Университет ИТМО

Кафедра информатики и прикладной математики

Машинное обучение

Лабораторная работа №3

Методы дискриминантного анализа

Выполнили: Иппо Вера, группа P4117

Преподаватель:

Санкт-Петербург
2017

1. Цель работы: получить практические навыки работы с методом дискриминантного анализа и визуализацией данных на практических примерах с использованием языка программирования python.

2. Исходные данные

Датасет: https://archive.ics.uci.edu/ml/datasets/Letter+Recognition

Предметная область: буквы латинского алфавиита

Задача: определить, какой из букв латинского алфавита соответствует набор характеристик ее написания.

Количество записей: 20000

Количество атрибутов: 16

Атрибуты:

1. lettr capital letter (26 values from A to Z)
2. x-box horizontal position of box (integer)
3. y-box vertical position of box (integer)
4. width width of box (integer)
5. high height of box (integer)
6. onpix total # on pixels (integer)
7. x-bar mean x of on pixels in box (integer)
8. y-bar mean y of on pixels in box (integer)
9. x2bar mean x variance (integer)
10. y2bar mean y variance (integer)
11. xybar mean x y correlation (integer)
12. x2ybr mean of x * x * y (integer)
13. xy2br mean of x * y * y (integer)
14. x-ege mean edge count left to right (integer)
15. xegvy correlation of x-ege with y (integer)
16. y-ege mean edge count bottom to top (integer)
17. yegvx correlation of y-ege with x (integer)

3. Ход работы

Код программы:

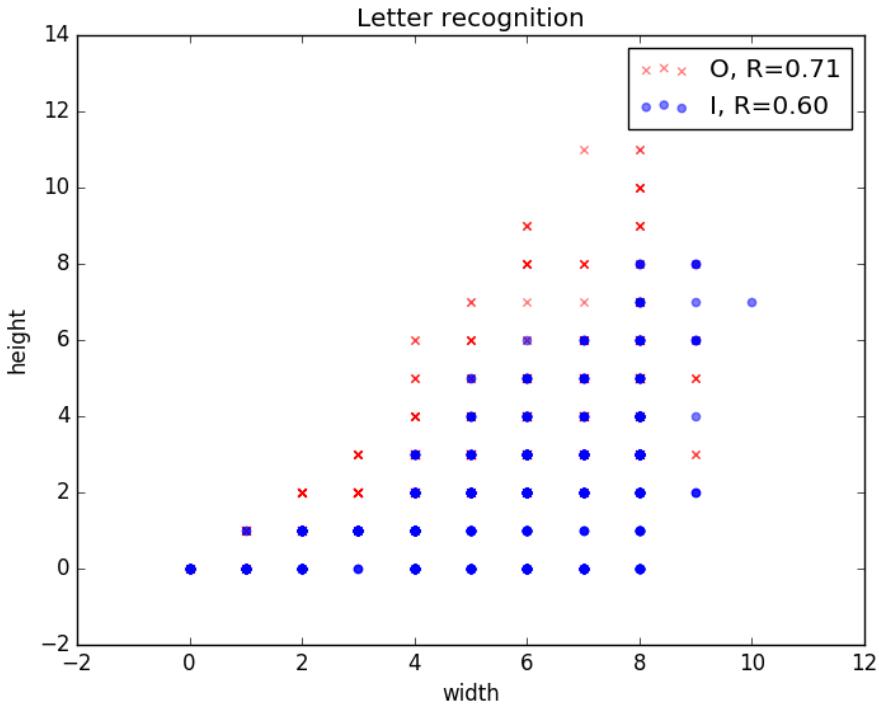
```
import pandas
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
from scipy.stats import pearsonr
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.discriminant_analysis import QuadraticDiscriminantAnalysis
#загрузить датасет
def load_dataset(filename):
    csv_dataset = pandas.read_csv(filename, header=None).values
    dataset=csv_dataset
    letter_attr = dataset[:,1:] # список атрибутов (признаков) для каждой буквы
    letter_class = dataset[:,0] # классы букв
    return letter_attr,letter_class
#разделить на обучающую и тестовую выборки
def split_dataset(letter_attr,letter_class,test_size):
    data_train, data_test, class_train, class_test = train_test_split(letter_attr, letter_class, test_size=test_size)
    return data_train, class_train, data_test, class_test
#визуализация ширины и высоты и расчет корреляции их значения
def visualize_data(letter_attr,letter_class,alphabet):
    count = int(len(letter_attr))
    for letter,marker,color in zip(('O', 'I'), ('x', 'o'), ('red', 'blue')):
    a1 = [letter_attr[i][3] for i in range(count) if letter_class[i] == letter]
    a2 = [letter_attr[i][4] for i in range(count) if letter_class[i] == letter]
    R = pearsonr(a1, a2)
```

```
        label = letter
        plt.scatter(x=a1, y=a2,  marker=marker, color=color, alpha=0.5,
                            label='{:}, R={:.2f}'.format(label, R[0]))
        # Отрисовка данных
        plt.title('Letter recognition')
        plt.xlabel('width')
        plt.ylabel('height')
        plt.legend(loc='upper right')
        plt.show()


def main():
        alphabet=[chr(i) for i in range(ord('A'), ord('Z')+1)]
        letter_attr,letter_class=load_dataset("letter-recognition.csv")
        x_train, y_train, x_test, y_test=split_dataset(letter_attr,letter_class,0.4)
        visualize_data(letter_attr,letter_class, alphabet)
        #Выполнить разбиение классов набора данных с помощью LDA
(LinearDiscriminantAnalysis). \
        #Осуществить визуализацию разбиения
        lda = LinearDiscriminantAnalysis(n_components=2)
        trans = lda.fit(x_train, y_train).transform(x_train)
        for letter,marker,color in zip(('O', 'I'), ('x', 'o'), ('red', 'blue')):
        x = [trans[i][0] for i in range(len(trans)) if y_train[i] == letter]
        y  = [trans[i][1] for i in range(len(trans)) if y_train[i] == letter]
        label = letter
        plt.scatter(x=x, y=y, marker=marker, color=color, alpha=0.5,
                            label=label)
```

