

## **Лабораторная работа № 3**

### **Тема: «Использование базовых структур данных и библиотек для обработки больших данных в Python»**

**Цель работы:** изучить методы работы с библиотекой numpy.

#### **Теоретическая справка**

NumPy (Numerical Python) - это библиотека Python, предназначенная для выполнения научных и числовых вычислений. Она предоставляет поддержку для многомерных массивов, высокоуровневых математических функций, таких как линейная алгебра, преобразование Фурье и генерация случайных чисел.

#### **Основы работы с массивами в NumPy**

Массивы в NumPy, известные как `ndarray`, являются основными объектами, с которыми работает NumPy. Массивы похожи на списки в Python, но они могут иметь больше одного измерения и все элементы массива должны быть одного типа (например, все элементы являются целыми числами или числами с плавающей точкой).

Создание массива в NumPy может быть таким простым, как преобразование обычного списка или списка списков в массив с помощью функции `numpy.array()`. Например:

```
import numpy as np

# Создание одномерного массива из списка
array1d = np.array([1, 2, 3, 4])

# Создание двумерного массива из списка списков
array2d = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
```

#### **Операции над массивами**

NumPy предоставляет большое количество операций для работы с массивами. Вы можете выполнять арифметические операции (сложение, вычитание, умножение и т.д.) над массивами так же, как вы бы это делали с обычными числами. Операции выполняются поэлементно. Например:

```
import numpy as np

a = np.array([1, 2, 3])
b = np.array([4, 5, 6])
```

## Курс «Базовая обработка данных на языке Python»

автор: Киреев В.С., к.т.н., доцент

```
# Сложение массивов
c = a + b # array([5, 7, 9])

# Умножение массива на скаляр
d = a * 2 # array([2, 4, 6])
```

### Индексация и срезы

Массивы в NumPy поддерживают индексацию и срезы, как и обычные списки в Python. Вы можете обращаться к отдельным элементам массива и получать или изменять их значения. Вы также можете получить подмассив из массива, используя срезы. Например:

```
import numpy as np

a = np.array([1, 2, 3, 4, 5])

# Получение значения элемента по индексу
first_element = a[0] # 1

# Изменение значения элемента по индексу
a[0] = 10 # array([10, 2, 3, 4, 5])

# Получение среза массива
slice = a[1:3] # array([2, 3])
```

### Математические функции и операции

NumPy включает множество математических функций, которые можно использовать для выполнения различных операций над массивами. Например, вы можете вычислить синус всех элементов массива, найти максимальное значение в массиве, вычислить среднее значение элементов массива и многое другое.

Все эти возможности делают NumPy мощным инструментом для научных вычислений в Python.

### Самостоятельное задание

1. Задания на создание массивов NumPy и простейшие операции над ними. В данных заданиях используйте генераторы случайных чисел, число уникальных значений должно быть 4-9, и лежать в пределах от 1 до 10.
  - 1.1. Создайте массивы V1, V2, V3 – типа int и размерности 10 элементов
  - 1.2. Создайте вектор V, объединив вертикально (один под другим) V1, V2, V3
  - 1.3. Создайте матрицу M размерности 30 x 30, типа int.
  - 1.4. Сложите поэлементно массивы V1 и V2
  - 1.5. Сравните массивы V1 и V2 и найдите несовпадающие значения
2. Задания на индексацию (поиск и сортировку значений) массивов NumPy
  - 2.1. Выберите из матрицы M – значение ячейки из последнего столбца и последней строки
  - 2.2. Выберите из матрицы M – строки с 10 по 20, и столбцы с 5 по 6

## Курс «Базовая обработка данных на языке Python»

автор: Киреев В.С., к.т.н., доцент

- 2.3. Найдите второе по убыванию значение матрицы  $M$
- 2.4. Извлеките диагональные элементы из матрицы  $M$
- 2.5. Найдите в матрице  $M$  значения больше 5 (создайте маску для матрицы)
3. Задания на манипуляцию массивами Numpy
  - 3.1. Измените тип данных массива  $V$  на строковый
  - 3.2. Обратите массив  $V$  (первое значение должно стать последним и наоборот)
  - 3.3. Сделайте матрицу  $M$  плоским массивом
  - 3.4. Переделайте форму вектора  $V$  на матрицу размерности  $5 \times 2$
  - 3.5. Удалите из вектора  $V$  значения меньше 2.
4. Задания на статистические операции
  - 4.1. Рассчитайте среднее, медиану и дисперсию по вектору  $V$
  - 4.2. Рассчитайте среднее, медиану и дисперсию по матрице  $M$
  - 4.3. Постройте эмпирическую функцию плотности распределения вектора  $V$ , визуализируйте полученную функцию с помощью `matplotlib`.
  - 4.4. Рассчитайте матрицу корреляции для матрицы  $M$
  - 4.5. Создайте два случайных вектора размерности 10, типа `int`, каждый из которых содержит 3 уникальных значения, и проверьте гипотезу о их независимости с помощью критерия Хи-квадрат
5. Задания на векторную и матричную алгебру
  - 5.1. Умножьте матрицу  $M$  на вектор  $V$ .
  - 5.2. Найдите определитель матрицы  $M$
  - 5.3. Найдите скалярное произведение векторов  $V$  и транспонированного  $V$
  - 5.4. Решите систему уравнений  $MX=V$
  - 5.5. Рассчитайте собственные значения и собственные вектора для матрицы  $M$
6. Скачайте два изображения товаров с Wildberries из одной категории (мужские костюмы), преобразуйте их в многомерные массивы Numpy и определите расстояние Евклида между ними.
7. Реализуйте метод главных компонент с помощью модуля Numpy и нахождения собственных векторов. Преобразуйте одну из картинок из п.6 в матрицу новых компонент, меньшей размерности и визуализируйте ее.
8. Кластеризуйте одну из картинок из п.6 – выделите отдельно брюки и отдельно пиджак на изображении. Можете использовать метод K-means.
9. Определите цвет фона картинки из п.8 – найдите наиболее частый цвет в матрице, и по вектору (R,G,B) в `matplotlib`, выведите полученный цвет.
10. Выберите картинку из п.6 и примените к ней фильтр Калмана.