

Повторение

ЗАНЯТИЕ 2

1. Что из перечисленного наиболее точно описывает обучение с учителем?

- a) Алгоритм заранее знает структуру данных и распределение классов
- b) Модель учится выявлять закономерности без разметки данных
- c) Модель строится на обучающих данных, где известны входы и ожидаемые выходы
- d) Модель самостоятельно группирует данные по признакам

Ответ: c

2. Что отличает обучение без учителя от обучения с учителем?

- a) Используются только числовые признаки
- b) Модель не знает правильных ответов и ищет структуру в данных
- c) Алгоритм обучается по паре "вопрос – ответ"
- d) Признаки заранее стандартизируются

Ответ: b

3. Чем отличается классификация от кластеризации?

- a) Классификация работает с числами, кластеризация — с текстами
- b) В классификации используются нейронные сети, в кластеризации — нет
- c) Классификация использует известные метки классов, кластеризация — нет
- d) Классификация строит регрессионную прямую

Ответ: c

4. Какой из примеров — задача регрессии?

- a) Определение категории новости: "спорт", "политика", "наука"
- b) Предсказание температуры воздуха по дате
- c) Распознавание цифр на картинке
- d) Разделение клиентов на группы по поведению

Ответ: b

5. Метод k-ближайших соседей (k-NN) делает следующее:

- a) Делит данные на k кластеров
- b) Ищет линейную зависимость между признаками
- c) Определяет класс объекта по классам ближайших точек
- d) Создаёт нейронную сеть из соседей

Ответ: c

6. Что показывает коэффициент наклона (k) в уравнении линейной регрессии $y=kx+by$?

- a) Количество классов
- b) Оценку ошибки модели
- c) Скорость изменения выходной переменной при изменении входной
- d) Точку пересечения с осью X

Ответ: c

7. Когда линейная регрессия будет давать плохие результаты?

- a) Когда между переменными есть чёткая линейная связь
- b) Когда данные случайны и нет зависимости
- c) Когда точек слишком много
- d) Когда все признаки числовые

Ответ: b

8. Что означает “ошибка” в контексте модели линейной регрессии?

- a) Количество неправильных классификаций
- b) Расстояние от предсказанного значения до реального
- c) Количество точек в датасете
- d) Количество признаков в модели

Ответ: b

9. Какая метрика лучше всего подходит для оценки качества линейной регрессии?

- a) F1-мера
- b) Среднеквадратичная ошибка (MSE)
- c) Точность (Accuracy)
- d) Количество соседей

Ответ: b

10. Какая из метрик чаще всего применяется в задачах классификации?

- a) MSE (Mean Squared Error)
- b) R^2 (коэффициент детерминации)
- c) Accuracy (доля правильных ответов)
- d) MAE (средняя абсолютная ошибка)

Ответ: c

11. Что происходит в процессе k-fold кросс-валидации?

- a) Модель тестируется только на последних 10% данных
- b) Модель обучается и тестируется k раз на разных разбиениях данных
- c) Все данные используются только для тестирования
- d) Модель обучается один раз, а потом настраивается вручную

Ответ: b

12. Что позволяет достичь кросс-валидация?

- a) Увеличить точность модели за счёт добавления новых признаков
- b) Проверить обобщающую способность модели на разных подвыборках
- c) Сократить время обучения
- d) Понизить количество классов

Ответ: b

13. Для чего может быть полезен метод k -ближайших соседей (k -NN)?

- a) Предсказание стоимости квартиры
- b) Поиск схожих товаров в интернет-магазине
- c) Группировка статей по темам без меток
- d) Создание сложной логики на основе линейной зависимости

Ответ: b

14. Когда линейная регрессия может работать некорректно?

