Лабораторная работа №7

Классификация (Байесовские методы, деревья)

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)
```

Байесовские методы

1. Проведем классификацию наблюдений наивный байесовским методов

```
from sklearn.naive_bayes import GaussianNB
gnb = GaussianNB()

y_pred = gnb.fit(X_train, y_train).predict(X_test)

print((y_test != y_pred).sum()) #количество наблюдений, который были
неправильно определены
```

Опишите атрибуты данного классификатора

2. Используя функцию score() выведите точность классификации
3. 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.
4. Проведите классификацию используя MultinomialNB, ComplementNB, BernoulliNB . Опишите особенности методов.

Классифицирующие деревья

1. Классификацию при помощи деревьях на тех же данных

```
from sklearn import tree

clf = tree.DecisionTreeClassifier()

y_pred = clf.fit(X_train, y_train).predict(X_test)
print((y_test != y_pred).sum())
```

2. Используя функцию score() выведите точность классификации
3. Выведите характеристики дерева, количество листьев и глубину, используя функции *get_n_leaves* и *get_depth*
4. Выведите изображение полученного дерева

```
import matplotlib.pyplot as plt

plt.subplots(1,1,figsize = (10,10))
tree.plot_tree(clf, filled = True)
plt.show()
```

Опишите полученный рисунок

5. 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.
6. Исследуйте работу классифицирующего дерева при различных параметрах **criterion**, **splitter**, **max_depth**, **min_samples_split**, **min_samples_leaf**