Д. Серікбаев атындағы
ШҚТУ

Основы искусственного интеллекта

Линейная регрессия.

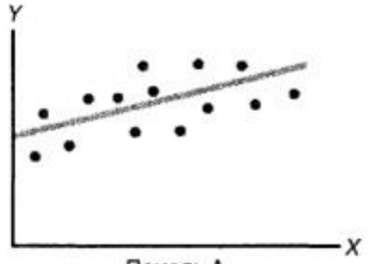
План лекции

- Линейная регрессия — один из самых простых методов машинного обучения.
- Настройка базового проекта по Data Science с использованием Python и основных библиотек для анализа данных и машинного обучения в среде Google Colab.

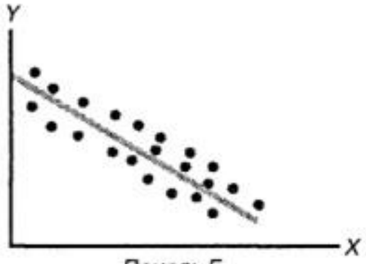
Линейная регрессия

- Линейная регрессия используется для прогнозирования непрерывных переменных.
- 📌 **Пример:** предсказание цены дома по площади.
- 📌 **Основное уравнение:** $y = w_0 + w_1 x + \varepsilon$
- **Где**
  - y – предсказываемая переменная (целевая переменная)
  - x – объясняющая переменная (фича)
  - w_0 – свободный член (intercept)
  - w_1 – коэффициент (наклон линии)
  - ε – ошибка

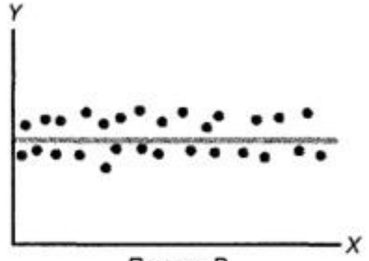
Виды зависимостей



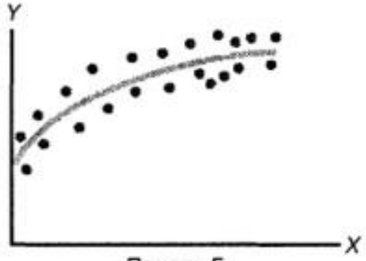
Панель А
Положительная линейная зависимость



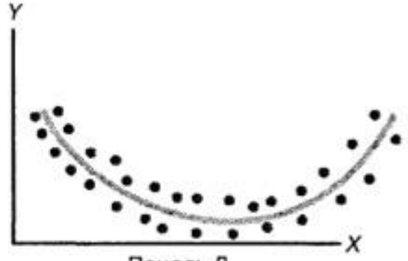
Панель Б
Отрицательная линейная зависимость



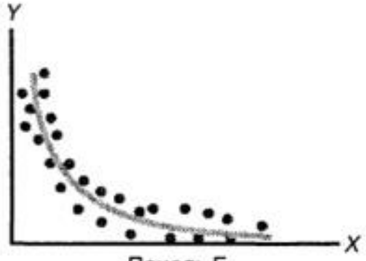
Панель В
Переменные *X* и *Y* не зависят друг от друга



Панель Г
Положительная криволинейная зависимость



Панель Д
U-образная криволинейная зависимость



Панель Е
Отрицательная криволинейная зависимость

Линейная регрессия

- Для того, чтобы спрогнозировать значение одной переменной на основе значения другой, или нескольких других переменных используется **линейный регрессионный анализ.**

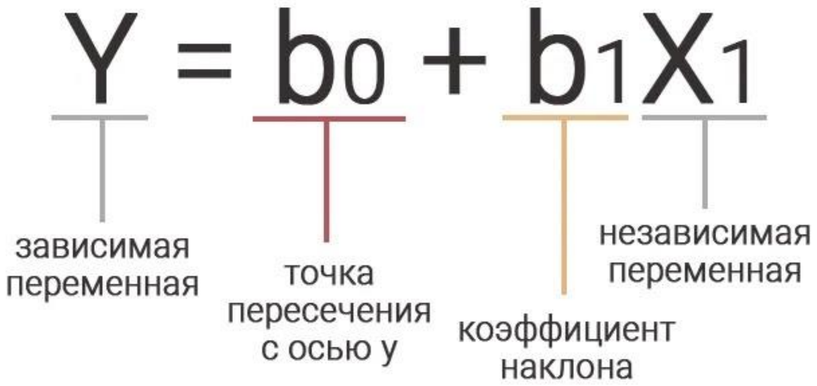
- Переменная, которую мы хотим предсказать, называется **зависимой переменной** (или регрессант).