- a) Когда все данные числовые
- b) Когда между признаками и результатом — нелинейная зависимость
- c) Когда у нас только один признак
- d) Когда обучающая выборка большая

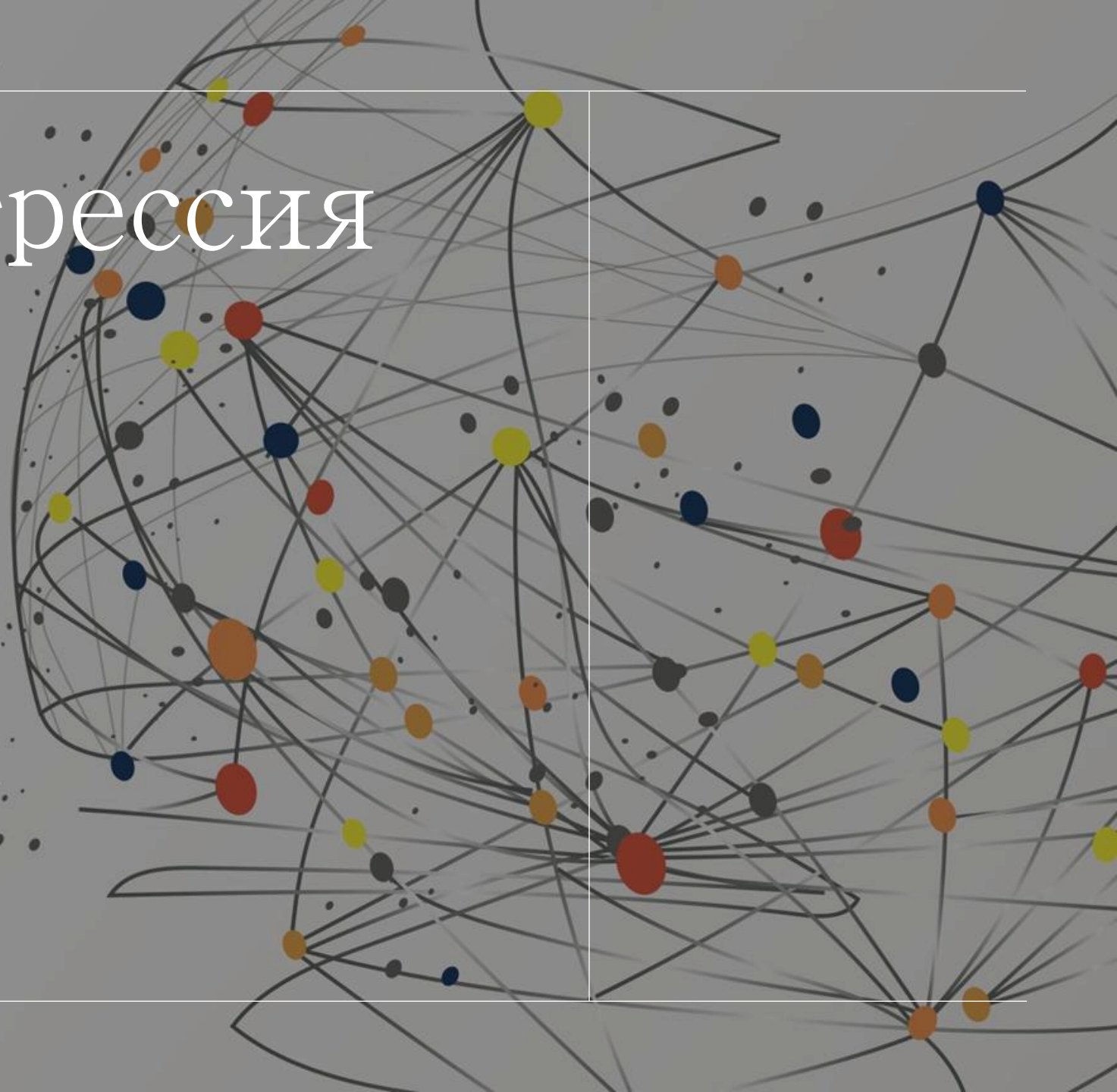
Ответ: b

15. Что делает алгоритм классификации?

