МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ

по лабораторной работе №3

по дисциплине «Информационные технологии»

Тема: Введение в анализ данных

Студент гр. 2384 _____ Кузьминых Е.М

Преподаватель _____ Иванов Д.В.

Санкт-Петербург

2023

Цель работы.

Научиться использовать библиотеку *sklearn*, создать программу, которая будет загружать информацию, на основе ее обучать модель и анализировать полученные данные.

Задачи.

Вариант №3

Вы работаете в магазине элитных вин и собираетесь провести анализ существующего ассортимента, проверив возможности инструмента классификации данных для выделения различных классов вин.

Для этого необходимо использовать библиотеку *sklearn* и встроенный в него набор данных о вине.

1) В функции *loadData()* загрузить набор данных о вине из библиотеки *sklearn* в переменную *wine*. Разбейте данные для обучения и тестирования в соотношении 80% к 20% соответственно, следующим образом: из данного набора запишите 80% данных из *data*, взяв при этом только 2 столбца в переменную *X_train* и 80% данных поля *target* в *y_train*. В переменную *X_test*- положите оставшиеся 20% данных из *data*, взяв при этом только 2 столбца, а в *y_tes*t — оставшиеся 20% данных поля *target*. В качестве результата работы функции верните *X_train, y_train, X_test, y_test*. В качестве состояния рандомизатора необходимо указать 42.

Пояснение: *X_train, X_test* - двумерный массив, *y_train, y_test*. — одномерный массив.

2) Обучение модели. Выполнение классификации:

В функции классификации однородных весов *distance()* создайте экземпляр классификатора соседей *KNeighborsClassifier* и загрузите в него данные *X_train, y_train* из пункта выше с *n_neighbors* = 15 и *weights* = *d*istance.

В качестве результата верните экземпляр классификатора соседей.

3) Выполнение анализа данных. Реализуйте функцию *predict()*. Выполните предсказание данных, используя данные *X_test* — (те 20% данных из *data*). Верните предсказанные данные.

4) Оцените качество полученных результатов классификации.

В функции *estimate()* сравните *y_test* с результатом, полученным Вами в пункте 3. Посчитайте и верните отношение предсказанных результатов, совпавших с «правильными» в *y_test* к количеству результатов. (Или другими словами, ответить на вопрос «На сколько качественно отработала модель в процентах»). Объясните полученные результаты.

Пояснение: так как это вероятность, то ответ должен находиться в диапазоне [0, 1].

Выполнение работы.

В начале программы импортируются необходимые библиотеки и классификатор *KNN* из *Scikit-Learn*.

Функция *loadData()*: Эта функция загружает набор данных о вине, используя функцию *load_wine()*. Затем функция разделяет данные на обучающую и тестовую выборки, используя функцию *train_test_split()*. Данные разбиваются на *X_train, X_test, y_train* и *y_test*. Для этого берутся только первые два столбца из wine.data и поле target. Функция возвращает *X_train, X_test, y_train* и *y_test*.

Функция *distance(X_train, y_train)*: Эта функция создает объект классификатора *KNN* с параметрами *n_neighbors=15* и *weights='distance'*. Затем она обучает классификатор на обучающей выборке, используя метод fit(), и возвращает обученный классификатор.

Функция *predict(clf, X_test)*: Эта функция использует обученный классификатор, чтобы сделать предсказания на тестовой выборке, используя метод predict().

Функция *estimate(y_pred, y_test)*: Эта функция использует функцию mean(), чтобы вычислить точность классификации. Она сравнивает предсказанные

метки *y_pred* с истинными метками *y_test* с помощью выражения *(y_pred==y_test).mean()* и возвращает точность классификации.

Тестирование.

№	Входные данные	Выходные данные	Комментарий
1	X_train, X_test, y_train, y_test = loadData() clf = distance(X_train, y_train) res = predict(clf, X_test) est = estimate(res, y_test) print(y_train)	[2 2 1 2 0 1 1 1 2 0 1 1 2 0 1 0 0 2 2 1 1 0 1 0 2 1 1 2 0 0 0 2 0 0 1 2 1 0 2 1 0 2 1 1 0 1 0 0 1 0 0 2 1 1 1 0 1 1 1 2 2 0 1 2 2 1 1 0 1 2 2 1 2 1 1 1 0 0 2 0 2 0 0 1 1 0 0 0 1 0 1 2 1 1 1 2 2 1 0 0 1 2 2 0 1 2 2 2 2 1 0 1 0 2 0 0 1 0 0 2 1 0 2 2 0 0 2 2 2 1 1 1 1 1 1 2 0 1 1 0 1 1]	Ответ верный
2	X_train, X_test, y_train, y_test = loadData() clf = distance(X_train, y_train) res = predict(clf, X_test) est = estimate(res, y_test) print(y_test)	[0 0 2 0 1 0 1 2 1 2 0 2 0 1 0 1 1 1 0 1 0 1 1 2 2 2 1 1 1 0 0 1 2 0 0 0]	Ответ верный

Вывод.

В ходе выполнения лабораторной работы была написана программа на языке программирования *Python* с использованием библиотеки *scikit-learn*. Программа загружает данные о вине, разбивает их на обучающую и тестовую выборки, обучает модель классификации методом k-ближайших соседей, делает предсказания для тестовой выборки и оценивает точность модели.

Приложение А.

Исходный код программы.

```
from sklearn.datasets import load_wine
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier


def loadData():
    wine = load_wine()

    X_train, X_test, y_train, y_test = train_test_split(
        wine.data[:,      :2],      wine.target,      test_size=0.2,
random_state=42)

    return X_train, X_test, y_train, y_test


def distance(X_train, y_train):
    clf = KNeighborsClassifier(n_neighbors=15, weights='distance')

    clf.fit(X_train, y_train)
    return clf


def predict(clf, X_test):
    return clf.predict(X_test)


def estimate(y_pred, y_test):
    return (y_pred == y_test).mean()
```