Корреляция и линейная регрессия

Введение

- **Корреляция:** Помогает найти связи между переменными.
- **Регрессия:** Позволяет строить модели для прогнозирования.

Пример: Анализ влияния рекламы на продажи.

Корреляция

Что такое корреляция?

Корреляция — это статистическая мера, показывающая, как две переменные связаны между собой.

Примеры:

- Положительная корреляция: рост доходов и потребления.
- Отрицательная корреляция: снижение температуры и продаж мороженого.

Виды корреляции

Положительная: обе переменные растут (пример: доход и расходы).

Отрицательная: одна переменная растёт, другая падает (пример: цена и спрос).

Нулевая: переменные не связаны (пример: размер обуви и оценка по математике).

Графики для каждого вида корреляции

Диаграммы рассеяния:

- ***Положительная корреляция:*** точки располагаются по возрастающей линии.
- ***Отрицательная корреляция:*** точки располагаются по убывающей линии.
- ***Нулевая корреляция:*** точки распределены хаотично.

Методы измерения корреляции

Коэффициент Пирсона: для линейных зависимостей.

Коэффициент Спирмена: для ранговых данных.

Диаграмма рассеяния для визуализации.

Коэффициент Пирсона

Коэффициент корреляции Пирсона — это статистическая мера линейной связи между двумя переменными.

Диапазон значений:

- $r=1 \rightarrow$ **Сильная положительная корреляция**
- $r=-1 \rightarrow$ **Сильная отрицательная корреляция**
- $r=0 \rightarrow$ **Отсутствие связи**

Формула коэффициента Пирсона

$$r = \frac{\sum (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum (x_i - \overline{x})^2} \cdot \sqrt{\sum (y_i - \overline{y})^2}}$$

Пример расчёта коэффициента Пирсона

Номер	x (Доход)	y (Расход)
1	10	15
2	20	25
3	30	35
4	40	45

Вычисление в Python

```
import numpy as np
import scipy.stats as stats

x = np.array([10, 20, 30, 40])
y = np.array([15, 25, 35, 45])

r, p_value = stats.pearsonr(x, y)

print(f"Коэффициент корреляции Пирсона: {r:.2f}")
```

Корреляционная матрица

```
import pandas as pd

df = pd.DataFrame({'Доход': [10, 20, 30, 40],
'Расход': [15, 25, 35, 45]})

corr_matrix = df.corr()
print(corr_matrix)
```

Визуализация корреляции

```
import seaborn as sns
import matplotlib.pyplot as plt

sns.heatmap(df.corr(), annot=True, cmap="coolwarm")
plt.show()
```

Интерпретация результатов

$r > 0.7$ → **Сильная связь**

$0.3 < r < 0.7$ → **Средняя связь**

$r < 0.3$ → **Слабая связь**

$r \approx 0$ → **Нет связи**

Ложные корреляции

Примеры ложной корреляции:

- Количество пиратов и изменение климата.
- Продажи мороженого и число нападений акул.

Коэффициент Спирмена

Коэффициент корреляции Спирмена — это статистическая мера, показывающая силу **монотонной** связи между двумя переменными.

Используется, когда данные **не удовлетворяют условиям Пирсона** (например, имеют ранговую шкалу).

Диапазон значений

rs=1 → **Сильная положительная связь**

rs=−1 → **Сильная отрицательная связь**

rs=0 → **Связь отсутствует**

В чем отличие от Пирсона?

Пирсон измеряет **линейную** связь.

Спирмен измеряет **монотонную** связь.

Используется для **ранговых данных** (позиции в соревнованиях, оценки по шкале, субъективные мнения).

