

Лабораторная работа №8

Классификация (линейный дискриминантный анализ, метод опорных векторов)

Цель:

Ознакомиться с методами классификации модуля Sklearn

Выполнение:

Загрузка данных:

1. Загрузить датасет по ссылке: <https://archive.ics.uci.edu/ml/datasets/iris> .
Данные представлены в виде data файла. Данные представляют собой информацию о трех классах цветов
2. Создать Python скрипт. Загрузить данные в датафрейм

```
import pandas as pd
import numpy as np

data = pd.read_csv('iris.data', header=None)
```

3. Выделим данные и их метки

```
x = data.iloc[:, :4].to_numpy()
labels = data.iloc[:, 4].to_numpy()
```

4. Преобразуем тексты меток к числам

```
le = preprocessing.LabelEncoder()
Y = le.fit_transform(labels)
```

5. Разобьём выборку на обучающую и тестовую

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.5)
```

Линейный дискриминантный анализ

1. Проведем классификацию наблюдений используя [LDA](#)

```

X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.5,
random_state=0)

from sklearn.discriminant_analysis import LinearDiscriminantAnalysis

clf = LinearDiscriminantAnalysis()
y_pred = clf.fit(X_train, y_train).predict(X_test)
print((y_test != y_pred).sum()) #количество наблюдений, который были
неправильно определены

```

Опишите атрибуты и параметры данного классификатора

- Используя функцию `score()` выведите точность классификации
- Постройте график зависимости неправильно классифицированных наблюдений и точности классификации от размера тестовой выборки. Размер тестовой выборки изменяйте от 0.05 до 0.95 с шагом 0.05. Параметр `random_state` сделайте равным номеру своей зачетной книжки. Обоснуйте полученные результаты.
- Опишите для чего нужна функция `transform`? Примените ее, и визуализируйте результаты.
- Исследуйте работу классификатор при различных параметрах **`solver`**, **`shrinkage`**.
- Задайте априорную вероятность классу с номером 1 равную 0.7, остальным классам задайте равные априорные вероятности. Как это сказалось на результате?

Метод опорных векторов

- Классификацию при [SVM](#) на тех же данных

```

clf = svm.SVC()
y_pred = clf.fit(X_train, y_train).predict(X_test)
print((y_test != y_pred).sum())
print(clf.score(X, Y))

```

- Используя функцию `score()` выведите точность классификации
- Выведите следующую информацию

```

print(clf.support_vectors_)
print(clf.support_)
print(clf.n_support_)

```

Объясните, что отображают данные параметры, от чего зависят.

- Постройте график зависимости неправильно классифицированных наблюдений и точности классификации от размера тестовой выборки. Размер тестовой выборки изменяйте от 0.05 до 0.95 с шагом 0.05. Параметр `random_state` сделайте равным номеру своей зачетной книжки. Обоснуйте полученные результаты.
- Исследуйте работу метода опорных векторов при различных значениях **`kernel`**, **`degree`**, **`max_iter`**
- Проведите исследование для методов [NuSVC](#) и [LinearSVC](#). В чем их отличие от `SVC`

