ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ ДАННЫХ

Лекция 2

Индексация, фильтрация и преобразования

Оспан Ә.Ғ., PhD, доцент-исследователь

КазНУ, кафедра ИИ и Big Data

2026-2027 учебный год

Цель и содержание лекции

Цель и результаты обучения

- Различать выбор по меткам, положениям и условиям.
- Проверять границы, размеры подвыборок и все ничьи экстремума.
- Создавать признаки и сортировать записи без потери соответствий.

Содержание лекции

1. **loc и iloc**
2. **Булевы маски**
3. **Пороги**
4. **Экстремумы**
5. **Преобразования**

Условный вопрос состоит из нескольких действий

1

Задать совокупность

Порог определяется по полному файлу или объявленной группе.

2

Рассчитать порог

Один раз сохранить значение и его смысл.

3

Построить маску

Проверить границы и число выбранных строк.

4

Вычислить показатель

Усреднить нужный признак и описать ограничение вывода.

- Порядок операций является частью аналитического вопроса.
- Проверяем состав выборки до расчета итогового числа.

loc и iloc: метки и положения

Вопрос	Запись	Что выбирается
Метка строки и имя поля	df.loc[20, 'Age']	Одна ячейка
Вторая строка и первое поле	df.iloc[1, 0]	Положение в текущем порядке
Первые две строки	df.iloc[:2]	Позиции 0 и 1
Метки от 10 до 30	df.loc[10:30]	Обе границы при подходящем индексе
Одна строка таблицей	df.loc[[20], ['Age']]	DataFrame формы 1 на 1

Два пути к одной записи

Пример Python

```
import pandas as pd

demo = pd.DataFrame({
    'Name': ['Ada', 'Boris', 'Chen', 'Dina'],
    'Age': [19, 27, 22, 31],
    'Wage': [10, 40, 20, 30],
    'SprintSpeed': [80, 65, 75, 60]
}, index=[10, 20, 30, 40])
print(demo.loc[20, 'Name'])
print(demo.iloc[1, 0])
```

Результат: Boris; Boris

По метке

- 20 является значением индекса.
- Имя поля задает второй выбор.

По позиции

- 1 означает вторую строку.
- 0 означает первый столбец.

Булева маска: решение для каждой строки

Структура маски

- Series со значениями True и False.
- Индекс связан с исходными строками.
- sum дает число истинных условий.

Применение

- df.loc[mask, columns] задает строки и поля.
- Выбор не изменяет исходную таблицу.
- После отбора проверяем размер и значения.

Каждая строка должна получить логическое решение именно по объявленному правилу отбора.

Сочетаем условия: «и» отличается от «или»

Пример Python

```
young_fast = (demo['Age'] < 25) & (demo['SprintSpeed'] >= 75)
edge_age = (demo['Age'] < 20) | (demo['Age'] > 30)
print(demo.loc[young_fast, 'Name'].tolist())
print(demo.loc[edge_age, 'Name'].tolist())
print(young_fast.sum())
```

Результат: ['Ada', 'Chen']; ['Ada', 'Dina']; 2

Пересечение: &

- Оба условия должны быть истинны.
- Каждое сравнение заключаем в скобки.

Объединение: |

- Достаточно одного истинного условия.
- ~ задает поэлементное отрицание.

Диапазоны, категории и отрицание

Задача	Прием	Что уточнить
Внутри диапазона	s.between(20, 30)	Включенность границ
Одна из категорий	s.isin(['GK', 'ST'])	Точное написание категорий
Все, кроме совпадений	~mask	Смысл неизвестных значений
Ровно максимум	s.eq(s.max())	Сохранение всех ничьих

Граничную строку полезно проверить вручную до запуска полного расчета.

Среднее подвыборки зависит от ее состава

mean(selected) = sum(x_i) / n_selected

Порог

- Wage.mean() рассчитываем по полной таблице.
- Строгие группы исключают равных среднему.

Проверка

- Нужны численности выше, ниже и на пороге.
- Для пустой группы среднее не определено.

В demo: средний Wage = 25; SprintSpeed выше порога = 62.5, ниже = 77.5.

