ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ ДАННЫХ

Лекция 7

NumPy: массивы и числовые расчеты

Оспан Ә.Ғ., PhD, доцент-исследователь

КазНУ, кафедра ИИ и Big Data

2026-2027 учебный год

Цель и содержание лекции

Цель и результаты обучения

- Объяснять shape, dtype и смысл осей массива.
- Использовать срезы, маски и broadcasting.
- Проверять числовые расчеты и интерпретировать статистики.

Содержание лекции

1. **Устройство ndarray**
2. **Оси и индексация**
3. **Копии и маски**
4. **Формы и broadcasting**
5. **Статистики**

ndarray: форма, тип и единая операция

Форма

- shape задает длины осей.
- Вектор и матрица требуют разных индексов.

Тип

- dtype описывает представление ячейки.
- Смешанные данные могут сменить общий тип.

Векторизация

- Операция применяется к массиву целиком.
- Короткая формула не отменяет контроль результата.

Форма и dtype - часть контракта числового расчета.

Как читать паспорт массива

Свойство или выражение	Для матрицы 3x4	Смысл
shape	(3, 4)	Длины осей
ndim / size	2 / 12	Число осей / элементов
dtype	Например, float64	Тип одной ячейки
a[1, 2]	Один скаляр	Вторая строка, третий столбец
a[:, 2]	Форма (3,)	Ось столбцов исчезла
a[:, 2:3]	Форма (3, 1)	Ось столбцов сохранена

Создаем матрицу и проверяем суммы

Пример Python

```
import numpy as np

grid = np.arange(1, 13).reshape(3, 4)
print(grid)
print(grid.sum(axis=0))
print(grid.sum(axis=1))
print(grid.shape, grid.ndim, grid.size)
```

Результат: axis=0: [15, 18, 21, 24]; axis=1: [10, 26, 42].

Создание

- arange не включает правую границу.
- reshape сохраняет число элементов.

Проверка

- axis=0: один результат на столбец.
- axis=1: один результат на строку.

Ось называет направление сворачивания

axis=0

- Сворачивается ось строк.
- Матрица (n, p) дает вектор (p,).
- Получаем показатель каждого признака.

axis=1

- Сворачивается ось столбцов.
- Матрица (n, p) дает вектор (n,).
- Получаем показатель каждого объекта.

Сначала назовите смысл осей, затем выбирайте axis и прогнозируйте форму результата до запуска кода.

Представление памяти и независимая копия

Представление, view

- Базовый срез обычно делит память.
- Изменение может затронуть исходник.
- Новое имя не означает новые данные.

Копия, copy

- Хранит независимые значения.
- Подходит для изменяемого эксперимента.
- Требует дополнительной памяти.

До изменения решите, какой объект обязан сохраниться, и создайте копию заранее.

Изменение среза: видимое последствие

Пример Python

```
base = np.arange(1, 10).reshape(3, 3)
view = base[:2, :2]
view[0, 0] = -1
independent = base[:2, :2].copy()
independent[0, 0] = 100
print(base[0, 0], independent[0, 0])
print(np.shares_memory(base, view))
```

Результат: -1 100; shares_memory(base, view) возвращает True.

Общая память

- Изменение view меняет base.
- Срез экономит копирование данных.

Независимость

- copy изолирует следующее присваивание.
- Проверяем несколько известных ячеек.

Маска: выбрать ячейки или целые строки

1

Назвать объект отбора

Выбираем отдельные ячейки, строки или столбцы.

2

Построить условие

Сравнения дают bool; условия соединяются через & или |.

3

Согласовать форму

Длина маски соответствует выбранной оси.

4

Проверить результат

Смотрим shape, число выбранных позиций и известный пример.

- grid[grid > 4] выбирает значения и дает один вектор.
- grid[grid[:, 0] >= 5, :] выбирает целые строки.

Broadcasting: совместимость форм

Формы	Совместимы?	Как интерпретировать
(3, 4) и число	Да	Одно число для каждой ячейки
(3, 4) и (4,)	Да	Одно значение на столбец
(3, 4) и (3, 1)	Да	Одно значение на строку
(3, 4) и (3,)	Нет	Последние размеры 4 и 3
(n, p) и (p,)	Да	Например, средние признаков

Совместимые формы еще не доказывают правильный смысл операции.

