

**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»**

Студентка гр. 2384

Валеева А.А.

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.

## Задание

### Вариант №1

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_test` — 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 `uniform()` создайте экземпляр классификатора соседей `KNeighborsClassifier` и загрузите в него данные `X_train`, `y_train` из пункта выше с `n_neighbors = 15` и `weights = uniform`.

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

Функция *loadData()*:

С помощью функции *load\_wine()* загружаем информацию о вине. Сохраняем в переменные *X* и *y* данные полей *data* и *target*. Функция разбивает данные *X\_train*, *X\_test*, *y\_train*, *y\_test* при помощи функции *train\_test\_split()*.

Функция *uniform()*:

Функция создает экземпляр классификатора соседей *KNeighborsClassifier* (где количество соседей равно 15) и с помощью функции *fit()* загружаем в него данные *X\_train*, *y\_train*. Функция возвращает экземпляр классификатора.

Функция *predict()*:

Функция принимает на вход экземпляр классификатора соседей *KNeighborsClassifier* и данные *X\_test*. С помощью функции *predict()* получаем и возвращаем предсказанные данные на основе данных *X\_test*.

Функция *estimate()*:

Вычисляет точность предсказаний, возвращает отношение суммы последнего столбца к его длине.

Разработанный программный код см. в приложении А.

Результаты тестирования см. в приложении Б.

## Выводы.

Изучен принцип работы с библиотекой *sklearn*, реализована программа, выполняющая действия с загруженной информацией. Выполнены функции на его загрузку, анализ информации, оценку полученных результатов классификации.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: Valeyeva\_Alina\_lb3.py

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

def loadData():
    wine = datasets.load_wine()
    return model_selection.train_test_split(wine.data[:,2:], wine.target,
train_size=0.8, random_state=42)

def uniform(X_train, y_train):
    knn = KNeighborsClassifier(n_neighbors=15, weights='uniform')
    knn.fit(X_train, y_train)
    return knn

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

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

## ПРИЛОЖЕНИЕ Б.

### ТЕСТИРОВАНИЕ

Таблица Б.1 – результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	X_train, X_test, y_train, y_test = loadData() clf = uniform(X_train, y_train) res = predict(clf, X_test) est = estimate(res, y_test)  print(X_train) print(y_train) print(X_test) print(y_test)  print(clf) print(res) print(est)	KNeighborsClassifier(n _neighbors=15) [2 2 2 0 1 0 1 2 1 2 0 2 2 2 0 2 1 1 0 1 0 1 1 2 2 2 1 1 1 0 0 1 2 0 0 0] 0.86111111111111112	Верный ответ.