Predict(X1, X2, Y1, Name):
    # Train the model on training set
    model = LinearRegression()
    model.fit(X1, Y1)
    print(model.coef_[:3], model.intercept_)
```

```
        print(model.coef_[1:], model.intercept_)

        # Obtain the model predictions
        # test_data[Name] = model.predict(X2)
        # print(test_data.head(), "\n")
        return model

    # Multilinear Regression
    model_lr = get_Predict(trainingX, testX, trainingY, "MLR")
    # Polynomial-2 Regression
    poly2 = get_Poly(2)
    model_pr2 = get_Predict(poly2, polyT2, trainingY, "PR2")
    # Polynomial-10 Regression
    poly10 = get_Poly(10)
    model_pr10 = get_Predict(poly10, polyT10, trainingY, "PR10")
```

```
     [0.04647098 0.18500167 0.00167098] 2.81215769581941
     [0.05187328 0.01109873 0.00582227] 5.165260292326725
     [-9.76509031e-12 -2.99379511e-12 -6.65245105e-13] 6.6806009920099045
```

```
    # Calculate MSE of each model
    from sklearn.metrics import mean_squared_error
    LR = model_lr

    # MSE of Multilinear Regression
    theta = np.array([LR.intercept_, LR.coef_[0], LR.coef_[1], LR.coef_[2]])
    list_errors = []

    for i in data.index:
        x = np.array([1, data.loc[i, "TV"], data.loc[i, "radio"], data.loc[i, "newspaper"]])
        theta_dot_x = theta.dot(x)
        y = data.loc[i, "sales"]
        squared_error = (theta_dot_x - y) ** 2
        list_errors.append(squared_error)
    print("Multilinear Regression MSE:", np.mean(list_errors))

    # MSE of Polynomial-2 Regression
    predictions_pr2 = model_pr2.predict(poly2)
    mse_pr2 = mean_squared_error(trainingY, predictions_pr2)
    print("Polynomial Regression 2 MSE:", mse_pr2)

    # MSE of Polynomial-10 Regression
    predictions_pr10 = model_pr10.predict(poly10)
    mse_pr10 = mean_squared_error(trainingY, predictions_pr10)
    print("Polynomial Regression 10 MSE:", mse_pr10)
```

```
     Multilinear Regression MSE: 2.7923499899613478
     Polynomial Regression 2 MSE: 0.3774857219812148
     Polynomial Regression 10 MSE: 0.11184043112007987
```