СӘТБАЕВ УНИВЕРСИТЕТІ SATBAYEV UNIVERSITY

Основы искусственного интеллекта. Лекция 4. Работа с данными: предобработка и визуализация. Методы предобработки данных, визуализация данных с Matplotlib и Seaborn

Преподаватель: **Қасенхан Арай Мейрамбайқызы, доктор PhD, ассоциированный профессор кафедры «Программная инженерия»**

a.kassenkhan@satbayev.university

Содержание

1. Зачем нужна предобработка данных
2. Обработка пропущенных значений
3. Масштабирование данных
4. Кодирование категориальных признаков
5. Визуализация данных
6. Подготовка данных для ML

По завершению урока Вы будете знать:

1. **Зачем нужна предобработка данных**
2. **Как находить и обрабатывать пропущенные значения**
3. **Чем отличаются нормализация и стандартизация**
4. **Как преобразовывать категориальные данные в числовые**
5. **Как исследовать данные с помощью Matplotlib и Seaborn**
6. **Как подготовить данные для модели машинного обучения**

Почему данные нужно предобрабатывать?

Реальные данные могут содержать:

- пропуски;
- выбросы;
- разные масштабы;
- категориальные значения;
- ошибки и шум.

Качество модели зависит не только от алгоритма, но и от качества данных.



Начинаем с исследования данных

```
import pandas as pd

df = pd.read_csv("students.csv")

print(df.head())
print(df.shape)
df.info()
print(df.isnull().sum())
```

Перед обработкой проверяем:

- **Что находится в данных?**
- **Есть ли пропуски?**
- **Какие типы данных?**
- **Сколько строк и столбцов?**

Сначала исследуем → потом изменяем данные

Обработка пропущенных значений

Удалить

```
df = df.dropna()
```

Заполнить

```
df["age"] = df["age"].fillna(
df["age"].median() )
```

Числовые данные → среднее / медиана
Категориальные → мода (Mode)

Не существует одного универсального способа обработки пропусков.

Масштабирование данных

Нормализация — MinMaxScaler

Приведение значений к диапазону [0, 1]

```
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()

X_scaled = scaler.fit_transform(X)
```

10 → 0.0
50 → 0.5
90 → 1.0

Стандартизация — StandardScaler

Среднее ≈ 0, стандартное отклонение ≈ 1

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

X_scaled = scaler.fit_transform(X)
```

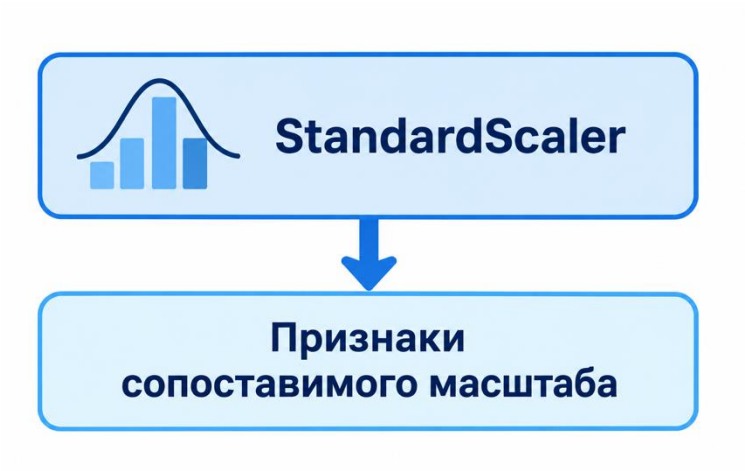
Масштабирование особенно важно для алгоритмов, чувствительных к масштабу признаков.

Зачем нужен единый масштаб?

Признак	Значение
Age	20
Salary	450 000
Experience	3

← огромное значение

Salary может доминировать только из-за масштаба



Кодирование категориальных данных

city
Almaty
Astana
Almaty
Shymkent

→ Модель ожидает числовое представление →

One-Hot Encoding

pd.get_dummies(df["city"])

Результат:

Almaty	Astana	Shymkent
1	0	0
0	1	0
1	0	0
0	0	1

Категория → числовое представление

Зачем визуализировать данные?

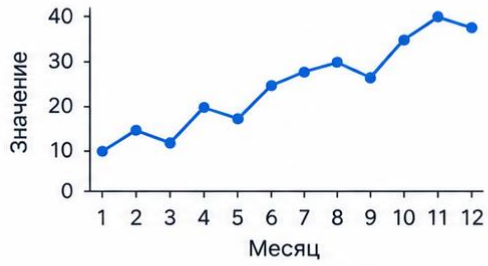


Таблица → График → Закономерность

Визуализация помогает увидеть то, что сложно заметить в таблице.

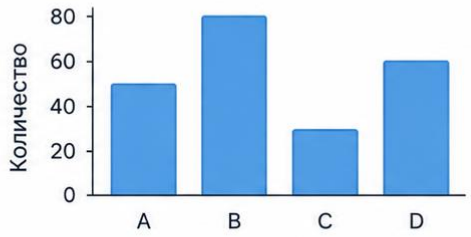
Matplotlib: основные графики

1 Line plot
Тренды



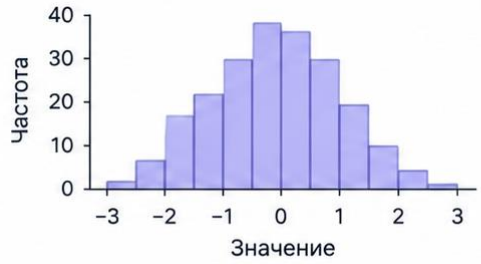