Формула коэффициента Спирмена

$$r_s = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$$

Пример ранговых данных

Ученик	Оценка за тест	Ранг теста	Оценка за проект	Ранг проекта
Аня	85	2	90	1
Иван	70	4	75	3
Мария	90	1	85	2
Петр	60	5	70	4
Елена	75	3	65	5

Вычисление в Python

```
import pandas as pd
from scipy.stats import spearmanr

df = pd.DataFrame({
    'Тест': [85, 70, 90, 60, 75],
    'Проект': [90, 75, 85, 70, 65]
})

corr, _ = spearmanr(df['Тест'], df['Проект'])
print(f'Коэффициент Спирмена: {corr:.2f}')
```

Интерпретация результатов

rs>0.7→ **Сильная связь**

0.3<rs<0.7 → **Средняя связь**

rs<0.3 → **Слабая связь**

rs≈0 → **Связи нет**

Применение в реальной жизни

Анализ клиентской лояльности: связь между оценками пользователей и повторными покупками.

Исследования: анализ предпочтений, психологические тесты.

Ошибки интерпретации корреляции

Корреляция ≠ причинно-следственная связь!

Высокая корреляция не означает, что одно вызывает другое.

Пример: "Чем больше людей тонет в бассейнах, тем больше продаётся мороженого."

Примеры ложной корреляции

Пример 1: Рост продаж мороженого и число утоплений.

- Общий фактор: жаркая погода.

Пример 2: Рост потребления органической пищи и рост числа диагнозов аутизма.

- Совпадение трендов, но нет реальной связи.

Как визуализировать корреляцию?

Визуализация корреляции помогает **увидеть** взаимосвязь между переменными, выявить выбросы и определить, линейная ли эта связь. Давайте рассмотрим основные способы.