- Переменные, которые мы используем для предсказания значения зависимой переменной, называются **независимыми переменными** (или предикторы/регрессоры).

- **Построить регрессию** – это значит подобрать такую линию (точнее, функцию), которая как можно точнее описывает данные.

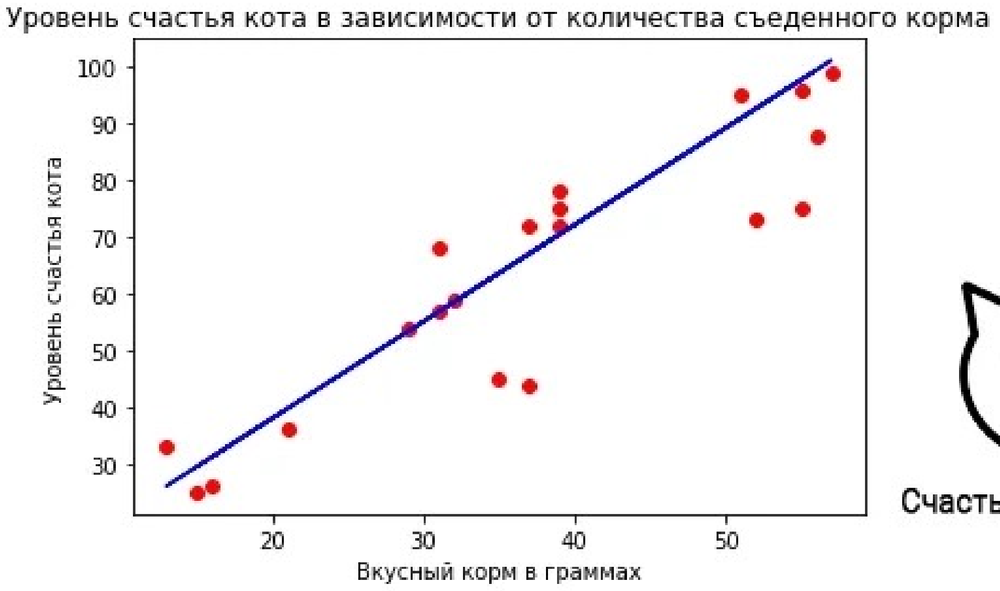
Парная (простая) линейная регрессия

- $Y = b0 + b1*X1$,
- где $X1$ — независимая переменная, а Y — зависимая переменная.
- Наклон линии равен $b1$, а $b0$ — точка пересечения с осью y.



Пример парной регрессии

- y – уровень счастья кота
- x – количество съеденного корма



Счастье кота = 3,89 + 1,7*кол-во корма (гр)

