

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт перспективной инженерии
Департамент цифровых, робототехнических систем и электроники

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №2
дисциплины
«Искусственный интеллект и машинное обучение»

Выполнил:
Митряшкина Дарина Сергеевна
2 курс, группа ИТС-б-о-23-1,
11.03.02 «Инфокоммуникационные
технологии и системы связи»,
направленность (профиль)
«Инфокоммуникационные системы и
сети», очная форма обучения

(подпись)

Проверил:
Доцент департамента цифровых,
робототехнических систем и
электроники Воронкин Р.А.

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2025 г.

Тема: Основы работы с библиотекой NumPy

Цель: исследовать базовые возможности библиотеки NumPy языка программирования Python.

Порядок выполнения работы:

1. Изучен теоретический материал по лабораторной работе
2. Создан общедоступный репозиторий на GitHub
3. Создан новый ноутбук на Google Colab

✓ Лабораторная работа №2

Тема: NumPy. Работа с массивами и матрицами **Название репозитория:** second laba ai **Язык:** русский

Рисунок 1. Новый ноутбук на Google Colab

4. Выполнение практических заданий, указанных в лабораторной работе:

✓ Задание 1

```
[ ]  
  
import numpy as np  
  
arr = np.arange(1, 10).reshape(3, 3)  
arr = arr * 2  
arr[arr > 10] = 0  
print(arr)
```

Рисунок 2. Задание 1

✓ Задание 2



```
arr = np.random.randint(1, 101, 20)
print("Исходный массив:", arr)

div_by_5 = arr[arr % 5 == 0]
print("Элементы, делящиеся на 5:", div_by_5)

arr[arr % 5 == 0] = -1
print("Обновленный массив:", arr)
```

Рисунок 3. Задание 2

✓ Задание 3



```
a = np.random.randint(0, 51, 5)
b = np.random.randint(0, 51, 5)
print("Массив A:", a)
print("Массив B:", b)

combined = np.vstack((a, b))
print("Объединенный массив:",
      combined)

split1, split2 = combined[0], combined[1]
print("Первый массив:", split1)
print("Второй массив:", split2)
```

Рисунок 4. Задание 3

✓ Задание 4

```
[ ]  
arr = np.linspace(-10, 10, 50)  
sum_all = np.sum(arr)  
sum_positive = np.sum(arr[arr > 0])  
sum_negative = np.sum(arr[arr < 0])  
  
print(f"Сумма всех элементов: {sum_all}")  
print(f"Сумма положительных элементов: {sum_positive}")  
print(f"Сумма отрицательных элементов: {sum_negative}")
```

Рисунок 5. Задание 4

Задание 5

```
]   
identity_matrix = np.eye(4)  
diagonal_matrix = np.diag([5, 10, 15, 20])  
  
sum_identity = np.sum(identity_matrix)  
sum_diagonal = np.sum(diagonal_matrix)  
  
print("Сумма элементов единичной матрицы:", sum_identity)  
print("Сумма элементов диагональной матрицы:", sum_diagonal)
```

Рисунок 6. Задание 5

Задание 6

```
A = np.random.randint(1, 21, (3, 3))
B = np.random.randint(1, 21, (3, 3))

print("Матрица A:
", A)
print("Матрица B:
", B)

print("Сумма A + B:
", A + B)
print("Разность A - B:
", A - B)
print("Поэлементное произведение A * B:
", A * B)
```

Рисунок 7. Задание 6

Задание 7

```
A = np.random.randint(1, 11, (2, 3))
B = np.random.randint(1, 11, (3, 2))

print("Матрица A:
", A)
print("Матрица B:
", B)

C = A @ B
print("Результат умножения A @ B:
", C)
```

Рисунок 8. Задание 7

Задание 8

```
matrix = np.random.randint(1, 11, (3, 3))
print("Матрица:
", matrix)

det = np.linalg.det(matrix)
print("Определитель:", det)

if np.isclose(det, 0):
    print("Матрица вырождена, обратной матрицы нет.")
else:
    inverse = np.linalg.inv(matrix)
    print("Обратная матрица:
", inverse)
```

Рисунок 9. Задание 8

Задание 9

```
]
matrix = np.random.randint(1, 51, (4, 4))
print("Исходная матрица:
", matrix)

print("Транспонированная матрица:
", matrix.T)

trace = np.trace(matrix)
print("След матрицы:", trace)
```

Рисунок 10. Задание 9

Задание 10

```
]
A = np.array([
    [2, 3, -1],
    [4, -1, 2],
    [-3, 5, 4]
])

B = np.array([5, 6, -2])

solution = np.linalg.solve(A, B)
print("Решение системы:", solution)
```

