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)
    X_polyT2 = poly_features.fit_transform(testX)
    return X_poly, X_polyT2

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_)

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

```
[ 0.04558287  0.18975776 -0.00061434] 2.8941799630329985
        TV  radio  newspaper  sales        MLR
186  205.0   45.1       19.6   22.6  20.784702
6      8.7   48.9       75.0    7.2  12.523830
102  296.4   36.3      100.9   23.8  23.231162
15   204.1   32.9       46.0   19.0  18.412414
178  170.2    7.8       35.2   11.7  12.110870

[0.053293  0.018441  0.0076701] 4.925784086694383
        TV  radio  newspaper  sales        MLR        PR2
186  205.0   45.1       19.6   22.6  20.784702  22.553006
6      8.7   48.9       75.0    7.2  12.523830   8.247367
102  296.4   36.3      100.9   23.8  23.231162  23.086957
15   204.1   32.9       46.0   19.0  18.412414  19.296373
178  170.2    7.8       35.2   11.7  12.110870  12.326893

[-4.07689229e-12 -1.28467403e-12 -9.45602856e-13] 6.391739584589704
        TV  radio  newspaper  sales        MLR        PR2           PR10
186  205.0   45.1       19.6   22.6  20.784702  22.553006   -1514.843805
6      8.7   48.9       75.0    7.2  12.523830   8.247367    5628.071880
102  296.4   36.3      100.9   23.8  23.231162  23.086957 -608165.059388
15   204.1   32.9       46.0   19.0  18.412414  19.296373     -26.571297
178  170.2    7.8       35.2   11.7  12.110870  12.326893       9.180745
```

```
# 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(polyT2)
mse_pr2 = mean_squared_error(testY, predictions_pr2)
print("Polynomial Regression 2 MSE:", mse_pr2)

# MSE of Polynomial-10 Regression
predictions_pr10 = model_pr10.predict(polyT10)
mse_pr10 = mean_squared_error(testY, predictions_pr10)
print("Polynomial Regression 10 MSE:", mse_pr10)
```

```
Multilinear Regression MSE: 2.7857517193194803
Polynomial Regression 2 MSE: 0.29345657721609386
Polynomial Regression 10 MSE: 9248605782.965168
```