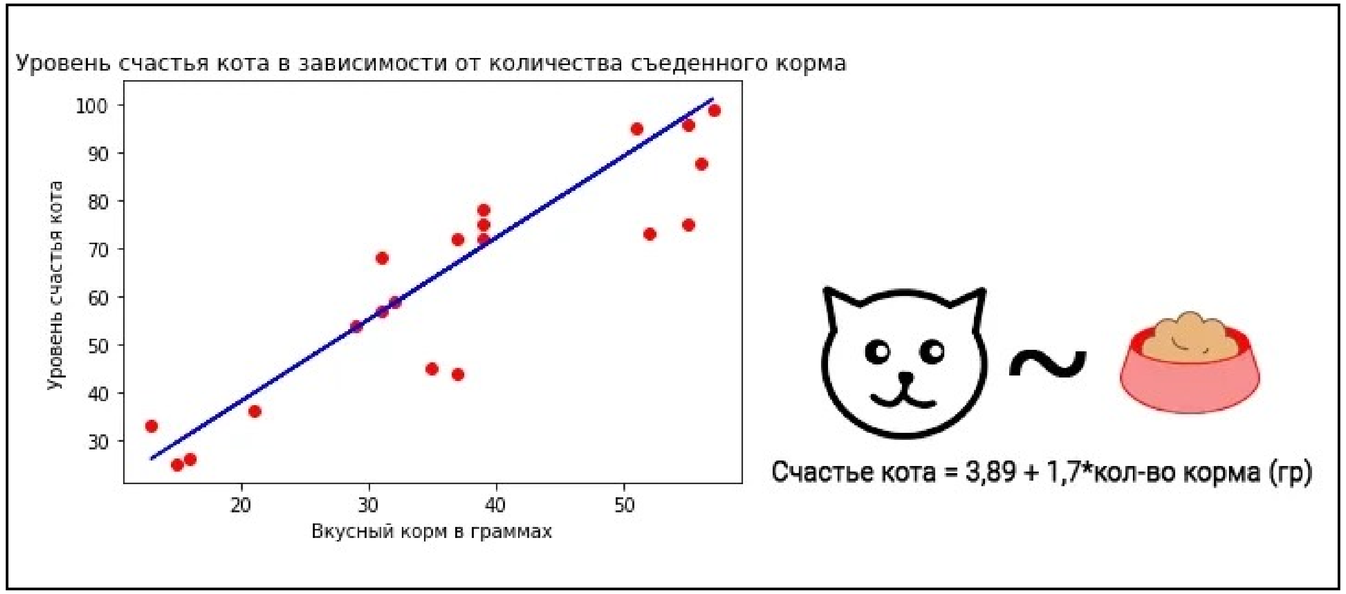
Пример парной регрессии

Уравнение регрессии

Уровень счастья кота = 3,89 + 1,7*кол-во вкусного корма (гр)

b0 = 3,89 - точка пересечения оси y (если бы количество корма было 0, то прогнозируемый уровень счастья кота был бы 3,89).

b1 = 1,7 – коэффициент наклона (если увеличивать количество корма на 1 грамм, то уровень счастья кота увеличится на 1,7).



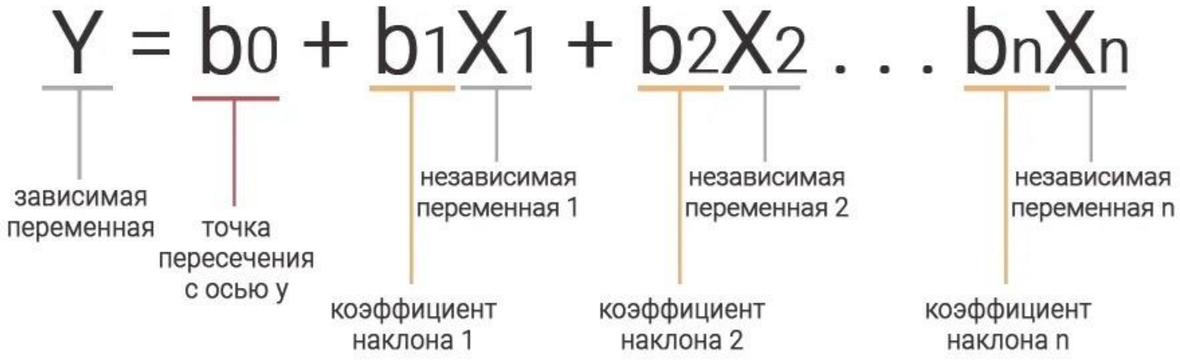
Множественная линейная регрессия

Уравнение множественной линейной регрессии:

$Y = b0 + b1X1 + b2X2 + \ldots + bnXn$,

где Xn — независимые переменные, а Y — зависимая переменная.

bn – коэффициент наклона для Xn , а $b0$ — точка пересечения с осью y.



Множественная линейная регрессия

Пример

- y – уровень счастья кота
- x1 – количество съеденного корма
- x2 – количество поглаживаний

Счастье кота = -5,13 + 1,55*кол-во корма (гр) + 1,77*кол-во поглаживаний

Если увеличивать количество корма на 1 грамм, то уровень счастья кота увеличится на 1,55, а если увеличивать количество поглаживаний на 1, то уровень счастья кота увеличится 1,77.