```
plt.plot(x, y)
```

2 Bar chart
Сравнение категорий



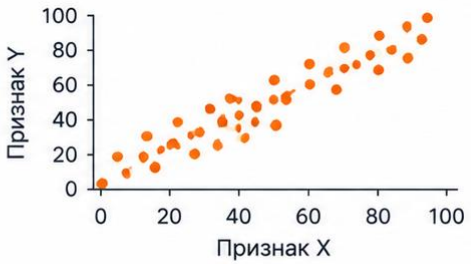

```
plt.bar(x, y)
```

3 Histogram
Распределение




```
plt.hist(data)
```

4 Scatter plot
Связь двух признаков



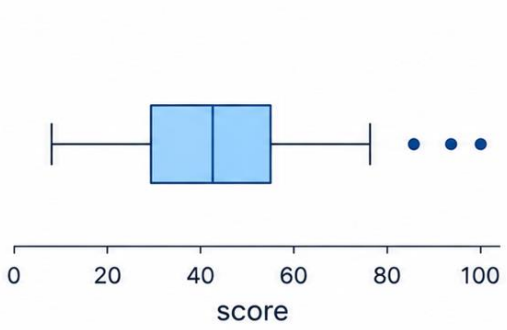

```
plt.scatter(x, y)
```

Выбор графика зависит от вопроса, который мы задаём данным.

Seaborn: анализ данных

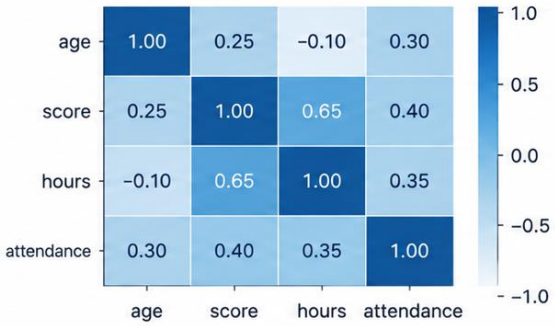
Seaborn работает поверх Matplotlib и удобно интегрируется с Pandas DataFrame.

Boxplot
Выбросы



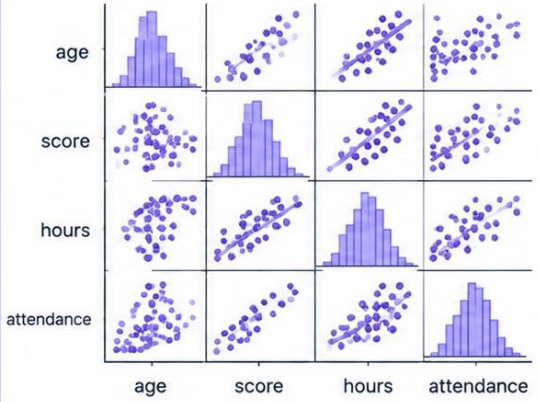

