BÀI TUẦN 11: PRINCIPAL COMPONENT ANALYSIS

1. Thông tin sinh viên

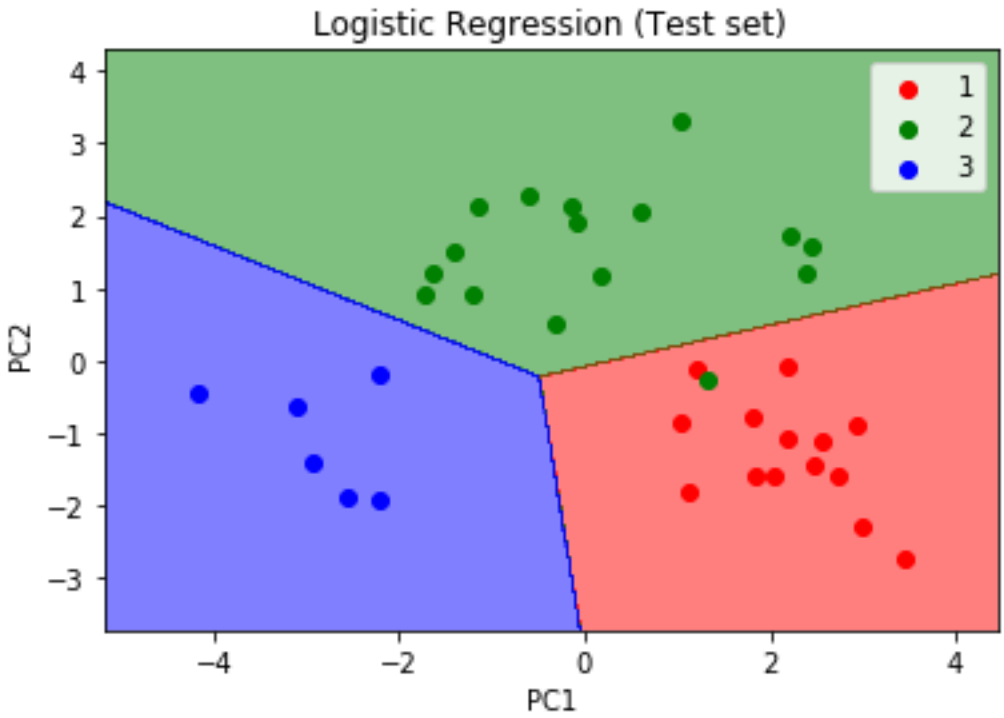
- **Họ và Tên: Dương Minh Lượng**
- **MSSV: 18521071**

2. Source

```
1. import numpy as np
2. import matplotlib.pyplot as plt
3. import pandas as pd
4. from sklearn.metrics import confusion_matrix
5. # Importing the dataset
6. dataset = pd.read_csv('Wine.csv')
7. X = dataset.iloc[:, :-1].values
8. y = dataset.iloc[:, -1].values
9.
10.     from sklearn.model_selection import
   train_test_split
11.     X_train, X_test, y_train, y_test =
   train_test_split(X, y, test_size = 0.2, random_state
   = 0)
12.     from sklearn.preprocessing import
   StandardScaler
13.     SC = StandardScaler()
14.     X_train_sc = SC.fit_transform(X_train)
15.     X_test_sc = SC.transform(X_test)
16.     from sklearn.decomposition import PCA
17.     pca = PCA(n_components= 2)
18.     X_train_sc_pca = pca.fit_transform(X_train_sc)
19.     X_test_sc_pca = pca.transform(X_test_sc)
20.     information_gain_ratio =
   pca.explained_variance_ratio_
21.     from matplotlib.colors import ListedColormap
22.     def VisualizingDataset(X_, Y_):
23.         X1 = X_[:, 0]
24.         X2 = X_[:, 1]
25.         for i, label in enumerate(np.unique(Y_)):
26.          plt.scatter(X1[Y_ == label], X2[Y_ ==
   label],color = ListedColormap(("red", "green",
   "blue"))(i),label = label)
27.         plt.legend()
```

```
28.      def VisualizingResult(model, X_):
29.          X1 = X_[:, 0]
30.          X2 = X_[:, 1]
31.          X1_range = np.arange(start= X1.min()-1,
   stop= X1.max()+1,step = 0.01)
32.          X2_range = np.arange(start= X2.min()-1,
   stop= X2.max()+1,step = 0.01)
33.          X1_matrix, X2_matrix =
   np.meshgrid(X1_range, X2_range)
34.          X_grid=
   np.array([X1_matrix.ravel(),X2_matrix.ravel()]).T
35.          Y_grid=
   model.predict(X_grid).reshape(X1_matrix.shape)
36.          plt.contourf(X1_matrix, X2_matrix, Y_grid,
   alpha = 0.5,cmap = ListedColormap(("red","green",
   "blue")))
37.      from sklearn.linear_model import
   LogisticRegression
38.      log_reg = LogisticRegression(random_state= 0)
39.      log_reg.fit(X_train, y_train)
40.      pca_log_reg = LogisticRegression(random_state =
   0)
41.      pca_log_reg.fit(X_train_sc_pca, y_train)
42.      cm1 = confusion_matrix(y_train,
   log_reg.predict(X_train))
43.      print(cm1)
44.      from sklearn.metrics import confusion_matrix
45.      cm2 = confusion_matrix(y_train,
   pca_log_reg.predict(X_train_sc_pca))
46.      print(cm2)
47.      VisualizingResult(pca_log_reg, X_train_sc_pca)
48.      VisualizingDataset(X_train_sc_pca, y_train)
49.      plt.title('Logistic Regression (Training set)')
50.      plt.xlabel('PC1')
51.      plt.ylabel('PC2')
52.      plt.show()
53.      cm3 = confusion_matrix(y_test,
   log_reg.predict(X_test))
54.      print(cm3)
55.      cm = confusion_matrix(y_test,
   pca_log_reg.predict(X_test_sc_pca))
56.      print(cm)
57.
58.      VisualizingResult(pca_log_reg, X_test_sc_pca)
59.      VisualizingDataset(X_test_sc_pca, y_test)
60.      plt.title('Logistic Regression (Test set)')
61.      plt.xlabel('PC1')
62.      plt.ylabel('PC2')
63.      plt.show()
```

3. Kết quả



Confusion_matrix test tập dữ liệu test chưa giảm chiều

	0	1	2
0	13	1	0
1	1	14	1
2	0	0	6

Confusion_matrix test tập dữ liệu test đã giảm chiều

	0	1	2
0	14	0	0
1	1	15	0
2	0	0	6

Nhận xét:

- Với kết quả dữ liệu test chưa giảm chiều thì:
    - Tổng số điểm dữ liệu dự đoán đúng là 13 + 14 + 6 = 33.
    - Tổng số điểm dữ liệu dự đoán sai là 3.
    - Tỉ lệ dự đoán sai là 3/36 ≈ 0.08(3) ≈ 8.33%.
- Với kết quả dữ liệu test đã giảm chiều thì:
    - Tổng số điểm dữ liệu dự đoán đúng là 14 + 15 + 6 = 35.
    - Tổng số điểm dữ liệu dự đoán sai là 1.
    - Tỉ lệ dự đoán sai là 1/36 ≈ 0.02(7) ≈ 2.78%.
- Tỉ lệ dự đoán sai của 2 mô hình tập test.

	Chưa giảm chiều	Đã giảm chiều
Test	8.33%	2.78%