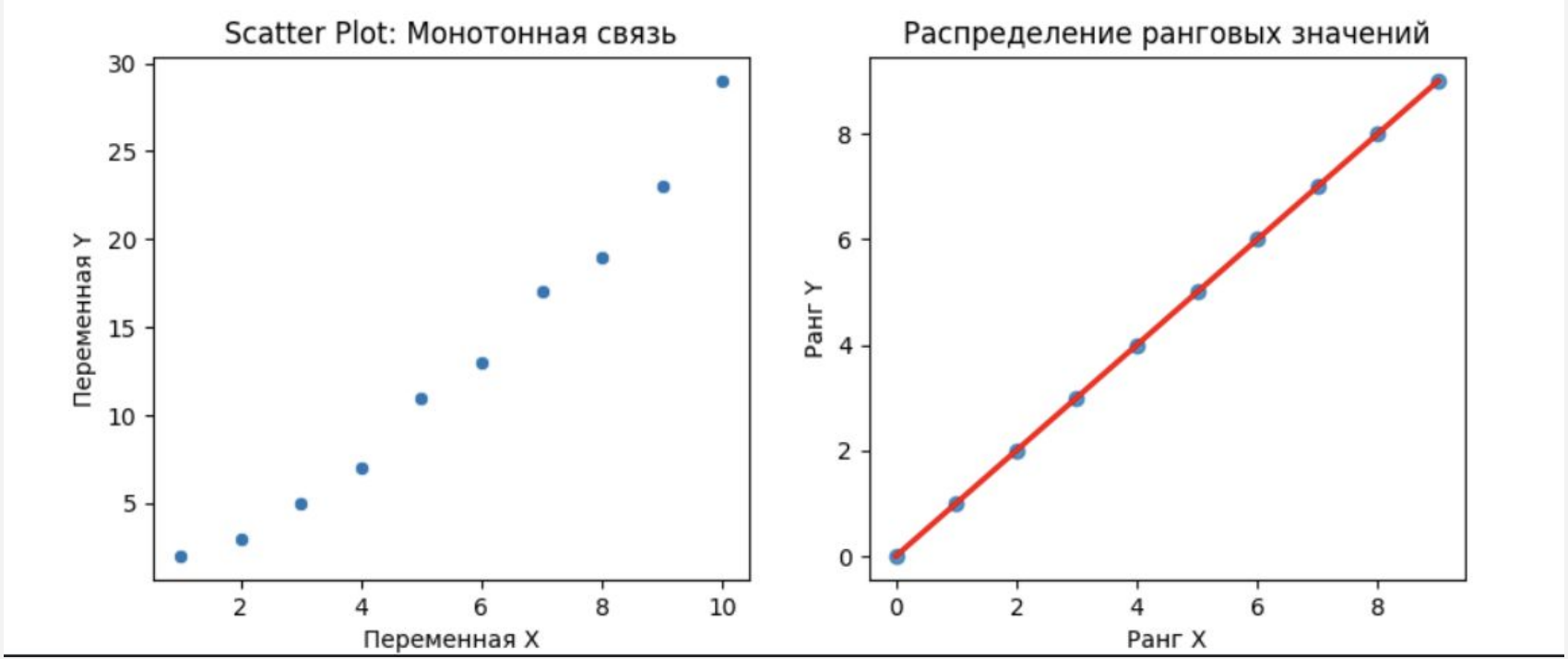
Методы визуализации корреляции

Диаграмма рассеяния (scatter plot) → лучше всего показывает корреляцию.

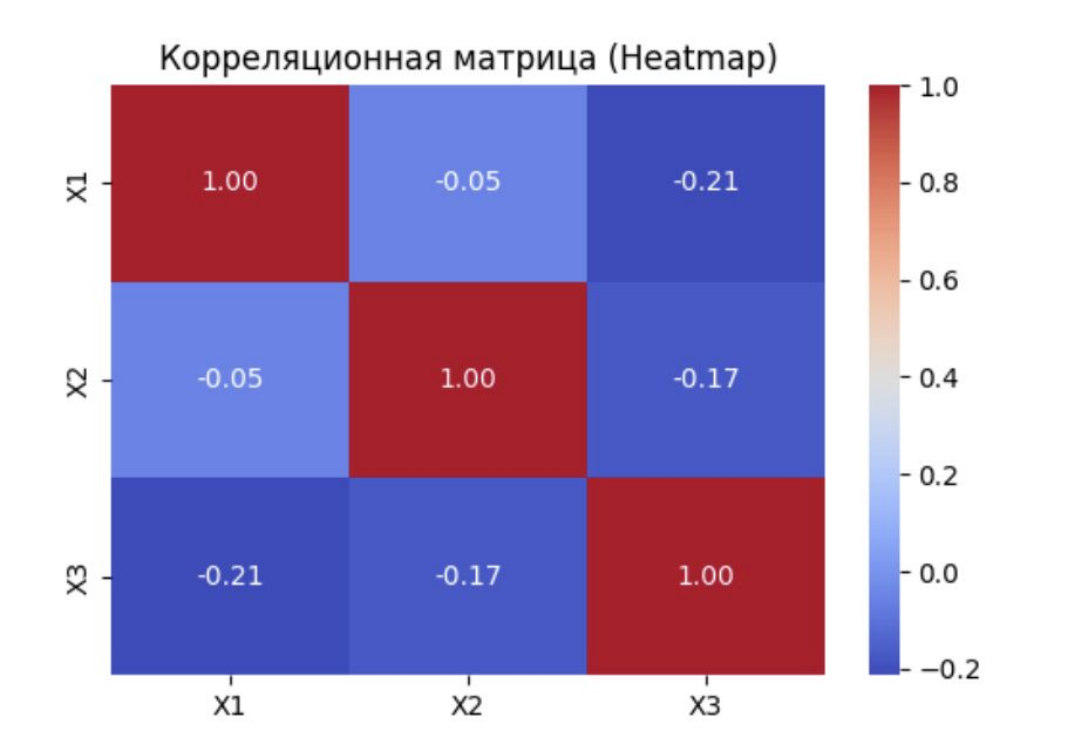
Корреляционная матрица (heatmap) → показывает связи между многими переменными.

Линия регрессии → помогает увидеть тренд данных.

