Университет ИТМО
Кафедра ИПМ

Машинное обучение
Лабораторная работа 3
«Методы дискриминантного анализа»

Выполнил:
Шаймарданов Руслан
группа P4117
Преподаватель:
Жукова Н. А.

Санкт-Петербург
2017

Выбранный датасет: «Statlog (Shuttle) Data Set»

http://archive.ics.uci.edu/ml/datasets/Statlog+%28Shuttle%29

Количество записей: 14500

Описание:

The shuttle dataset contains 9 attributes all of which are numerical. The first one being time. The last column is the class which has been coded as follows :

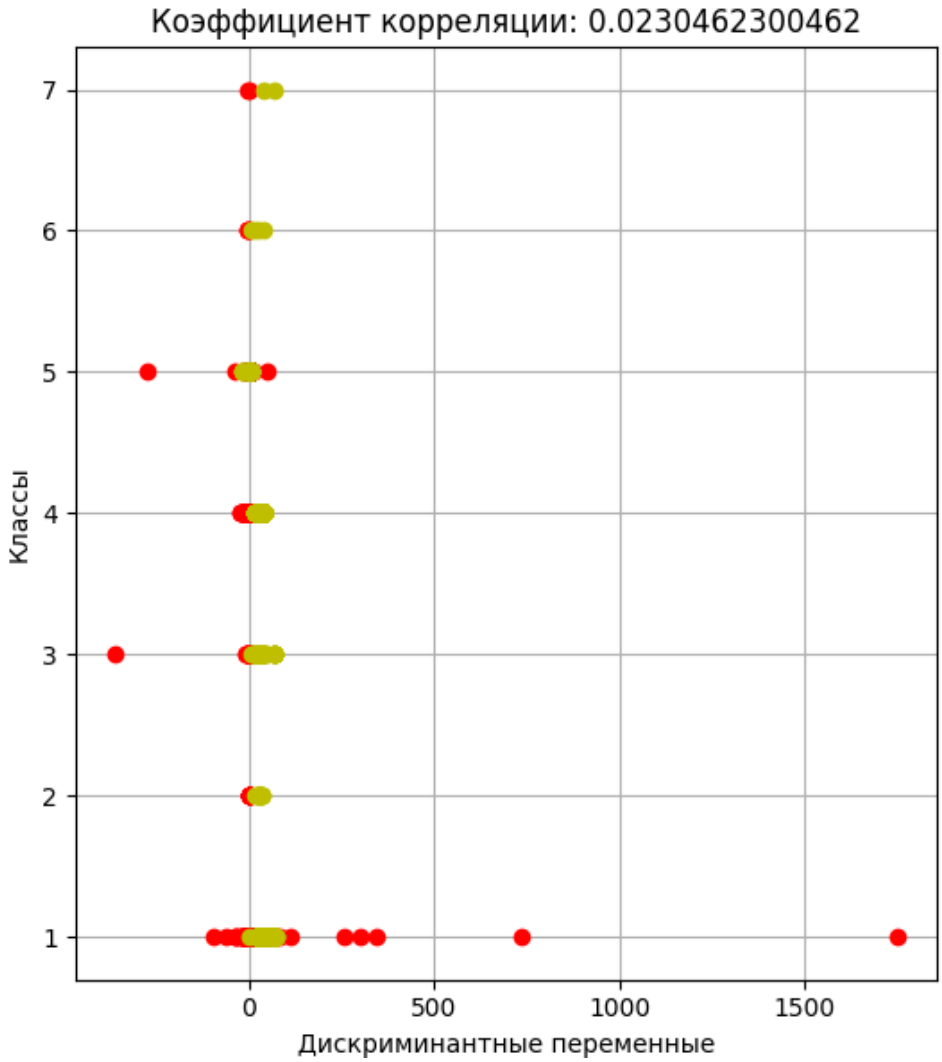
1 Rad Flow
2 Fpv Close
3 Fpv Open
4 High
5 Bypass
6 Bpv Close
7 Bpv Open