Рисунок 11. Задание 10

Задание 11

```
# Финансовая стратегия инвестора
# Обозначим x – вложение во 2-ю компанию
# Тогда в 1-ю – 2x, в 3-ю – x+10000
#  $2x + x + (x + 10000) = 300000$ 
# Решим систему

import numpy as np

A = np.array([
    [2, 1, 0],
    [0, 1, 1],
    [1, 1, 1]
])

B = np.array([0, 10000, 300000])

# Решение с помощью np.linalg.solve
solution = np.linalg.solve(A, B)
print("Решение через np.linalg.solve:", solution)

# Решение методом Крамера
from numpy.linalg import det
```

```

detA = det(A)
detA1 = det(np.column_stack((B, A[:,1], A[:,2])))
detA2 = det(np.column_stack((A[:,0], B, A[:,2])))
detA3 = det(np.column_stack((A[:,0], A[:,1], B)))

x1 = detA1 / detA
x2 = detA2 / detA
x3 = detA3 / detA

print("Решение методом Крамера:", [x1, x2, x3])

```

Рисунок 12. Задание 11

Ответы на контрольные вопросы:

1. **Каково назначение библиотеки NumPy?**

Библиотека NumPy предназначена для эффективной работы с многомерными массивами и матрицами, а также предоставляет широкий спектр математических функций.

2. **Что такое массивы ndarray?**

Это основная структура данных в NumPy — многомерный массив фиксированного размера, содержащий элементы одного типа.

3. **Как осуществляется доступ к элементам многомерного массива?**

Через индексирование и срезы, аналогично спискам в Python, но с использованием нескольких индексов.

4. **Как осуществляется расчет статистик по данным?**

С помощью функций NumPy: mean, median, std, var, sum, min, max и других.

5. **Как выполняется выборка данных из массивов ndarrays?**

Путем индексирования, срезов, логической маскировки и fancy-indexing (расширенное индексирование массивами индексов).

6. **Приведите основные виды матриц и векторов.**

Векторы (1D массивы), матрицы (2D массивы), единичная матрица, диагональная матрица, нулевая матрица.

7. Как выполняется транспонирование матриц?

С помощью атрибута `.T`, функции `transpose()` или метода `swapaxes()`.

8. Приведите свойства операции транспонирования матриц.

$(AB)^T = B^T A^T$, $(A^T)^T = A$.

9. Какие имеются средства в библиотеке NumPy для выполнения транспонирования матриц?

`.T`, `numpy.transpose(array)`.

10. Какие существуют основные действия над матрицами?

Сложение, вычитание, умножение на число, матричное умножение, транспонирование, нахождение детерминанта, обратной матрицы.

11. Как осуществляется умножение матриц на число?

Поэлементное умножение: каждый элемент матрицы умножается на скаляр.

12. Какие свойства операции умножения матриц на число?

Ассоциативность: $(kA)B = k(AB)$.

13. Как осуществляются операции сложения и вычитания матриц?

Поэлементно: складываются или вычитаются соответствующие элементы двух матриц одинаковой размерности.

14. Какие свойства операций сложения и вычитания матриц?

Коммутативность $(A+B = B+A)$ и ассоциативность $((A+B)+C = A+(B+C))$.

15. Какие имеются средства в библиотеке NumPy для выполнения операций сложения и вычитания матриц?

Операторы `+`, `-`, а также функции `numpy.add()`, `numpy.subtract()`.

16. Как осуществляется операция умножения матриц?

С помощью оператора `@` или функции `numpy.dot(A, B)`.

17. Каковы свойства операции умножения матриц?

Ассоциативность $(A(BC) = (AB)C)$, но не коммутативность $(AB \neq BA)$ в общем случае).

18. Какие имеются средства в библиотеке NumPy для выполнения операций умножения матриц?

Оператор `@`, функции `numpy.matmul()`, `numpy.dot()`.

19. Что такое определитель матрицы? Какие свойства определителя матрицы?

Определитель (детерминант) показывает масштаб изменения площади/объема после преобразования матрицей. Свойства: $\det(AB) = \det(A)\det(B)$, $\det(A^T) = \det(A)$.

20. Какие имеются средства в библиотеке NumPy для нахождения значения определителя матрицы?

Функция `numpy.linalg.det(matrix)`.

21. Что такое обратная матрица? Какой алгоритм нахождения обратной матрицы?

Обратная матрица A^{-1} такая, что $AA^{-1} = I$. Находится через метод Гаусса-Жордана или с помощью функции `numpy.linalg.inv(matrix)`.

22. Какие имеются средства в библиотеке NumPy для нахождения обратной матрицы?

Функция `numpy.linalg.inv()`.

23. Самостоятельно изучите метод Крамера для решения систем линейных уравнений. Какие имеются средства библиотеки NumPy для решения систем линейных уравнений методом Крамера?

Метод Крамера: решение через отношение определителей. Для ручного применения нужно использовать `numpy.linalg.det()` для подсчёта определителей. Встроенной отдельной функции для метода Крамера нет его реализуют вручную.

24. **Что такое матричный метод решения систем линейных уравнений? Какие имеются средства библиотеки NumPy для решения систем линейных уравнений матричным методом?**

Метод: если $Ax = B$, то $x = A^{-1}B$. Для решения в NumPy используется `numpy.linalg.solve(A, B)` (работает быстрее и точнее, чем явное нахождение A^{-1}).

Ссылка на Git Hub: <https://github.com/darina-rtm/second-lab-ai.git>

Ссылка на Google Colab: <https://colab.research.google.com/drive/1ndoWY-RLLhWYLRleRuH65nIb3dVaJ1zZ?usp=sharing>