作業3 Linear Regression

NCU MIS 106403551 呂晟維

用 Weka 軟體對 BreastCancerNUM.arff, BreastCancerNOM.arff 分別建立 Linear Regression model 及 Logistic model 選擇 Use training set, 設定 Attribute: type 為 diagnosis ,在過程中對重要步驟截圖加以說明並回答以下問題：

- (a) LinearRegression model 中多少因素與乳癌為正相關?多少為負相關? 10%
- (b) 舉出三個 LinearRegression model 認為與乳癌無關的因素? 10%
- (c) 比較 Logistic model 與 Linear Regression model 的均方根誤差兩者差距多少哪個模型表現較好? 30%

用 python 對 BreastCancer.csv 建立 LinearRegression model 及 Logistic model(solver='liblinear') 在過程中對重要步驟截圖加以說明並回答以下問題：

- (a) 以 coef_function 印出各 Attribute 的相關係數請問 LinearRegression model 中 Attribute 'area_mean' 的係數是多少? 10%
- (b) LinearRegression model 中影響最大的因素為何? 20%
- (c) 以 score() function 印出兩模型的正確率，請問何者較高? 20%

Weka部分

本次的乳癌資料有31筆attr，有30筆當作Input(都是數值資料)，欄位diagnosis是Output，Instance總共有569列。

- BreastCancerNUM.arff的Output為numeric資料，只能run Linear Regression model
- BreastCancerNOM.arff的Output是nominal，只能run Logistic model

(a) LinearRegression model 中多少因素與乳癌為正相關?多少為負相關? 10%

首先我們用 `BreastCancerNUM.arff` 來當數據集，執行得出的回歸模型如下，，一共會有30個權重w_1~w_{30}，和一個常數項w_0；由結果得知有18個非0權重，正相關的有6個(weight>0)，負相關的有12個(weight<0)。

```
=== Classifier model (full training set) ===
Linear Regression Model

diagnosis =

     -0.0709 * radius_mean +
     -0.0047 * perimeter_mean +
      0.0008 * area_mean +
      3.8735 * compactness_mean +
     -1.1601 * concavity_mean +
     -1.9936 * concave points_mean +
     -0.969  * radius_se +
      0.0363 * perimeter_se +
      0.0029 * area_se +
    -19.4375 * smoothness_se +
      3.0168 * concavity_se +
     -0.0105 * texture_worst +
     -0.0058 * perimeter_worst +
      0.0002 * area_worst +
     -0.3631 * concavity_worst +
     -1.8387 * concave points_worst +
     -0.811  * symmetry_worst +
     -3.909  * fractal_dimension_worst +
      3.133
```

正相關的6個因素：

```
      0.0008 * area_mean
      3.8735 * compactness_mean
      0.0363 * perimeter_se
      0.0029 * area_se
      0.0002 * area_worst
      3.0168 * concavity_se
```

負相關的12個因素：

```
     -0.0709 * radius_mean
     -0.0047 * perimeter_mean
     -1.1601 * concavity_mean
     -1.9936 * concave points_mean
     -0.969  * radius_se
    -19.4375 * smoothness_se
     -0.0105 * texture_worst
     -0.0058 * perimeter_worst
     -0.3631 * concavity_worst
     -1.8387 * concave points_worst
     -0.811  * symmetry_worst
     -3.909  * fractal_dimension_worst
```

(b) 舉出三個 LinearRegression model 認為與乳癌無關的因素? 10%

與乳癌無關的因素有$30 - 18 = 12$個，全部列出來如下：

```
texture_mean
smoothness_mean
symmetry_mean
fractal_dimension_mean
texture_se
compactness_se
concave points_se
symmetry_se
fractal_dimension_se
radius_worst
smoothness_worst
compactness_worst
```

(c) 比較 Logistic model 與 Linear Regression model 的均方根誤差兩者差距多少 哪個模型表現較好? 30%

兩者的均方根誤差相差蠻多的，其實均方根誤差就是testing data誤差的標準差；Linear的均方根誤差為 `0.2331` ，Logistic的均方根誤差為 `0.0003` 。

從數字上看起來，Linear的表現較差，但是其實很難從數值判斷，因為Llinear算出來的數值不是機率，只是用回歸來估計的數值結果，predict值會為負或大於1；如果是predict = 0.9 還是不等於 1。所以會有誤差 = 0.1。

而Logistic是分類法，輸出是non-linear的機率，所以predict機率 = 0.9即意會有乳癌，預測正確。因此判斷沒有誤差。若細看Confusion Matrix的話，發現Logistic的testing根本沒有任何錯誤，因為java浮點數運算問題才會得到 $\sqrt{0.0} = 0.0003$。