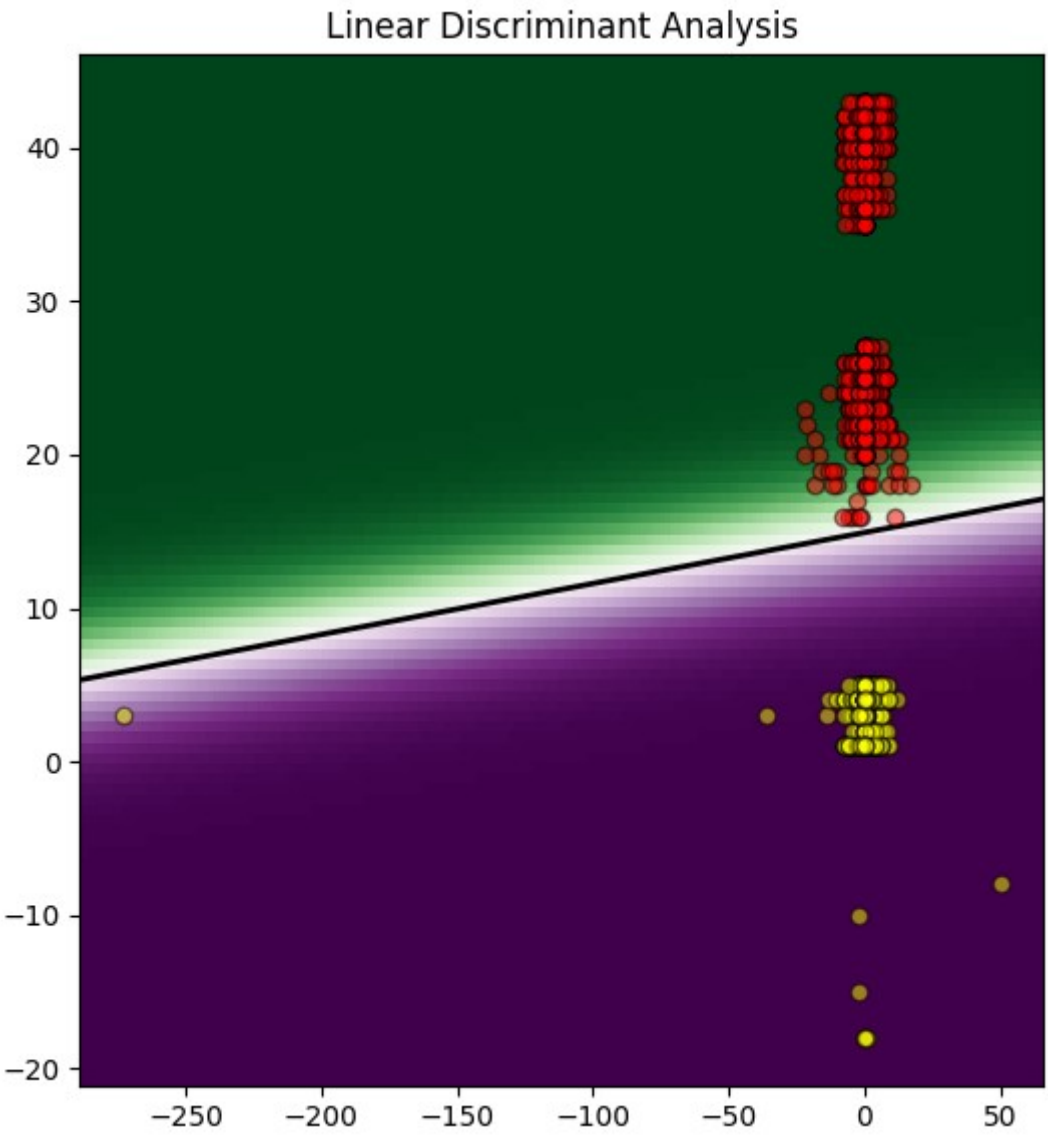
Алгоритм lab3.py

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import colors
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis as LDA
from sklearn.discriminant_analysis import QuadraticDiscriminantAnalysis as QDA