```
sns.boxplot(x=df["score"])
```

Heatmap
Корреляции




```
sns.heatmap(
  df.corr(numeric_only=True),
  annot=True
)
```

Pairplot
Связи между несколькими признаками



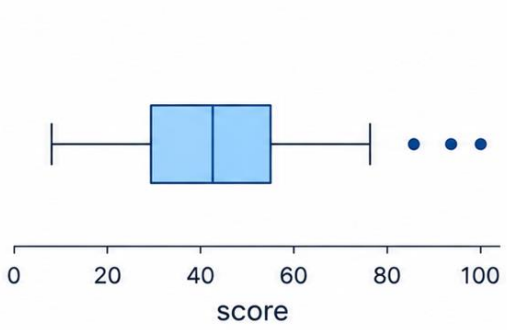

```
sns.pairplot(df)
```

Matplotlib → базовое построение графиков
Seaborn → удобный анализ статистических зависимостей

Seaborn: анализ данных

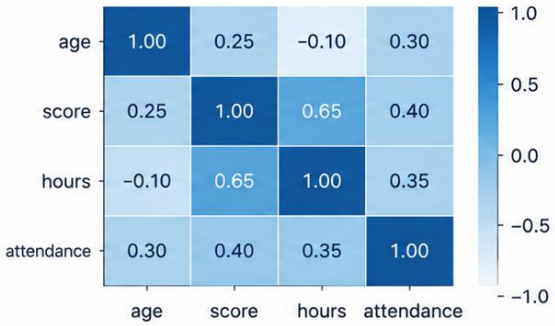
Seaborn работает поверх Matplotlib и удобно интегрируется с Pandas DataFrame.

Boxplot
Выбросы



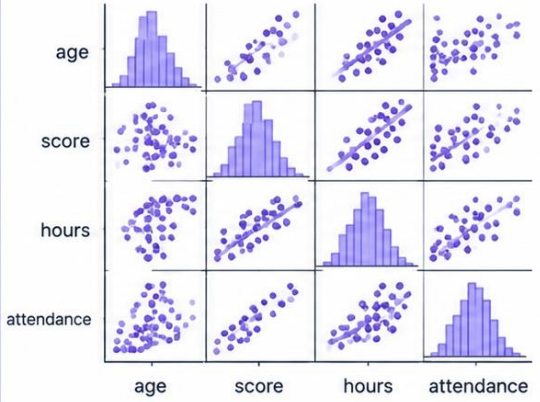

```
sns.boxplot(x=df["score"])
```

Heatmap
Корреляции




```
sns.heatmap(
  df.corr(numeric_only=True),
  annot=True
)
```

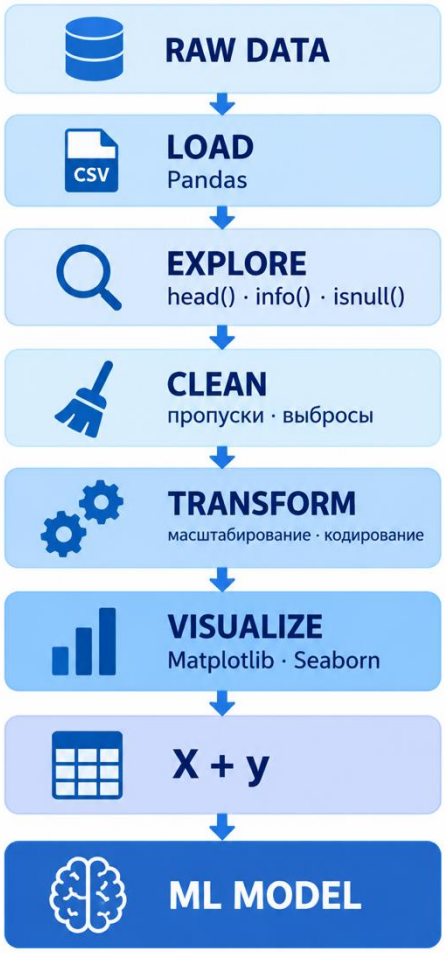
Pairplot
Связи между несколькими признаками




```
sns.pairplot(df)
```

Matplotlib → базовое построение графиков
Seaborn → удобный анализ статистических зависимостей

От сырых данных к ML-модели



Предобработка превращает исходные данные в форму, пригодную для машинного обучения.

Заключение

- **Исследовали качество исходных данных;**
- **Обработали пропущенные значения;**
- **Рассмотрели нормализацию и стандартизацию;**
- **Преобразовали категориальные признаки;**
- **Познакомились с Matplotlib и Seaborn;**
- **Подготовили данные для дальнейшего машинного обучения.**



Хорошая модель начинается с качественно подготовленных данных.