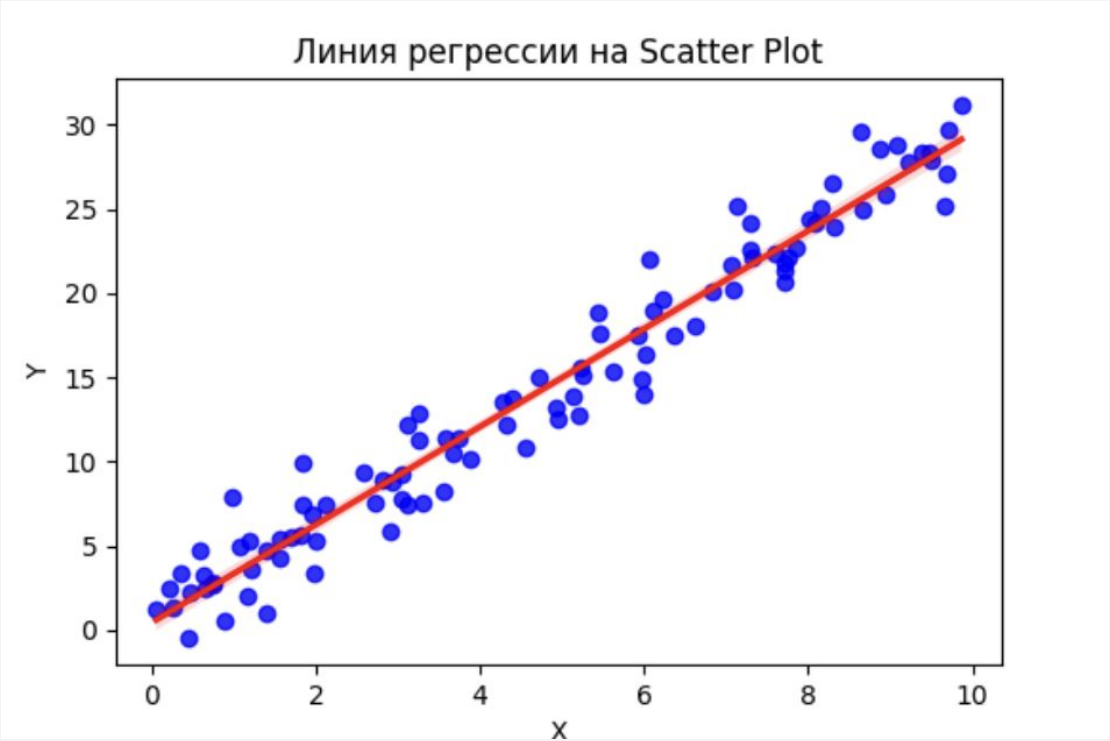
Визуализация связи (Scatter Plot & Rank Plot)



Корреляционная матрица (Heatmap)



Линия регрессии (Регрессионный тренд)



Вывод

Scatter plot помогает увидеть связь между двумя переменными.

Heatmap показывает корреляцию между многими параметрами.

Линия регрессии даёт представление о тренде.

Как избежать ошибок?

✅ Проверять наличие третьей переменной (скрытого фактора).

✅ Строить графики (scatter plot, heatmap).

✅ Анализировать тренды во времени.

✅ Использовать экспериментальные методы для проверки причинности.

Линейная регрессия

Что такое линейная регрессия?