def visualization(x1, x2, y):
    splot = plt.subplot(1, 2, 1)
    plt.plot(x1, y, 'ro', x2, y, 'yo')
    plt.xlabel(r'Дискриминантные переменные')
    plt.ylabel(r'Классы')
    plt.title(r'Коэффициент корреляции: ' + str(np.corrcoef(x1, x2)[0, 1]))
    plt.grid(True)  # Сетка
    return splot

def plot_data(lda, X, y, y_pred):
    splot = plt.subplot(1, 2, 2)
    plt.title('Linear Discriminant Analysis')

    tp = (y == y_pred)  # True Positive
    tp0, tp1 = tp[y == 0], tp[y == 1]
    X0, X1 = X[y == 0], X[y == 1]
    X0_tp, X0_fp = X0[tp0], X0[~tp0]
    X1_tp, X1_fp = X1[tp1], X1[~tp1]
    alpha = 0.5

    plt.plot(X0_tp[:, 0], X0_tp[:, 1], 'o', alpha=alpha, color='red', markeredgecolor='k')
    plt.plot(X0_fp[:, 0], X0_fp[:, 1], '*', alpha=alpha, color='red', markeredgecolor='k')
    plt.plot(X1_tp[:, 0], X1_tp[:, 1], 'o', alpha=alpha, color='yellow', markeredgecolor='k')
    plt.plot(X1_fp[:, 0], X1_fp[:, 1], '*', alpha=alpha, color='yellow', markeredgecolor='k')

    #areas
    nx, ny = 200, 100
    x_min, x_max = plt.xlim()
    y_min, y_max = plt.ylim()
    xx, yy = np.meshgrid(np.linspace(x_min, x_max, nx), np.linspace(y_min, y_max, ny))
    Z = lda.predict_proba(np.c_[xx.ravel(), yy.ravel()])
    Z = Z[:, 1].reshape(xx.shape)
    plt.pcolormesh(xx, yy, Z, cmap='PRGn_r', norm=colors.Normalize(0., 1.))
    plt.contour(xx, yy, Z, [0.5], linewidths=2., colors='k')
    return splot
```

```
def main():
    dataset = pd.read_csv("shuttle.csv", header=None).values.astype(np.int32, copy=False)
    data_train = dataset[0:int(len(dataset) * 0.6)]
    data_test = dataset[int(len(dataset) * 0.6) + 1:]
    x, y = np.array([]), np.array([])
    for row in dataset:
        if (row[-1] == 4 or row[-1] == 5):
            x = np.vstack((x, [row[3], row[6]])) if len(x) != 0 else [row[3], row[6]]
            y=np.append(y, row[-1]-4)
   #<class 'list'>: [11478, 13, 39, 2155, 809, 4, 2] =>  4, 5

    lda = LDA(solver="svd", store_covariance=True)
    splot = visualization(dataset[:, 3], dataset[:, 6], dataset[:, -1])
    splot = plot_data(lda, x, y, lda.fit(x, y).predict(x))
    plt.axis('tight')
    plt.show()

    lda = lda.fit(data_train[:, :-1], data_train[:, -1])
    lda = lda.score(data_test[:, :-1], data_test[:, -1])
    qda = QDA(store_covariances=True)
    qda = qda.fit(data_train[:, :-1], data_train[:, -1])
    qda = qda.score(data_test[:, :-1], data_test[:, -1])

    print("Linear Discriminant Analysis: ", lda)
    print("Quadratic Discriminant Analysis: ", qda)

main()
```

Вывод программы





Linear Discriminant Analysis: 0.950508708398

Quadratic Discriminant Analysis: 0.946370063804

Вывод

В ходе работы выполнена визуализация двух признаков, имеющих малую корреляцию – 0.023.

Сделано разбиение классов набора данных с помощью Linear Discriminant Analysis. Для наглядности визуализация осуществлена для двух классов (иначе возникала неясная и неинформативная мешанина).

Классификация данных методами LDA и QDA показали высокую точность, LDA лидирует с небольшим отрывом.