Linear Regression model的summary如下：

```
Correlation coefficient                  0.8761
Mean absolute error                      0.1841
Root mean squared error                  0.2331  <== '均方根誤差'
Relative absolute error                 39.3731 %
Root relative squared error             48.2196 %
Total Number of Instances              569
```

Logistic model的summary如下：

```
Correctly Classified Instances         569               100      %
Incorrectly Classified Instances         0                 0      %
Kappa statistic                          1
Mean absolute error                      0
Root mean squared error                  0.0003  <== '均方根誤差'
Relative absolute error                  0.0085 %
Root relative squared error              0.0538 %
Total Number of Instances              569

=== Confusion Matrix ===

   a   b   <-- classified as
 212   0 |   a = M
   0 357 |   b = B
```

Pyhton部分

程式碼解說請至 `Linear Regression - Breast Cancer.ipynb` 以及 `Logistic Regression - Breast Cancer.ipynb` 查看唷。兩者大同小異，都是由sklearn官網範例更改來的。作業最後面也有稍做講解。

(a) 以 coef_function 印出各 Attribute 的相關係數請問 LinearRegression model 中 Attribute 'area_mean' 的係數是多少? 10%

來看看Linear Regression model 的輸出，已將科學記號的結果轉成浮點數:

```
Coefficients:
 [  0.2178  -0.0045  -0.0237  -0.0003  -0.0847   4.222   -1.398   -2.1418
  -0.1027  -0.0333  -0.435    0.0068   0.0225   0.0009 -15.8543  -0.0649
   3.5655 -10.568   -1.6973   7.1464  -0.1952  -0.0072   0.0024   0.001
  -0.5429  -0.0672  -0.3812  -0.4643  -0.5568  -4.3035]
Intercept:  3.021811738437393
Mean squared error: 0.05
Coefficient of determination: 0.77
```

Coefficients列出了30個屬性的權重各是多少，順序是照著csv欄位排序，所以 'area_mean' 的係數是第四個input 也就是 `-0.0003` ，其實相關度是非常低的負相關。

(b) LinearRegression model 中影響最大的因素為何? 20%

由a小題的輸出結果可以發現Coefficient數值取絕對值後最大的是 `-15.8543` ，他是第十五個係數，也就是csv資料欄位的第十六欄 `smoothness_se` 。

(c) 以 score() function 印出兩模型的正確率，請問何者較高? 20%

`Coefficient of determination` 就是 `score()` function。代表模型的正確率，是0~1的數值。

- Linear Regression model的score是 `0.77`
- Logistic Regression model的score是 `0.96`

很明顯還是Logistic的精準度高，但若同樣考量到classification的原理(weka部分講過)，精準度高也是很合理的。順便看一下Logistic Regression model的Confusion Matrix，正確率也是很高。

```
array([[198,  14],
       [  9, 348]], dtype=int64)
```

Appendix 程式解說

As same as previous HWs, it requires 3 steps. First step is pre-process the data into x(input), y(output).

In [ ]:

```
# Load the BreastCancer dataset
df =  pd.read_csv('BreastCancer.csv')
# Label of data > y
breastCancer_y = df['diagnosis'].values

# Training data > X
breastCancer_X = df.iloc[:,1:]
breastCancer_X = breastCancer_X.values
```

Then we train the LR model & do predictions.

- linear regression model:

In [ ]:

```
# Create linear regression object
regr = linear_model.LinearRegression()

# Train the model using the training sets
regr.fit(breastCancer_X, breastCancer_y)

# Make predictions using the testing set = training sets
breastCancer_y_pred = regr.predict(breastCancer_X)
```

- logistic regression model:

In [ ]:

```
# Create linear regression object
logreg = linear_model.LogisticRegression(solver = 'liblinear')

# Train the model using the training sets
logreg.fit(breastCancer_X, breastCancer_y)

# Make predictions using the testing set = training sets
breastCancer_y_pred = logreg.predict(breastCancer_X)
```

Finally, ends with printing the results.

In [ ]:

```
# The coefficients 各項的係數/權重，有30個
print(regr.coef_)
# The intercept 截距
print(regr.intercept_)
# The mean squared error
print(mean_squared_error(breastCancer_y, breastCancer_y_pred))
# The coefficient of determination: 1 is perfect prediction 沒測試集so代原本訓練集
print(regr.score(breastCancer_X, breastCancer_y))
```