Предположения линейной регрессии

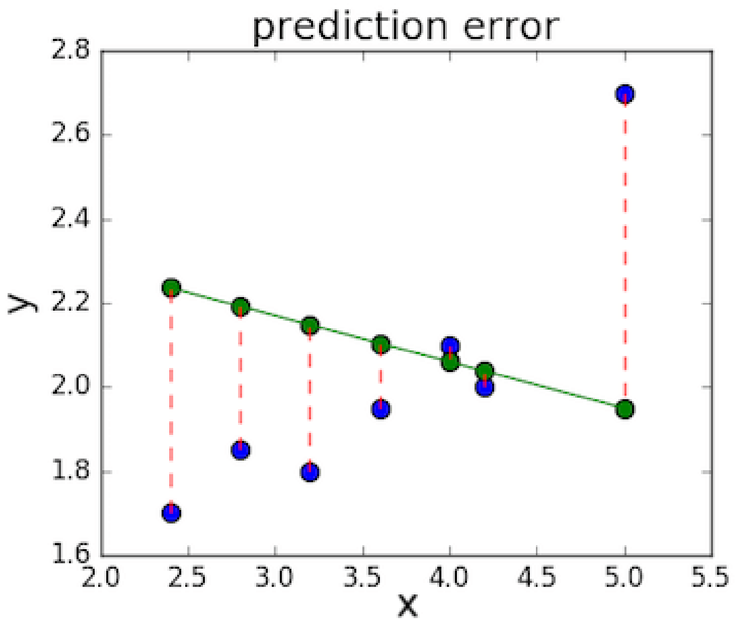
- Переменные должны быть непрерывными.
- Между переменными должна существовать линейная связь.
- Наличие гомоскедастичности. Гомоскедастичность – свойство, означающее постоянство дисперсии на протяжении всей линии регрессии.
- Наблюдения независимы друг от друга
- Данные не должны иметь значительных выбросов.

Применения линейной регрессии

- Производственные компании используют регрессионный анализ для оценки вероятности выхода из строя деталей оборудования.
- Медицинские исследователи используют регрессионные модели для прогноза вероятности заболевания у пациентов.
- Финансовые компании используют регрессионный анализ для оценки финансовых рисков, оценки платежеспособности клиентов.
- Маркетинговые компании оценивают потенциал клиентов на будущие периоды, владея некоторой информацией о клиенте: пол, возраст, доход.

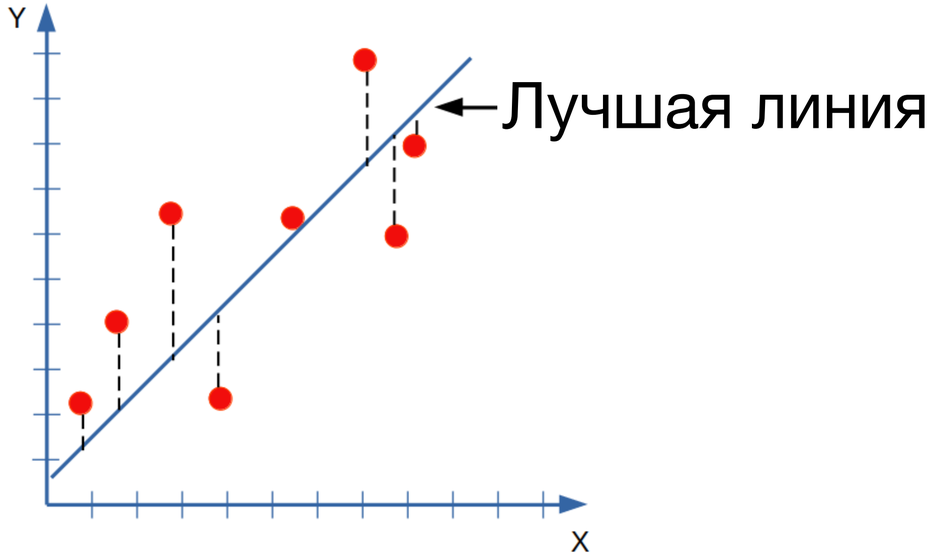
Как подобрать лучшую линию регрессии?

- Провести прямую линию через все точки на графике не получится, если они расположены в хаотичном порядке.
- Поэтому с помощью линейной регрессии определяется оптимальный вариант расположения этой прямой.
- Некоторые точки все равно останутся на расстоянии, но оно должно быть минимальным.
- Расчет этого минимального расстояния от прямой до каждой точки называется **функцией потерь**.



Как подобрать лучшую линию регрессии?

