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 №4
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: Введение в pandas. Изучение структуры Series и базовых операций.

Цель: и познакомить с основами работы с библиотекой pandas, в частности, со структурой данных Series.

Порядок выполнения работы:

1. Создан новый ноутбук, добавлены необходимые библиотеки

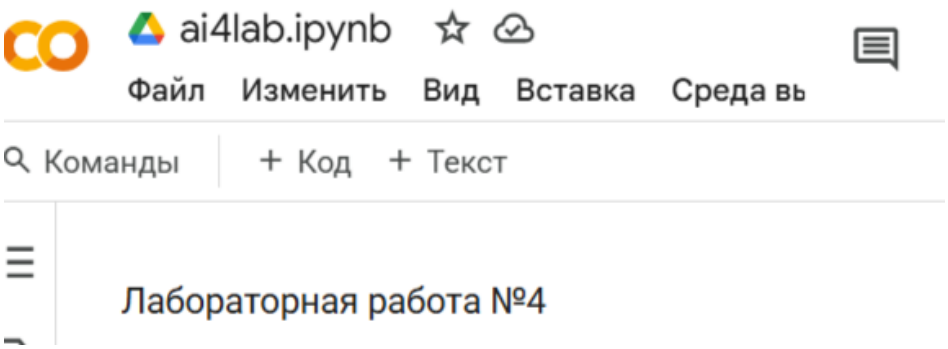

Рисунок 1. Новый ноутбук для выполнения заданий

2. Выполнение задания №1.

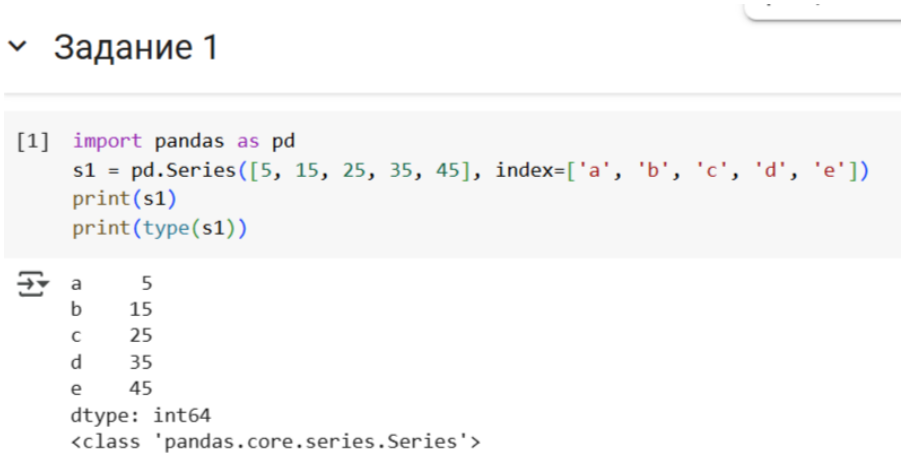

Рисунок 2. Задание №1

3. Выполнение задания №2.

Задание 2

```
[ ]  import pandas as pd
     s2 = pd.Series([12, 24, 36, 48, 60], index=['A', 'B', 'C', 'D', 'E'])
     print('loc[C]:', s2.loc['C'])
     print('iloc[2]:', s2.iloc[2])
```

Рисунок 3. Задание №2

4. Выполнение задания №3.

Задание 3

```
[ ]  import pandas as pd
     import numpy as np
     arr = np.array([4, 9, 16, 25, 36, 49, 64])
     s3 = pd.Series(arr)
     print(s3[s3 > 20])
```

Рисунок 4. Задание №3

5. Выполнение задания №4.

Задание 4

```
[ ]  import pandas as pd
     import numpy as np
     s4 = pd.Series(np.random.randint(1, 101, 50))
     print('head():')
     print(s4.head(7))
     print('tail():')
     print(s4.tail(5))
```

Рисунок 5. Задание №4

6. Выполнение задания №5.

Задание 5

```
[ ]  import pandas as pd
     s5 = pd.Series(['cat', 'dog', 'rabbit', 'parrot', 'fish'])
     print('dtype до:', s5.dtype)
     s5 = s5.astype('category')
     print('dtype после:', s5.dtype)
```

Рисунок 6. Задание №5

7. Выполнение задания №6.

Задание 6

```
[ ]  import pandas as pd
     import numpy as np
     s6 = pd.Series([1.2, np.nan, 3.4, np.nan, 5.6, 6.8])
     print('NaN индексы:', s6[s6.isna()].index.tolist())
```

Рисунок 7. Задание №6

8. Выполнение задания №7.