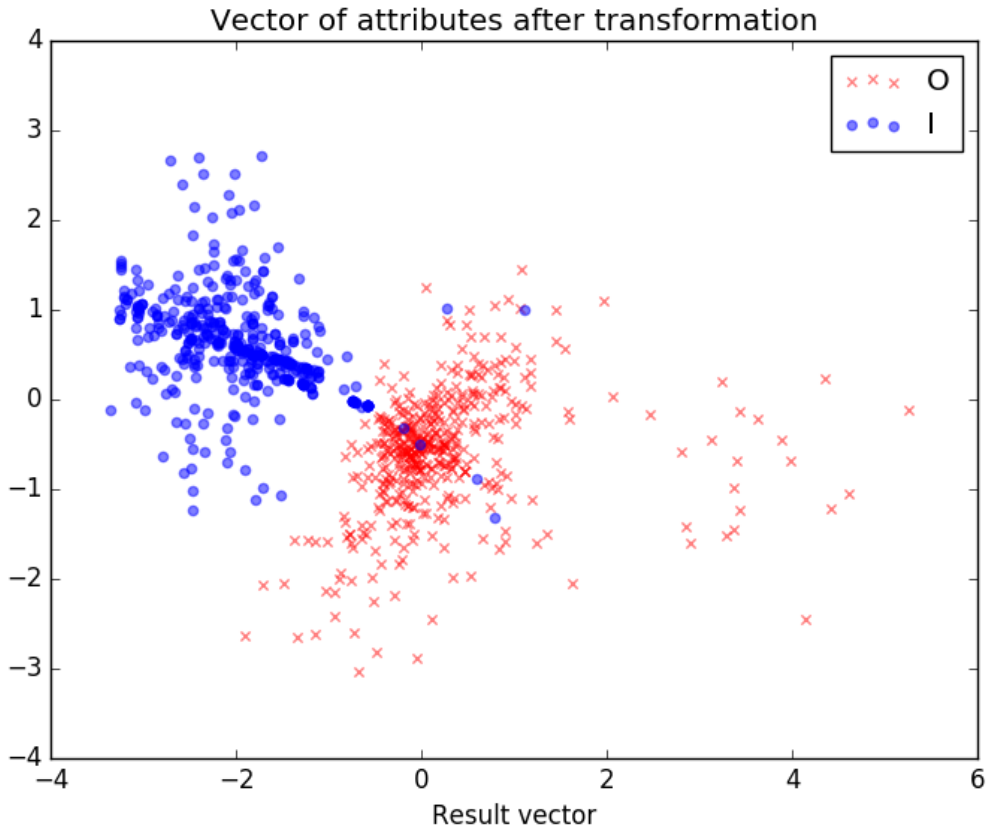
```
    plt.title('Vector of attributes after transformation')
    plt.xlabel('Result vector')
    plt.legend(loc='upper right')
    plt.show()
    #Осуществить классификацию с помощью методов LDA и QDA (LinearDiscriminantAnalysis и QuadraticDiscriminantAnalysis).
    # Обучение и тестирование
    qda = QuadraticDiscriminantAnalysis()
    qda.fit(x_train, y_train)
    #Сравнить полученные результаты
    print('Точность LDA: {:.2f}'.format(lda.score(x_test, y_test)))
    print('Точность QDA: {:.2f}'.format(qda.score(x_test, y_test)))

main()
```

Результаты работы программы:

Визуализация корреляции ширины и высоты букв



Визуализацию после разбиения с помощью LDA



Точность LDA: 0.70

Точность QDA: 0.89

4. Выводы

В ходе данной лабораторной работы были рассчитаны коэффициенты корреляции между шириной и высотой для каждой буквы латинского алфавита, результаты были отражены на графике. Также было проведено сравнения алгоритмов LDA и QDA, и можно сделать вывод о большей точности алгоритма QDA.