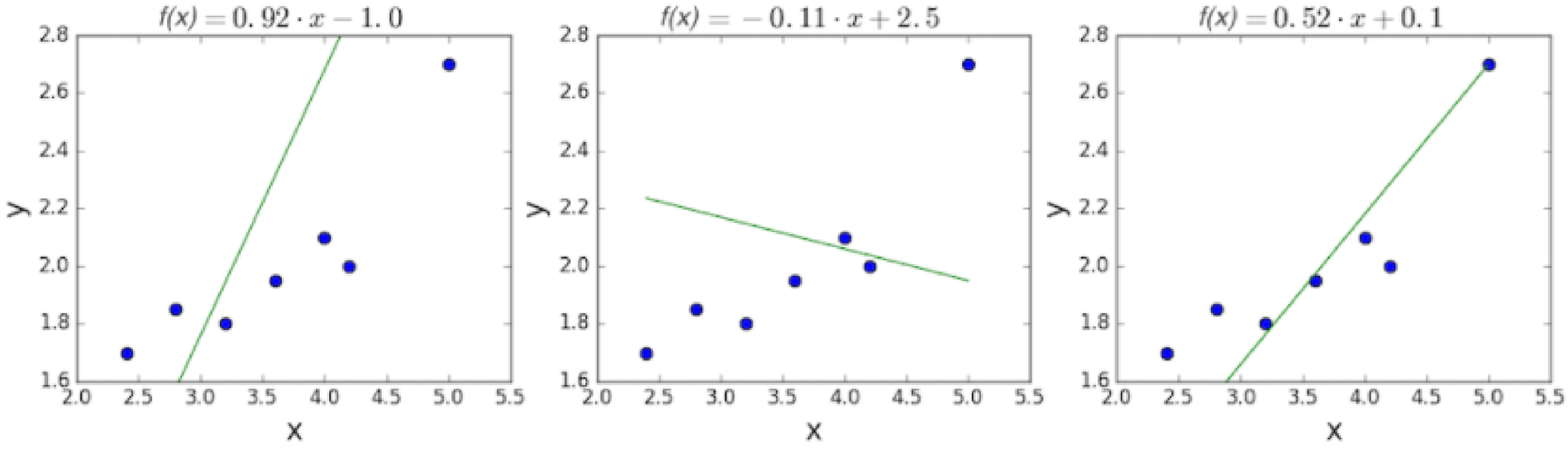
- Наиболее распространенный метод подбора линии регрессии - **метод наименьших квадратов (МНК)**.
- МНК вычисляет наиболее подходящую линию для наблюдаемых данных путем минимизации суммы квадратов вертикальных отклонений от каждой точки данных до линии.



Как подобрать лучшую линию регрессии?

MSE - функция потерь

- Средняя квадратическая ошибка (MSE - Mean Squared Error)



MSE = 0.17 MSE = 0.08 MSE = 0.02

Google Colab
Какие шаги необходимо сделать далее?

Этап 1

1. **Настройка окружения**
   - Google Colab: язык прогарммирования Python + зависимости (библиотеки)
2. **Загрузка и подготовка данных**
   - Загрузили набор данных fetch_california_housing - дома в калифорнии из библиотеки sklearn.datasets
3. **Предобработка данных**
   - Проверили есть ли пустые значения в наборе и выполнили визулизацию зависимости между двумя переметрами

Google Colab
Какие шаги необходимо сделать далее?

Этап 2

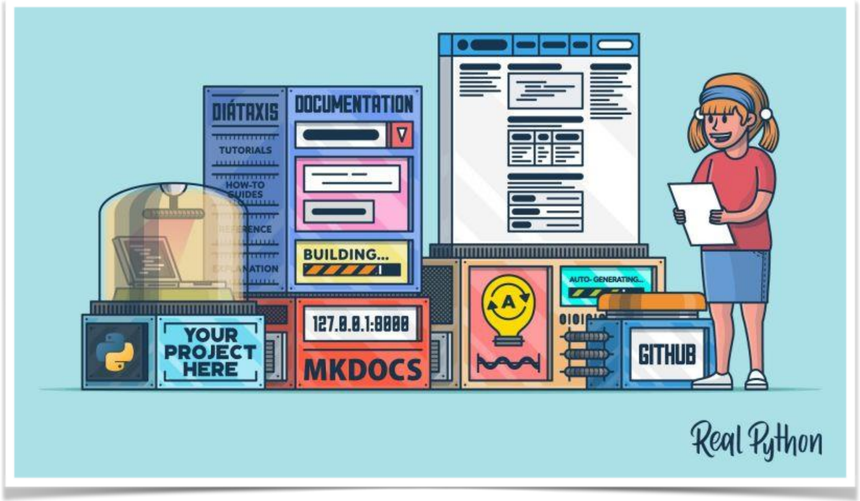
Реализация линейной регресии

1. Разделение данных
2. Обучение модели
3. Оценка модели
4. Визуализация результатов

