Assignment #1 Multivariable Regression

F64126147 胡瑀真

注意事項 1: 書面報告可參考網路資料，但須理解與整理，自己理解與整理後撰寫報告，使用整段網路內容所獲的評分不高

注意事項 2: 勿分享報告給課程同學，相似內容的報告所獲評分不高

1. 說明是否參考或使用網路上程式，或程式自行撰寫?

(允許參考網路程式資源，在書面報告中須說明參考來源與參考程式的範圍。如程式與模型全部自做，亦在報告中說明，有額外分數)

Step5 中的函式 train_test_split()使用方法參考網路資料：CSDN 博客的內容，又 Step9 中的程式則參考生成式 AI 撰寫，其餘皆是參考講義內容自行撰寫。

2. 解析程式步驟 Step1-Step4

Step1

```
1 import numpy as np
2 import matplotlib
3 import matplotlib.pyplot as plt
4 from sklearn import linear_model
5 import pandas as pd
```

(1~5) 運用 import 匯入模組 numpy、matplotlib、matplotlib.pyplot、sklearn 的函式 linear_model、pamdas。

Step2

```
1 fn_src='https://raw.githubusercontent.com/sdrangan/introml/master/unit03_mult_lin_reg/Concrete_Data_Yeh.csv'
2 fn_dst='data.csv'
3
4 import os
5 from six.moves import urllib
6
7 if os.path.isfile(fn_dst):
8     print('File %s is already downloaded' % fn_dst)
9 else:
10     urllib.request.urlretrieve(fn_src, fn_dst)
11     print('File %s downloaded' % fn_dst)
```

(1~2) 將變數 fn_src 設為檔案網址，再將變數 fn_dst 設為檔案名稱。

(4~5) 運用 import 匯入模組 os、six.moves 的函式 urllib。

(7~8) 運用 if 敘述和 os.path.isfile(檔案名稱)判斷檔案是否存在資料夾內。若存在，則印出：File data.csv is already downloaded。

(9~11) 運用 else 敘述，執行檔案不存在資料夾內的情況：用 urllib.request.urlretrieve(檔案網址, 檔案名稱)將檔案下載，再印出 File data.csv downloaded。

```
1 df = pd.read_csv('data.csv')
```

(1) 宣告一變數 df，並運用 pandas 的函式 read_csv(檔案名稱)讀取 CSV 格式的檔案，將讀

取內容存入變數 df 中。

```
print(len(df))
df.head(20)
```

(1) 運用函式 len(變數)得到 df 的列數，並印出結果。
(2) 運用函式 head(20)得出 df 中，前 20 列的資料。

Step3

```
df.describe()
```

(1) 運用函式.describe()統計 df 資料的 count（此行的有效值數量）、mean（平均值）、std（標準差）、min（最小值）、25%（第一四分位數）、50%（第二四分位數）、75%（第三四分位數）、max（最大值）

```
names=df.columns.tolist()
print(names)
xnames=names[0:8]
print(xnames)
```

(1) 宣告一變數 names，並運用函式 columns 取得 df 的欄位名稱，再用 tolist()將欄位名稱轉換為列表。
(2) 印出 names。
(3) 根據講義可知自變數是前八欄，依變數是最後一欄，因此使用 names[0:8]將前八欄取出，並存入變數 xnames 中。
(4) 印出 xnames。

Step4

```
X = df.iloc[:,:-1]
y = df.iloc[:,-1]

print(X.shape)
print(y.shape)
X
```

(1) 宣告一變數 X，並運用 iloc[:, : -1]將 df 的資料分割，其中[:, : -1]前一項表示選取所有列，後一項表示選取除最後一行以外的資料。
(2) 宣告一變數 y，類似步驟(1) 但選取的資料是最後一行的每一列。
(4~5) 印出 X 和 y 的尺寸。
(6) 將變數 X 的內容印出。

3. 所完成的程式步驟 Step5-Step9

Step5

```
1 from sklearn.model_selection import train_test_split
2
3 # TODO
4 #  Xtr,Xts,ytr,yts = train_test_split(...)
5
6 Xtr, Xts, ytr, yts = train_test_split(X, y, test_size=0.3, random_state=42)
7
8 print("Xtr shape:", Xtr.shape)
9 print("ytr shape:", ytr.shape)
10 print("Xts shape:", Xts.shape)
11 print("yts shape:", yts.shape)
```

(1) 運用 import 匯入模組 sklearn.model_selection 的函式 train_test_split。

(6) 運用函式 train_test_split()分割資料，其中 test_size=0.3 表示將 30%的資料存於 Xts、yts，random_state=42 使每次分割結果相同。此函式可隨機分割資料，與講義中的分割方式 Xtr = [:ns_train,:]不同。