- a) Находит числа, ближайšie к нулю
- b) Разбивает данные на кластеры
- c) Предсказывает категорию или класс для нового объекта
- d) Создаёт нейросеть из признаков

Ответ: c

Линейная регрессия

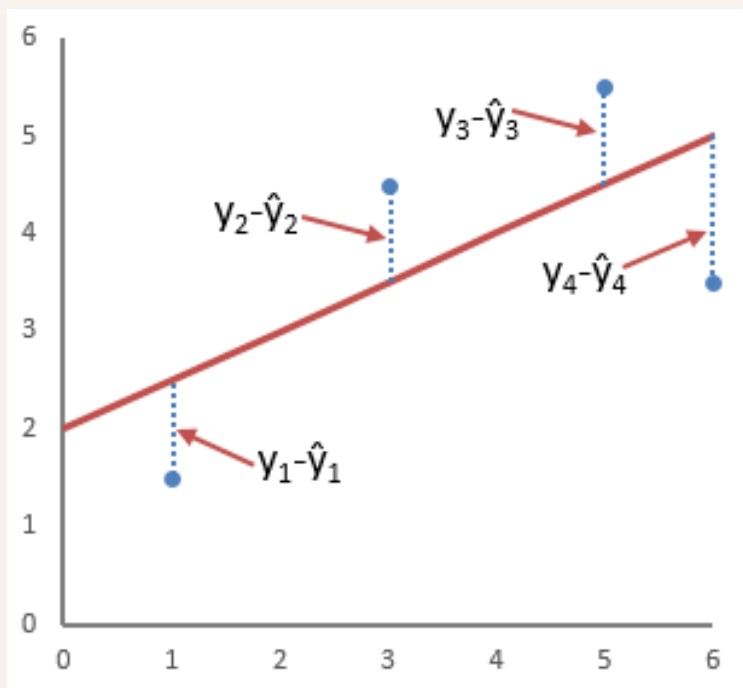


Задача

- У нас есть 10 наблюдений зависимости времени изучения (в часах) и оценки (от 1 до 10). Данные сильно разбросаны. Предсказать оценку для ученика, который учился 5.5 часа.

Время (часы)	Оценка
1.0	6.0
2.0	5.0
3.0	7.5
4.0	4.5
5.0	8.0
6.0	5.5
7.0	9.0
8.0	7.0
9.0	6.5
10.0	10.0