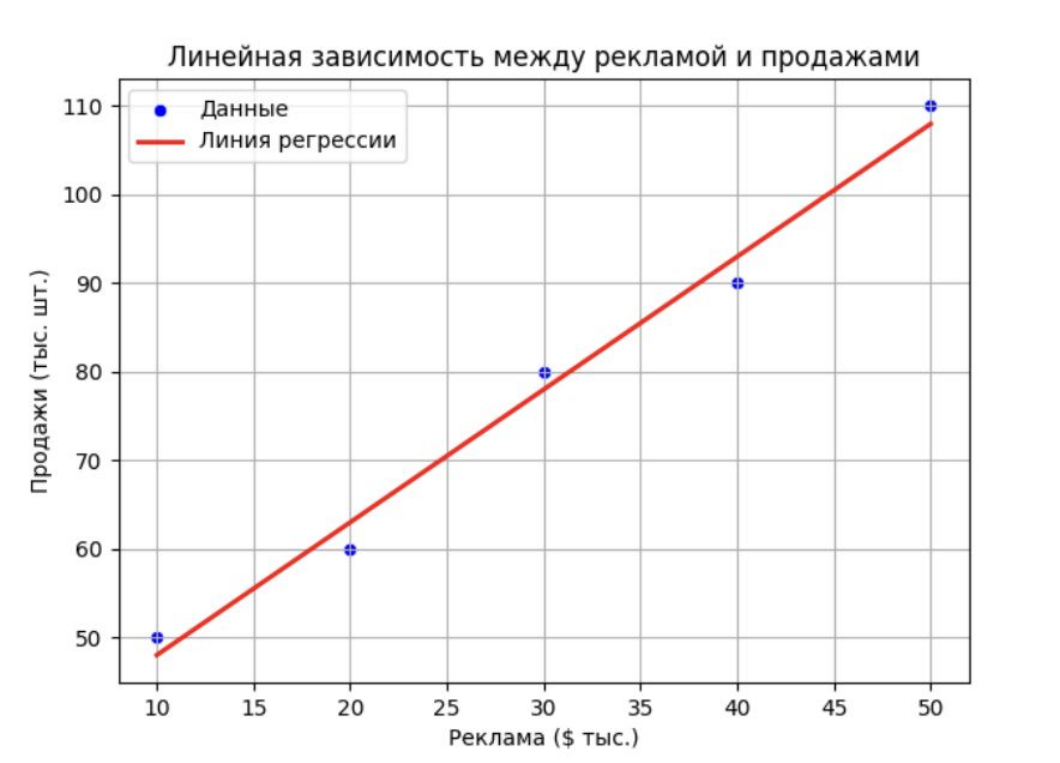
Линейная регрессия — это статистический метод, который **моделирует зависимость одной переменной (Y) от другой (X)** с помощью **прямой линии**.

$$y = b_0 + b_1 \cdot x$$

Где применяется линейная регрессия?

- **Экономика**: прогноз цен, анализ спроса.
- **Медицина**: зависимость дозировки лекарства и выздоровления.
- **Финансы**: прогноз доходов.
- **Маркетинг**: оценка влияния рекламы.

Визуализация линейной зависимости



Метод наименьших квадратов

Идея: ищем такую прямую, чтобы минимизировать ошибки.

Ошибка: разница между реальным значением и предсказанным.

Формула ошибки:

$$\sum (y_i - \hat{y}_i)^2$$

Реальный пример с расчётом

Реклама ($ тыс.)	Продажи (тыс. шт.)
10	50
20	60
30	80
40	90

Линейная регрессия в Python

```
import numpy as np

X = np.array([10, 20, 30, 40])
Y = np.array([50, 60, 80, 90])

b1 = np.cov(X, Y, bias=True)[0][1] / np.var(X)
b0 = np.mean(Y) - b1 * np.mean(X)

print(f"Уравнение регрессии: Y = {b0:.2f} +
{b1:.2f}X")
```