Повторяем условный расчет на football.csv

Пример Python

```
from pathlib import Path

football = pd.read_csv(Path('data') / 'football.csv')
threshold = football['Wage'].mean()
mask = football['Wage'] > threshold
selected = football.loc[
    mask, ['Name', 'Wage', 'SprintSpeed']
]
print('Строк:', len(selected))
print('Средний рейтинг:', selected['SprintSpeed'].mean())
```

Сначала отбор

- Порог получен по полному файлу.
- В результате остаются нужные поля.

Затем интерпретация

- Сообщаем численность и средний рейтинг.
- Сравнение не доказывает причину.

Экстремум: одна метка или все лидеры?

Одна запись

- idxmax возвращает одну метку.
- Подходит, если нужна одна строка по объявленному правилу.
- Сам по себе не перечисляет ничьи.

Все совпадения

- s.eq(s.max()) задает маску лидеров.
- Выбираем имена и проверяемое значение.
- Для группы максимум считаем внутри этой группы.

Экстремум определяется внутри выбранной совокупности, а все совпадения сохраняются.

Отношение двух средних: проверки до деления

1

Определить группы

Например, сначала Position == GK, затем максимумы навыков.

2

Сохранить ничьи

Включить все строки, равные нужному максимуму.

3

Проверить числа

Обе группы непусты; среднее знаменателя не равно нулю.

4

Разделить и описать

Показать оба средних, численности и порядок отношения.

- Пересечение групп допустимо, если оно следует из условий.
- Отсутствие результата нельзя маскировать заменой NaN на ноль.

Новый признак и сортировка целых записей

Пример Python

```
work = football[['Name', 'Age', 'Wage']].copy()
work['AgeCentered'] = work['Age'] - work['Age'].mean()
work['IsYoung'] = work['Age'] < 23
ranked = work.sort_values(
    ['Wage', 'Name'], ascending=[False, True]
)
print(ranked.head(3))
print(round(work['AgeCentered'].mean(), 10))
```

Результат: Среднее AgeCentered близко к 0.

Преобразование

- Копия отделяет рабочую таблицу.
- Новый столбец явно получает результат формулы.

Порядок

- sort_values переставляет целые строки.
- .iloc теперь относится к новому порядку.

Содержательный контроль выбранных показателей

Penalties

- Рейтинг навыка исполнения пенальти.
- Для голов нужны события и период наблюдения.

90% максимума

- Порог равен 0.9 * max по указанной группе.
- Это другое условие, чем рейтинг > 90.

Wage и Value

- Единицы и период не подтверждены.
- Сравнение относится к историческому файлу.

Что проверить перед сохранением ответа

Логика

- Скобки вокруг каждого сравнения.
- Маска относится к текущей таблице.

Границы

- Строгость порога записана явно.
- Пустые группы и все ничьи учтены.

Изменения

- Одно .loc для условного присваивания.
- Округление выполняется в конце.

В отчете должны быть видны условие, численность и смысл вычисленного показателя.

Пять правил условного анализа

1. loc использует метки, iloc использует положения текущего порядка.
2. Составная маска строится через &, | и отдельные скобки сравнений.
3. Порог и экстремум вычисляются на заранее указанной совокупности.
4. Пустота, ничьи и нулевой знаменатель проверяются до вывода.
5. Новый признак и сортировка должны сохранять связь всех полей записи.

Проверьте отбор и интерпретацию

1. Почему loc[20] и iloc[20] могут означать разные действия?
2. Чем группа выше среднего отличается от группы не ниже среднего?
3. Как сохранить всех обладателей максимального значения?
4. Что проверить до отношения двух групповых средних?
5. Почему сумма Penalties не дает число голов?

Самостоятельная работа

Создайте таблицу с двумя равными максимумами и пустой после фильтрации группой. Получите всех лидеров и понятное сообщение вместо среднего пустой группы.

Источники по доступу и преобразованиям

Документация

1. **pandas: индексация и отбор**
   https://pandas.pydata.org/docs/user_guide/indexing.html
2. **pandas: базовые операции**
   https://pandas.pydata.org/docs/user_guide/basics.html
3. **pandas: пропущенные значения**
   https://pandas.pydata.org/docs/user_guide/missing_data.html

Основа лекции

- Лекция 2.pdf
- Лабораторная работа 2.docx
- Лабораторная работа 2 с ответами.docx