(8~11) 分別印出 Xtr、ytr、Xts、yts 的尺寸。

Step6

```
1 # TODO
2 #   reg = ...
3 #   reg.fit(...)
4
5 reg = linear_model.LinearRegression()
6 reg.fit(Xtr, ytr)
```

(5) 運用函式 linear_model.LinearRegression()創造一個 LinearRegression 物件，並存入 reg 中。

(6) 使用 fit()和分割好的訓練資料 Xtr、ytr 訓練模型。

Step7

```
1 # TODO
2 #    yhat_tr = ...
3 #    rsq_tr = ...
4 yhat_tr = reg.predict(Xtr)
5 RSS_tr = np.mean((yhat_tr-ytr)**2/(np.std(ytr))**2)
6 rsq_tr = 1-RSS_tr
7
8 print("R^2 value on the training data:", rsq_tr)
```

(4) 運用 predict()將訓練好的模型以 Xtr 為資料做預測，並將結果存於 yhat_tr 中。

(5~6) 仿效講義的程式，使用 R^2 的公式，算出 R^2 值，並將結果存於 rsq_tr 中。

(8) 印出 R^2。

Step8

```
1 # TODO
2 #    yhat_val = ...
3 #    rsq_val = ...
4
5 yhat_val = reg.predict(Xts)
6 RSS_ts = np.mean((yhat_val-yts)**2/(np.std(yts))**2)
7 rsq_val = 1-RSS_ts
8
9 print("R^2 value on the validation data.", rsq_val)
```

(5) 運用 predict()將訓練好的模型以 Xts 為資料做預測，並將結果存於 yhat_val 中。

(6~7) 仿效講義的程式，使用 R^2 的公式，算出 R^2 值，並將結果存於 rsq_val 中。

(9) 印出 R^2。

Step9

```
# TODO
plt.scatter(yts, yhat_val, alpha=0.5)
plt.xlabel("Actual y values")
plt.ylabel("Predicted y values")
plt.title("Actual vs. Predicted Values on Validation Data")
plt.plot([min(yts), max(yts)], [min(yts), max(yts)], color="red", linestyle="--")
plt.show()
```

(2) 使用函式 plt.scatter(X 軸, Y 軸, 透明度)繪製散布圖。

(3~5) 設定圖表、X 軸和 Y 軸的標題。

(6) 使用函式 plt.plot(X 軸範圍, Y 軸範圍, 線條顏色, 設定線條種類)，繪製完美預測的參考線。

(7) 顯示繪製的圖表。

4. 「水泥構件抗壓強度」資料線性迴歸(linear regression)結果分析與結論?

從決定係數 R^2 在 training data 是 0.6196723710532992，在 validation data 是 0.5943782479239206 可知，依變數的變異中，可由自變數解釋的比例約是 60%，故本線性迴歸模型在預測水泥構建抗壓強度上，表現雖然不差，但也沒有很好。另外，training data 的 R^2 和 validation data 的 R^2 接近，表示本模型沒有 overfitting（過擬合，模型在 training data 有高 R^2 但在 validation data 有低 R^2）的情形。

推測模型的表現不優，是因為依變數和自變數間有非線性的關係，若使用 Multinomial model、polynomial model 或 exponential model 可能會使模型的 R^2 提高。

5. 以此作業為例，描述對 multivariable linear regression 的理解

multivariable linear regression 是一種有多個自變數的線性迴歸方式，線性模型為$\hat{y} = \beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k$。在此作業中 k 為 8，自變數 x 是資料的前 8 欄（水泥、高爐渣、飛灰等），而依變數 y 則為資料的最後一欄（抗壓強度）。

作業中，首先運用 sklearn.model_selection 的函式 train_test_split 將資料隨機拆分，其中 70%用於訓練，30%用於驗證。之後，再用函式 linear_model.LinearRegression()、fit()和 70%的資料訓練模型，得出β_0至β_8的數值，並用 predict()將訓練好的模型分別對 70%和 30%的資料做預測，得到$\hat{y}$。最後，運用 R^2 的公式，求得此線性迴歸模型應用於資料的適配度。

6. Supervised learning 與 un-supervised learning 有何不同? multivariable linear regression 屬於前述那一種 learning? 為什麼?

兩者最大的差異在於訓練使用的資料不同。supervised learning（監督式學習），會使用被標記過、有已知依變數的資料做訓練；而 un-supervised learning（非監督式學習）則使用沒有被標記過、無已知依變數的資料做訓練，讓模型自行尋找資料中的機制。

multivariable linear regression 屬於 supervised learning（監督式學習），因為訓練使用的

資料是被標記過的，有具體已知的自變數(水泥、高爐渣、飛灰等)和依變數(抗壓強度)。

參考資料 (書面報告的參考資料)

1. 上課講義第一章
2. http://www.linearalgebra.com