Задание 7

```
[ ]  import pandas as pd
     import numpy as np
     s6 = pd.Series([1.2, np.nan, 3.4, np.nan, 5.6, 6.8])
     mean_val = s6.mean()
     s6_filled = s6.fillna(mean_val)
     print(s6_filled)
```

Рисунок 8. Задание №7

9. Выполнение задания №8.

Задание 8

```
[ ]  import pandas as pd
     a = pd.Series([1, 2, 3])
     b = pd.Series([4, 5, 6])
     print('Сложение:')
     print(a + b)
     print('Умножение:')
     print(a * b)
```

Рисунок 9. Задание №8

10. Выполнение задания №9.

Задание 9

```
[ ]  import pandas as pd
     import numpy as np
     s9 = pd.Series([2, 4, 6, 8, 10])
     print(s9.apply(np.sqrt))
```

Рисунок 10. Задание №9

11. Выполнение задания №10.

Задание 10

```
[ ]  import pandas as pd
     import numpy as np
     s10 = pd.Series(np.random.randint(50, 151, 20))
     print('Сумма:', s10.sum())
     print('Среднее:', s10.mean())
     print('Мин:', s10.min())
     print('Макс:', s10.max())
     print('Ст. отклонение:', s10.std())
```

Рисунок 11. Задание №10

12. Выполнение задания №11.

Задание 11

```
[ ]  import pandas as pd
     import numpy as np
     dates = pd.date_range('2024-03-01', periods=10)
     s11 = pd.Series(np.random.randint(10, 100, 10), index=dates)
     print(s11['2024-03-05':'2024-03-08'])
```

Рисунок 12. Задание №11

13. Выполнение задания №12.

Задание 12

```
[ ]  import pandas as pd
     s12 = pd.Series([10, 20, 30, 40, 50, 60], index=['A', 'B', 'A', 'C', 'D', 'B'])
     print('Уникальны ли индексы?', s12.index.is_unique)
     print(s12.groupby(s12.index).sum())
```

Рисунок 13. Задание №12

14. Выполнение задания №13.

Задание 13

```
[ ]  import pandas as pd
     s13 = pd.Series([100, 200, 300], index=['2024-03-10', '2024-03-11', '2024-03-12'])
     s13.index = pd.to_datetime(s13.index)
     print('Тип индекса:', type(s13.index))
```

Рисунок 14. Задание №13

15. Выполнение задания №14.

Задание 14

```
[ ]  import pandas as pd
     s14 = pd.Series([90, 95, 100], index=pd.to_datetime(['2024-01-01', '2024-01-02', '2024-01-03']))
     print(s14)
```

Рисунок 15. Задание №14

16. Выполнение задания №15.

Задание 15

```
[ ]  import pandas as pd
     import numpy as np
     s15 = pd.Series([10, None, 30, None, 20], index=['a', 'b', 'c', 'd', 'e'])
     print(s15.sort_values())
     print(s15.sort_values(na_position='first'))
```

Рисунок 16. Задание №15

17. Выполнение индивидуального задания №10.

Индивидуальное задание 10

```
[ ]  import pandas as pd
     import numpy as np
     s_ind10 = pd.Series(np.random.randint(50, 151, 20))
     print(s_ind10)
     print("\nСумма:", s_ind10.sum())
     print("Среднее:", s_ind10.mean())
     print("Минимум:", s_ind10.min())
     print("Максимум:", s_ind10.max())
     print("Стандартное отклонение:", s_ind10.std())
```

Рисунок 17. Индивидуальное задание №10

Ответы на контрольные вопросы:

1. Что такое pandas.Series и чем она отличается от списка в Python?

pandas.Series — это одномерная структура данных, похожая на список или массив, но с возможностью именования индексов. В отличие от списка, Series имеет встроенную ось индексов и множество встроенных методов анализа данных.

2. Какие типы данных можно использовать для создания Series?

Можно использовать: числа, строки, логические значения, даты, объекты, списки, словари, массивы NumPy и др.

3. Как задать индексы при создании Series?

Передав аргумент index:

pd.Series([10, 20, 30], index=['a', 'b', 'c'])

4. Каким образом можно обратиться к элементу Series по его индексу?

По метке: s['a']

По позиции: s[0]

5. В чём разница между .iloc[] и .loc[] при индексации Series?

- .iloc[] — индексирует по **позиции** (числовой индекс).
- .loc[] — индексирует по **метке** (именованному индексу).

6. Как использовать логическую индексацию в Series?