Этап 1

Настройка окружения

**В Google Colab уже установлена необходимая версия Python



Этап 1

Загрузка и подготовка данных

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.datasets import fetch_california_housing

# Загрузка данных
data = fetch_california_housing()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['Price'] = data.target

# Просмотр первых строк
df.head()
```

Этап 1

Предобработка данных

- Проверим пропущенные значения и основные статистики:

```
print(df.isnull().sum())  # Проверка пропущенных значений
print(df.describe())  # Основные статистики
```

Этап 1

Визуализация данных

- Визуализируем зависимость цены от одной из характеристик:

```
sns.scatterplot(x=df['MedInc'], y=df['Price'])

plt.xlabel('Медианный доход')

plt.ylabel('Цена недвижимости')

plt.show()
```

Google Colab
Какие шаги необходимо сделать далее?

Этап 2

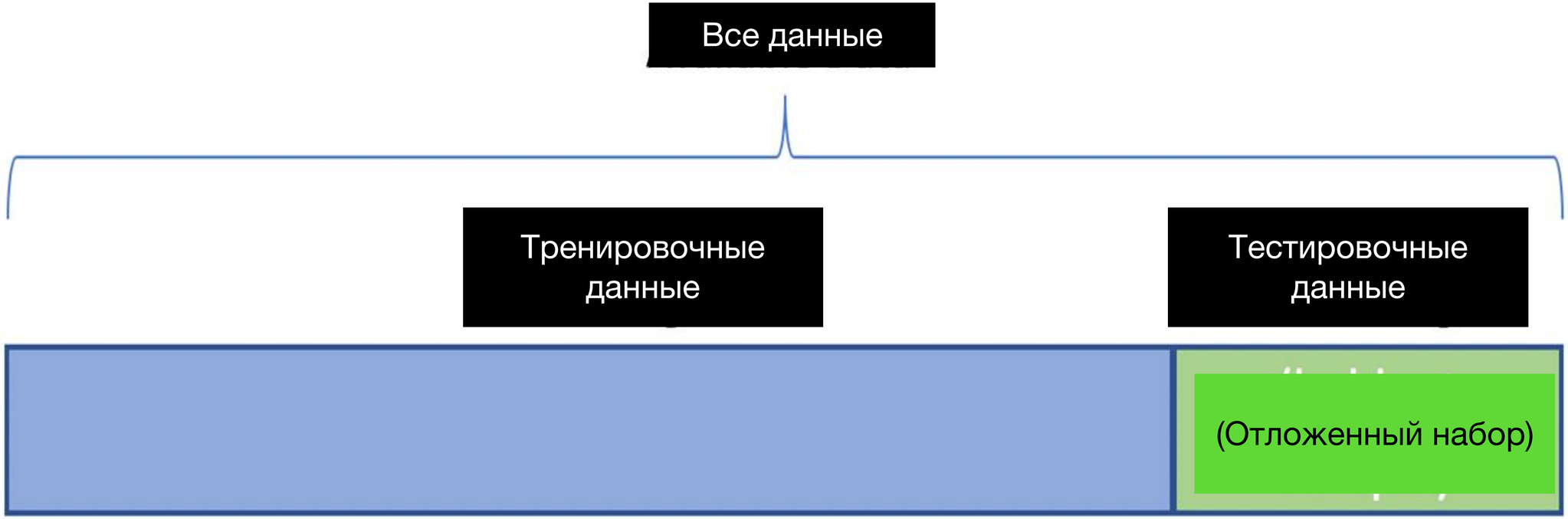
Реализация линейной регресии

1. Разделение данных
2. Обучение модели
3. Оценка модели
4. Визуализация результатов

Этап 2