Решение



$$a = \frac{n \sum(xy) - \sum x \sum y}{n \sum(x^2) - (\sum x)^2}$$

$$b = \frac{\sum y - a \sum x}{n}$$

Решение

```
# Функция для расчёта коэффициентов линейной регрессии вручную
def linear_regression(x, y):
    n = len(x)
    x_mean = sum(x) / n
    y_mean = sum(y) / n

    numerator = sum((xi - x_mean)*(yi - y_mean) for xi, yi in zip(x, y))
    denominator = sum((xi - x_mean)**2 for xi in x)

    k = numerator / denominator
    b = y_mean - k * x_mean
    return k, b

k, b = linear_regression(x, y)
print(f"Уравнение регрессии: y = {k:.2f} * x + {b:.2f}")

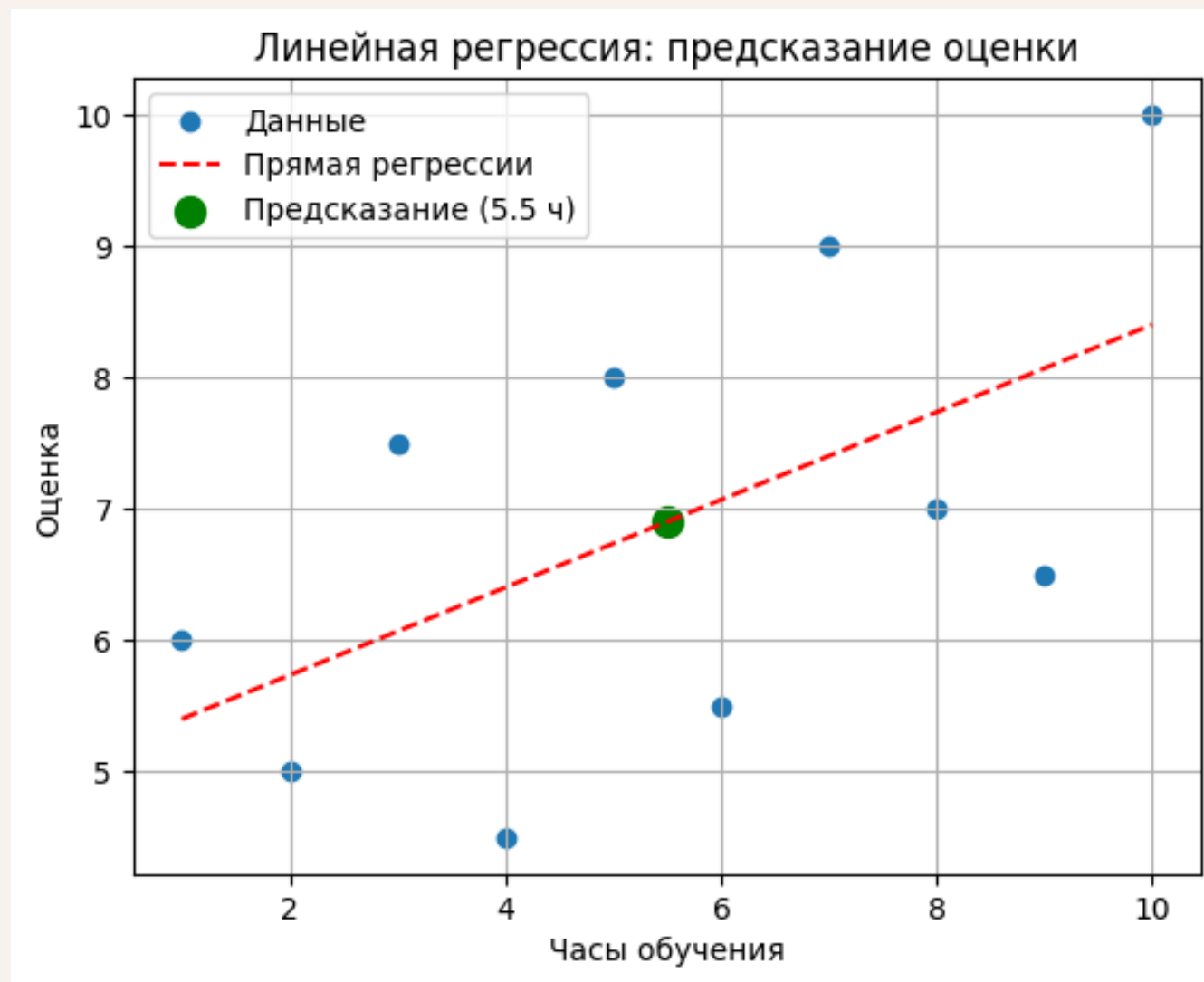
x_new = 5.5
y_pred = k * x_new + b
print(f"Предсказанная оценка для {x_new} часов: {y_pred:.2f}")

# График
plt.scatter(x, y, label='Данные')
plt.plot([min(x), max(x)], [k*min(x)+b, k*max(x)+b], 'r--', label='Прямая регрессии')
plt.scatter(x_new, y_pred, c='green', label='Предсказание (5.5 ч)', s=100)

plt.xlabel("Часы обучения")
plt.ylabel("Оценка")
plt.title("Линейная регрессия: предсказание оценки")
plt.legend()
plt.grid(True)
plt.show()
```

Уравнение регрессии: $y = 0.33 * x + 5.07$
Предсказанная оценка для 5.5 часов: 6.90

Решение



Решение

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

# Данные
X = np.array([[1], [2], [3], [4], [5], [6], [7], [8], [9], [10]])
y = np.array([6.0, 5.0, 7.5, 4.5, 8.0, 5.5, 9.0, 7.0, 6.5, 10.0])