Передаётся булев массив:

s[s > 10]

7. Какие методы можно использовать для просмотра первых и последних элементов Series?

- .head(n) — первые n элементов.
- .tail(n) — последние n элементов.

8. Как проверить тип данных элементов Series?

Через .dtype:

s.dtype

9. Каким способом можно изменить тип данных Series?

С помощью .astype():

s.astype(float)

10. Как проверить наличие пропущенных значений в Series?

- s.isna() или s.isnull() — булев массив.
- s.hasnans — булевый флаг.

11. Какие методы используются для заполнения пропущенных значений в Series?

- .fillna(value)
- .ffill() — заполнение предыдущим значением.
- .bfill() — заполнение следующим значением.

12. Чем отличается метод .fillna() от .dropna()?

- .fillna() — **заменяет** NaN.
- .dropna() — **удаляет** элементы с NaN.

13. Какие математические операции можно выполнять с Series?

Сложение, вычитание, умножение, деление, возведение в степень и др. Например:

s + 10

s * 2

14. В чём преимущество векторизированных операций по сравнению с циклами Python?

Они **быстрее**, потому что используют оптимизированные низкоуровневые реализации (C/NumPy).

15. Как применить пользовательскую функцию к каждому элементу Series?

С помощью .apply() или map():

s.apply(lambda x: x * 2)

16. Какие агрегирующие функции доступны в Series?

- .sum()
- .mean()
- .min()
- .max()
- .std()
- .count()
- .median()

17. Как узнать минимальное, максимальное, среднее и стандартное отклонение Series?

s.min(), s.max(), s.mean(), s.std()

18. Как сортировать Series по значениям и по индексам?

- По значениям: s.sort_values()
- По индексам: s.sort_index()

19. Как проверить, являются ли индексы Series уникальными?

s.index.is_unique

20. Как сбросить индексы Series и сделать их числовыми?

s.reset_index(drop=True)

21. Как можно задать новый индекс в Series?

Присвоить список индексов:

s.index = ['a', 'b', 'c']

22. Как работать с временными рядами в Series?

Установить DatetimeIndex и использовать срезы по датам.

s = pd.Series(data, index=pd.date_range(...))

23. Как преобразовать строковые даты в формат DatetimeIndex?

pd.to_datetime(['2024-01-01', '2024-01-02'])

24. Каким образом можно выбрать данные за определённый временной диапазон?

С помощью срезов по дате:

s['2024-01-01':'2024-01-31']

25. Как загрузить данные из CSV-файла в Series?

pd.read_csv('file.csv', squeeze=True, usecols=[1])

26. Как установить один из столбцов CSV-файла в качестве индекса Series?

pd.read_csv('file.csv', index_col='column_name')

27. Для чего используется метод .rolling().mean() в Series?

Для **скользящего среднего**, сглаживания колебаний.

s.rolling(window=3).mean()

28. Как работает метод .pct_change()? Какие задачи он решает?

Вычисляет процентное изменение между текущим и предыдущим значением. Полезно в анализе финансовых данных.

s.pct_change()

29. В каких ситуациях полезно использовать .rolling() и .pct_change()?

- .rolling() — для анализа трендов, сглаживания.
- .pct_change() — для анализа динамики, доходности, изменений.

30. Почему NaN могут появляться в Series, и как с ними работать?

Причины:

- Отсутствие данных.
- Операции с пропущенными значениями.
- Расчёты вроде деления на 0 или rolling.

Работа:

- Обнаружение: .isna()
- Заполнение: .fillna()
- Удаление: .dropna()

Вывод: в ходе изучения структуры данных pandas.Series были рассмотрены её основные свойства, способы создания и обработки. Series представляет собой мощную и удобную структуру для работы с одномерными данными, обеспечивающую поддержку индексов, встроенные методы анализа, а также высокую производительность за счёт векторизированных операций.

Были изучены различные способы обращения к элементам Series, включая позиционную и меточную индексацию, методы сортировки,

логическую фильтрацию и агрегацию данных. Особое внимание уделено работе с пропущенными значениями, типами данных, пользовательскими функциями, а также методам для анализа временных рядов.

Важными инструментами анализа временных данных являются методы .rolling() и .pct_change(), которые позволяют проводить сглаживание и оценивать изменения между периодами. Также был рассмотрен процесс импорта данных из CSV-файлов и преобразования строковых дат в формат DatetimeIndex.

Ссылка на Google Colab: https://colab.research.google.com/drive/1LrmjDasRRJto5vMvfweoG4Dm3IlWq20?usp=sharing

Ссылка на Git Hub: https://github.com/darina-rtm/ai4laba.git