Линейная регрессия с Sklearn

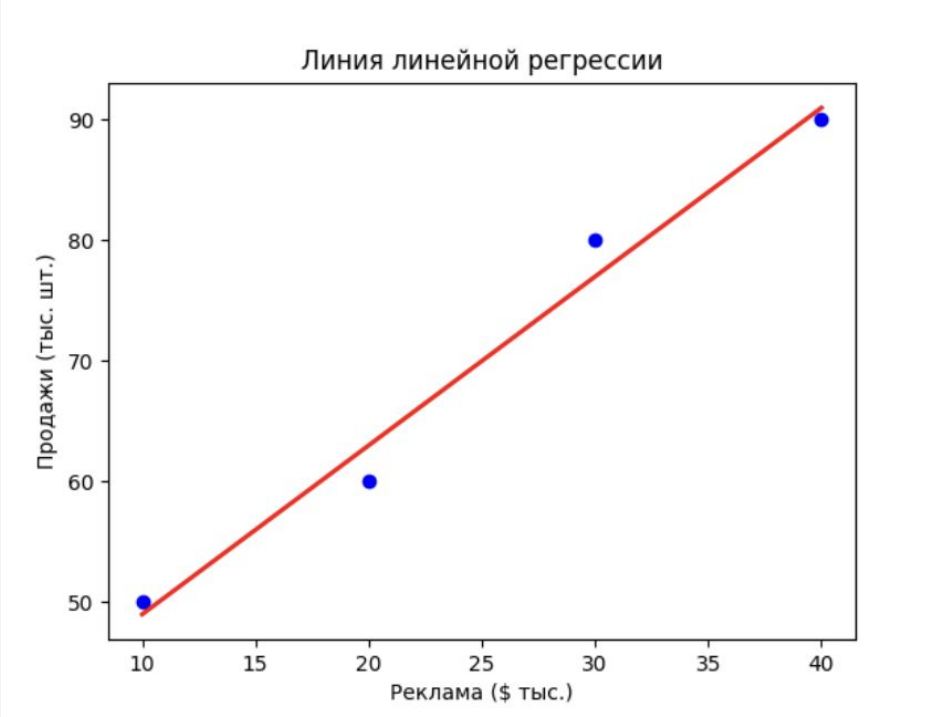
```
from sklearn.linear_model import LinearRegression
import numpy as np

X = np.array([10, 20, 30, 40]).reshape(-1, 1)
Y = np.array([50, 60, 80, 90])

model = LinearRegression()
model.fit(X, Y)

print(f"Коэффициенты: b0={model.intercept_:.2f},
b1={model.coef_[0]:.2f}")
```

График линии регрессии



Интерпретация модели

b0 — начальная точка (продажи без рекламы).

b1— насколько увеличиваются продажи при росте X.

Если b1>0, значит, зависимость **положительная**.

Когда линейная регрессия НЕ работает?

Если зависимость **нелинейная** (например, экспоненциальный рост).

Если в данных **много выбросов**.

Если **категориальные переменные** не закодированы правильно.

Оценка качества модели линейной регрессии

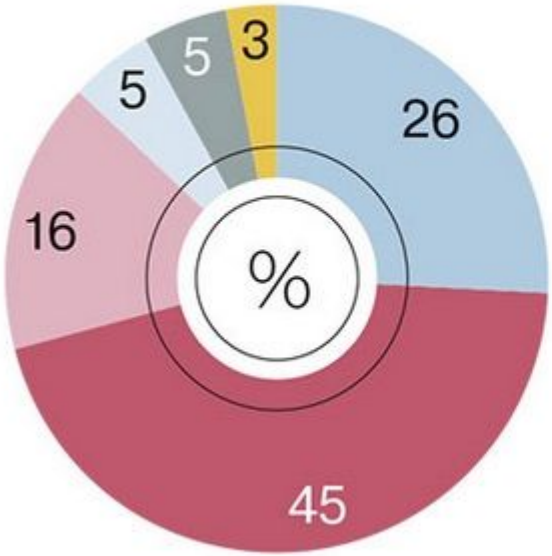
Почему важно оценивать модель?

📌 **Ключевые вопросы:**

✔ Насколько точно модель предсказывает?

✔ Какие ошибки она допускает?

✔ Можно ли её улучшить?



Коэффициент детерминации R^2

R^2=1 → идеальная модель.

R^2=0 → модель бесполезна.

R^2<0 → модель хуже случайного угадывания.

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \overline{y})^2}$$

Среднеквадратичная ошибка (MSE)

✔ Показывает, насколько в среднем предсказания модели отклоняются от реальных значений.

✔ Чем меньше MSE, тем лучше модель.

$$MSE = \frac{1}{n}\sum(y_i - \hat{y}_i)^2$$

Как улучшить качество модели?

✅ Добавить **новые признаки** в данные.

✅ Исключить **выбросы**.

✅ Использовать **полиномиальную регрессию**, если зависимость нелинейная.

✅ Увеличить количество **обучающих данных**.