# Модель
model = LinearRegression()
model.fit(X, y)



# Предсказание
hours = np.array([[5.5]])
predicted_score = model.predict(hours)
print("Предсказанная оценка для 5.5 часов:", round(predicted_score[0], 2))
```

Предсказанная оценка для 5.5 часов: 6.9

Метод k- ближайших соседей

ПОВТОРЕНИЕ.
ЗАНЯТИЕ 2

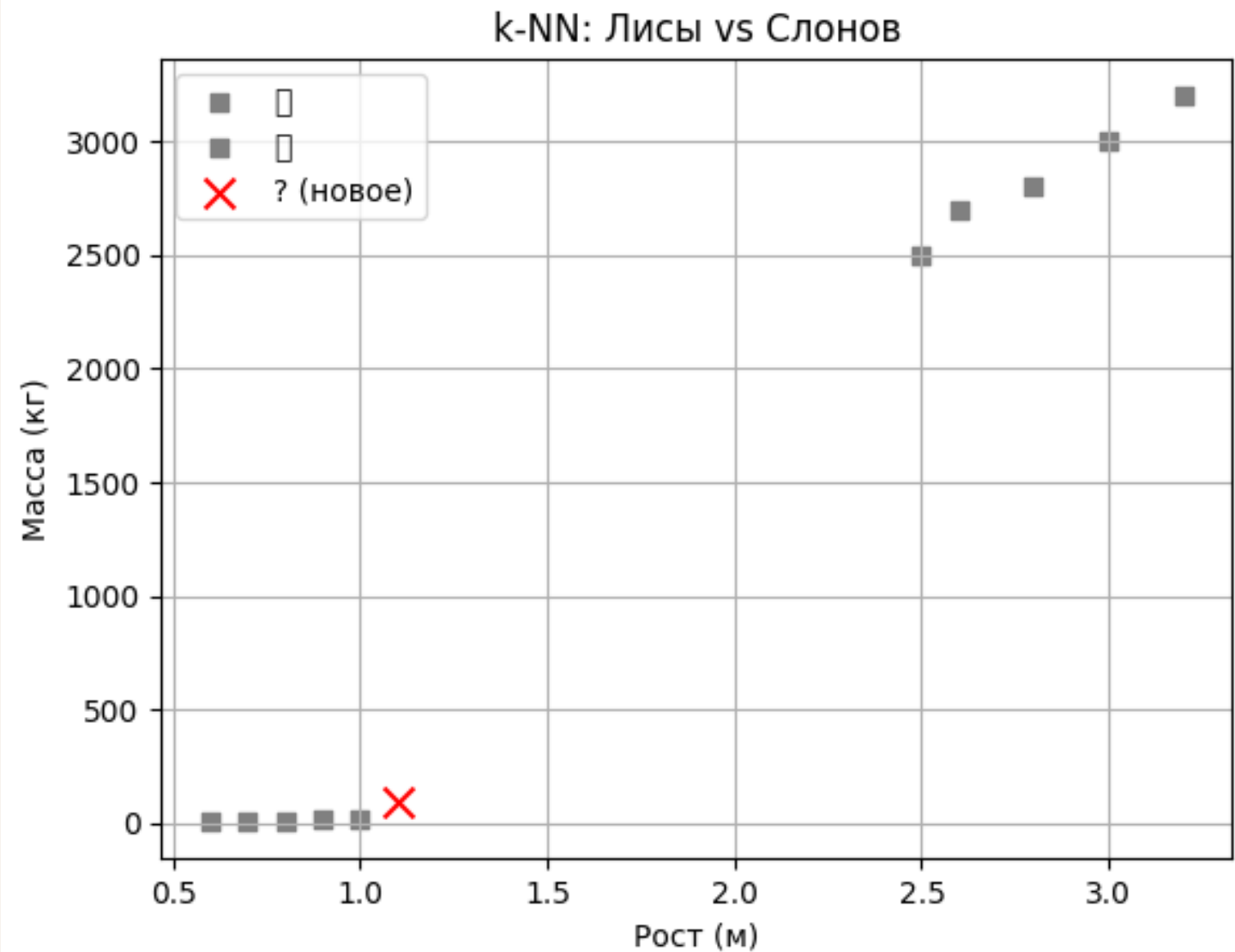
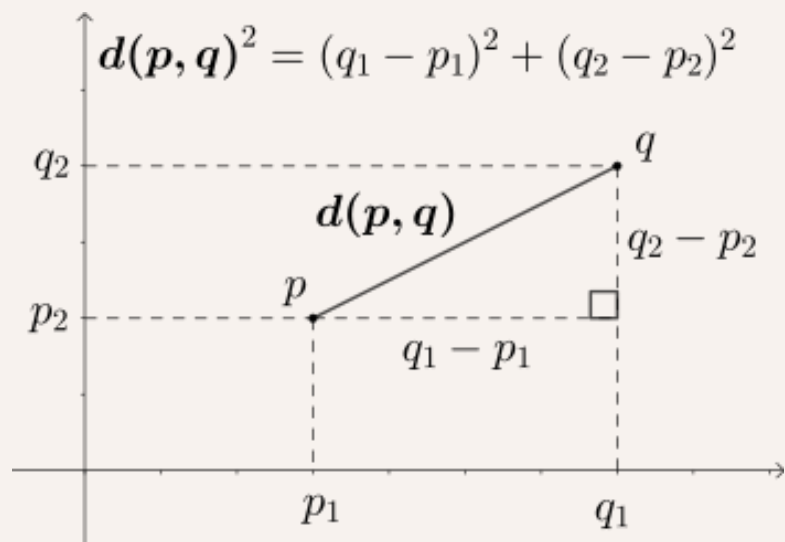
Задача

- Вы биолог, изучающий диких животных. У вас есть данные о **росте (м)** и **массе (кг)** 10 животных. Нужно определить, кто перед вами:  **Лиса** или  **Слон**, используя алгоритм **k-ближайших соседей** ($k = 3$).
 - **Новая особь**: рост 1.1 м, масса 90 кг. Кто это?
-

Датасет

Рост (м)	Масса (кг)	Класс
0.6	8.0	 Лиса
0.7	10.5	 Лиса
0.8	12.0	 Лиса
0.9	14.0	 Лиса
1.0	18.0	 Лиса
2.5	2500.0	 Слон
2.6	2700.0	 Слон
3.0	3000.0	 Слон
3.2	3200.0	 Слон
2.8	2800.0	 Слон

Формулы



Решение