Разделение данных

Train-test split



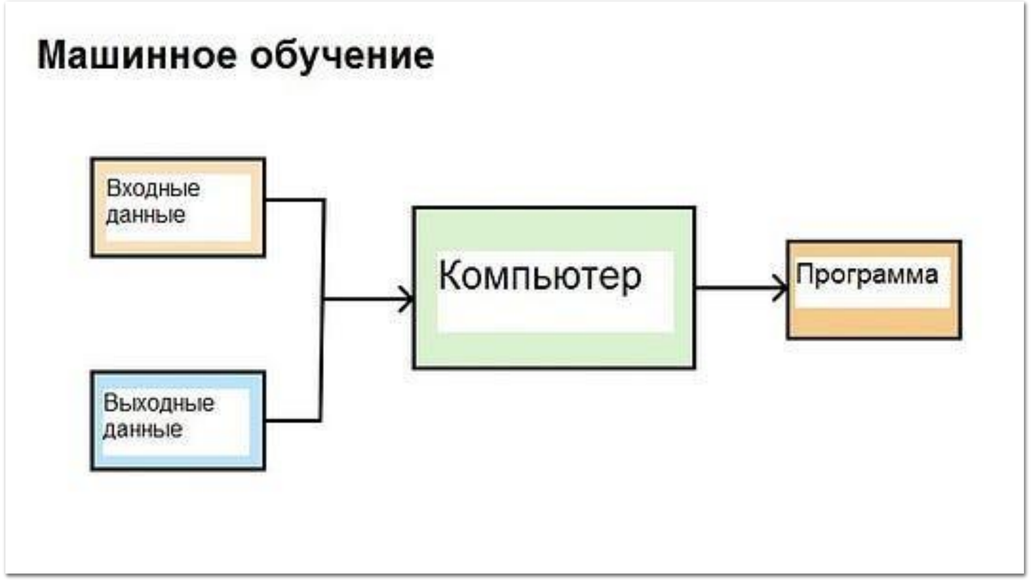
Этап 2

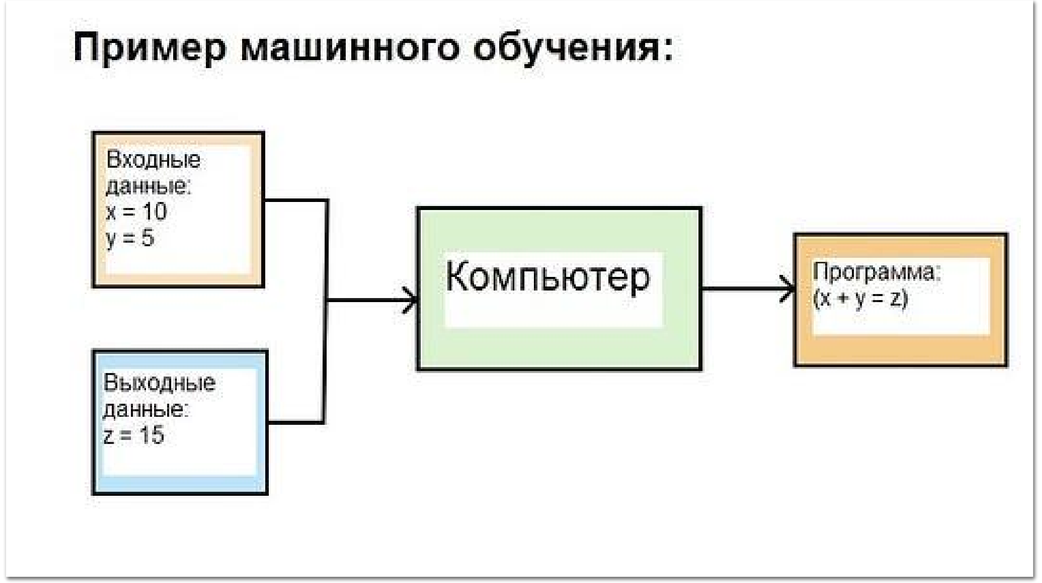
Разделение данных

```
from sklearn.model_selection import train_test_split


X = df[['MedInc']]

y = df['Price']


X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
```

Этап 2

Машинное обучение/Обучение модели





Этап 2

Обучение модели

```
from sklearn.linear_model import LinearRegression

model = LinearRegression()

model.fit(X_train, y_train)
```

Этап 2

Оценка модели

```
from sklearn.metrics import mean_squared_error, r2_score


y_pred = model.predict(X_test)

mse = mean_squared_error(y_test, y_pred)

r2 = r2_score(y_test, y_pred)


print(f'MSE: {mse}')

print(f'R^2: {r2}')
```

Этап 2

Визуализация результатов

- plt.scatter(X_test, y_test, color='blue', label='Реальные значения')
- plt.plot(X_test, y_pred, color='red', linewidth=2, label='Линейная регрессия')
- plt.xlabel('Медианный доход')
- plt.ylabel('Цена недвижимости')
- plt.legend()
- plt.show()

Спасибо за внимание!