Центрирование по двум разным осям

Пример Python

```
column_means = grid.mean(axis=0)
centered = grid - column_means
row_means = grid.mean(axis=1, keepdims=True)
row_centered = grid - row_means
assert np.allclose(centered.mean(axis=0), 0)
assert np.allclose(row_centered.mean(axis=1), 0)
print(centered.shape, row_centered.shape)
```

Результат: Оба результата имеют форму (3, 4), но разный смысл.

По столбцам

- Каждый признак получает нулевое среднее.
- Вычитается вектор формы (4,).

По строкам

- Каждая строка получает нулевое среднее.
- Вычитается массив формы (3, 1).

Reshape, транспонирование и сортировка

Изменение формы

- reshape сохраняет общее число элементов.
- .T переставляет оси, а не просто меняет shape.
- flatten создает независимую плоскую копию.

Сохранение объектов

- sort(axis=0) сортирует колонки отдельно.
- argsort задает порядок по одному признаку.
- Один порядок применяют ко всем полям строки.

Проверяйте не только форму, но и происхождение значений.

Дисперсия и соглашение о знаменателе

var = sum((x - mean)2) / (n - ddof)**

ddof=0

- Знаменатель n: описание имеющегося набора.
- Это значение по умолчанию у NumPy.

ddof=1

- Знаменатель n-1: оценочное соглашение.
- Для сравнения библиотек задаем явно.

Для [2, 4, 6, 8]: mean=5; var(ddof=0)=5; var(ddof=1)=20/3.

std = sqrt(var); промежуточные значения не округляем.

Числовая матрица из pandas

Пример Python

```
from pathlib import Path
import pandas as pd

football = pd.read_csv(Path('data') / 'football.csv')
features = ['Acceleration', 'SprintSpeed', 'Stamina']
matrix = football[features].to_numpy(dtype=float, copy=True)
assert matrix.shape == (12897, 3)
assert np.isfinite(matrix).all()
corr = np.corrcoef(matrix, rowvar=False)
print(corr.shape)
```

Результат: Матрица признаков: (12897, 3); корреляции: (3, 3).

Контракт

- Порядок features сохраняется отдельно.
- copy=True изолирует последующие изменения.

Корреляция

- rowvar=False: признаки в столбцах.
- Линейная связь не устанавливает причину.

Типичные числовые ошибки

Ошибка	Почему опасна	Проверка
Перепутать axis	Ответит на другой вопрос	Ожидаемая форма и смысл
Изменить view	Испортит исходник	copy до эксперимента
Сортировать колонки отдельно	Разрушит записи	argsort и общий порядок
Путать * и @	Другая операция	Ручной пример и shape
Забыть ddof	Разные знаменатели	Явный параметр
Округлять слишком рано	Накопление погрешности	Округление только при выводе

Основные результаты лекции

1. ndarray задается значениями, формой и dtype.
2. axis указывает сворачиваемую ось; форму результата прогнозируем заранее.
3. Срез и копия по-разному связаны с памятью исходника.
4. Broadcasting согласует формы, но не проверяет смысл.
5. Статистики требуют явного ddof и содержательной интерпретации.

Контрольные вопросы и самостоятельная работа

1. Чем shape, ndim и size отличаются друг от друга?
2. Почему a[:, 1] и a[:, 1:2] дают разные формы?
3. Когда изменение выбранного фрагмента меняет исходник?
4. Как согласуются формы (n, p), (p,) и (n, 1)?
5. Почему корреляция не доказывает причинную связь?

Самостоятельная работа

Для матрицы 4x3 выполните центрирование по столбцам и по строкам. Запишите формы промежуточных массивов и проверьте нулевые средние.

Источники и документация

Документация

1. NumPy: основы

https://numpy.org/doc/stable/user/absolute_beginners.html

2. NumPy: broadcasting

https://numpy.org/doc/stable/user/basics.broadcasting.html

3. NumPy: копии и представления

https://numpy.org/doc/stable/user/basics.copies.html

Основа лекции

- Лекция 7 Numpy.pptx
- Лабораторная работа 7 Numpy.docx
- Лабораторная работа 7 с ответами.docx