ML_Lab_LogisticRegression

June 21, 2021

0.1 Online Lab 1 : 21.June.2021

0.1.1 Logistic Regression

0.1.2 Dr Neeraj Gupta

[5]:
```
import pandas as pd
pima = pd.read_csv("image\datasets_228_482_diabetes.csv",␣
 ↪header=None,names=col_names)
col_names = ['pregnant', 'glucose', 'bp', 'skin', 'insulin', 'bmi',␣
 ↪'pedigree','age', 'label']
pima.head()
```

[5]:

	pregnant	glucose	bp	skin	insulin	bmi	pedigree	age	label
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

[8]:
```
feature_cols = ['pregnant', 'insulin', 'bmi', 'age','glucose','bp','pedigree']
X = pima[feature_cols] #Features
y = pima.label #Target variable
```

[12]:
```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.
 ↪25,random_state=0)
print(X_train)
print(X_test)
```

```
     pregnant  insulin   bmi  age  glucose  bp  pedigree
762         9        0  22.5   33       89  62     0.142
127         1       94  33.3   23      118  58     0.261
564         0        0  32.4   27       91  80     0.601
375        12      325  39.2   58      140  82     0.528
663         9      130  37.9   40      145  80     0.637
..        ...      ...   ...  ...      ...  ..       ...
763        10      180  32.9   63      101  76     0.171
192         7        0  30.4   36      159  66     0.383
```

```
629         4        0  24.7   21       94  65     0.148
559        11        0  30.1   35       85  74     0.300
684         5        0   0.0   69      136  82     0.640

[576 rows x 7 columns]
     pregnant  insulin   bmi  age  glucose  bp  pedigree
661         1        0  42.9   22      199  76     1.394
122         2      100  33.6   23      107  74     0.404
113         4        0  34.0   25       76  62     0.391
14          5      175  25.8   51      166  72     0.587
529         0        0  24.6   31      111  65     0.660
..        …      …   …  …      …   ..       …
366         6        0  27.6   29      124  72     0.368
301         2      135  31.6   25      144  58     0.422
382         1      182  25.4   21      109  60     0.947
140         3        0  21.1   55      128  78     0.268
463         5        0  27.6   37       88  78     0.258

[192 rows x 7 columns]
```

[13]:
```
#import the class
from sklearn.linear_model import LogisticRegression

#instantiate the model (create the object of class LogisticRegression)
logreg = LogisticRegression()

#fit the model with data
logreg.fit(X_train, y_train) #%75 data for training

y_pred = logreg.predict(X_test) #%25 data for test

print(y_test.shape)
```

```
(192,)

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:432:
FutureWarning: Default solver will be changed to 'lbfgs' in 0.22. Specify a
solver to silence this warning.
  FutureWarning)
```

[14]:
```
y_test.values
```

[14]:
```
array([1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1,
       1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1,
       1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
       1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1,
       0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
```

```
       0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0,
       1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0,
       0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0], dtype=int64)
```

[15]:
```
y_pred
```

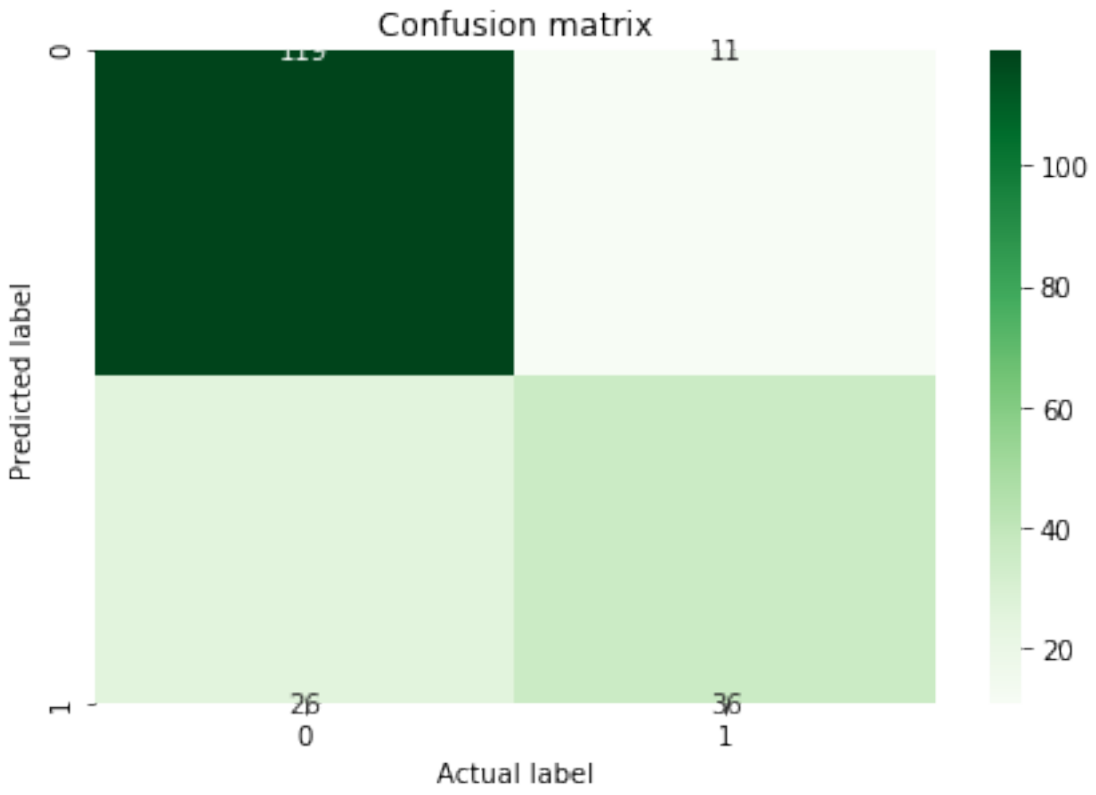
[15]:
```
array([1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0,
       0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1,
       1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1,
       1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1,
       0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
       1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0], dtype=int64)
```

[16]:
```
#import the metrics class
from sklearn import metrics
cnf_matrix = metrics.confusion_matrix(y_test, y_pred)
cnf_matrix
```

[16]:
```
array([[119,  11],
       [ 26,  36]], dtype=int64)
```

[19]:
```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
class_names=[0,1]
fig, ax = plt.subplots()
tick_marks = np.arange(len(class_names))
plt.xticks(tick_marks, class_names)
plt.yticks(tick_marks, class_names)
print
# create heatmap
sns.heatmap(pd.DataFrame(cnf_matrix), annot=True,␣
 ↪cmap="Greens",fmt='d',annot_kws={"size": 10})
ax.xaxis.set_label_position('bottom')
plt.tight_layout()
plt.title('Confusion matrix', y=1.4)
plt.xlabel('Actual label')
plt.ylabel('Predicted label')
```

[19]:
```
Text(33.0, 0.5, 'Predicted label')
```



[20]:
```
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
print("Precision:",metrics.precision_score(y_test, y_pred))
print("Recall:",metrics.recall_score(y_test, y_pred))
print("F1-score", metrics.f1_score(y_test, y_pred))
```

```
Accuracy: 0.8072916666666666
Precision: 0.7659574468085106
Recall: 0.5806451612903226
F1-score 0.6605504587155964
```