```
new_point = (1.1, 90.0)
k = 3

def euclidean(p1, p2):
    return math.sqrt((p1[0]-p2[0])**2 + (p1[1]-p2[1])**2)

distances = sorted([(euclidean((x, y), new_point), label) for x, y, label in data])
k_nearest = distances[:k]

votes = {}
for _, label in k_nearest:
    votes[label] = votes.get(label, 0) + 1

predicted_class = max(votes, key=votes.get)
print("Предсказанный класс:", predicted_class)

for x, y, label in data:
    plt.scatter(x, y, label=label if label not in plt.gca().get_legend_handles_labels()[1] else "",
               c='orange' if label == 'Лиса' else 'gray', marker='o' if label == 'Слон' else 's')
```

Решение

```
from sklearn.neighbors import KNeighborsClassifier
import numpy as np

# Данные: рост (м), масса (кг)
X = np.array([
    [0.6, 8.0],
    [0.7, 10.5],
    [0.8, 12.0],
    [0.9, 14.0],
    [1.0, 18.0],
    [2.5, 2500.0],
    [2.6, 2700.0],
    [3.0, 3000.0],
    [3.2, 3200.0],
    [2.8, 2800.0]
])

y = ['🦊 Лиса'] * 5 + ['🐘 Слон'] * 5

# Новая особь
new_animal = np.array([[1.1, 90.0]])

# Модель
knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(X, y)

# Предсказание
prediction = knn.predict(new_animal)
print("Предсказанное животное:", prediction[0])
```

Предсказанное животное: 🦊 Лиса