

Искусственный Интеллект

Лекция 5: Предобработка данных

Мартынюк Полина Антоновна

telegram: @PAMartynyuk

email: pa-martynyuk@yandex.ru



Числовые данные: непрерывные и дискретные значения

Датасет «diamonds»

- **carat (караты)**: Вес бриллианта, измеряемый в каратах.
- **depth (глубина)**: Процент глубины бриллианта относительно его диаметра.
- **table (стол)**: Процент ширины верхней части бриллианта относительно его диаметра.
- **price (цена)**: Цена бриллианта в долларах США.
- **x (длина)**: Длина бриллианта в миллиметрах.
- **y (ширина)**: Ширина бриллианта в миллиметрах.
- **z (глубина)**: Глубина бриллианта в миллиметрах.

	carat	depth	table	price	x	y	z
0	0.23	61.5	55.0	326	3.95	3.98	2.43
1	0.21	59.8	61.0	326	3.89	3.84	2.31
2	0.23	56.9	65.0	327	4.05	4.07	2.31
3	0.29	62.4	58.0	334	4.20	4.23	2.63
4	0.31	63.3	58.0	335	4.34	4.35	2.75
...
53935	0.72	60.8	57.0	2757	5.75	5.76	3.50
53936	0.72	63.1	55.0	2757	5.69	5.75	3.61
53937	0.70	62.8	60.0	2757	5.66	5.68	3.56
53938	0.86	61.0	58.0	2757	6.15	6.12	3.74
53939	0.75	62.2	55.0	2757	5.83	5.87	3.64

Числовые данные: временные ряды

Датасет «Airline Passengers»

"Датасет пассажирских авиаперевозок" (Airline Passengers Dataset) содержит данные о числе пассажиров, перевозимых авиакомпанией с января 1949 года по декабрь 1960 года по месяцам.



	Month	Passengers
0	1949-01	112
1	1949-02	118
2	1949-03	132
3	1949-04	129
4	1949-05	121
...
139	1960-08	606
140	1960-09	508
141	1960-10	461
142	1960-11	390
143	1960-12	432

Категориальные данные

- **Бинарные**

Всего 2 категории (класса)

- **Небинарные**

Произвольное число категорий (классов)

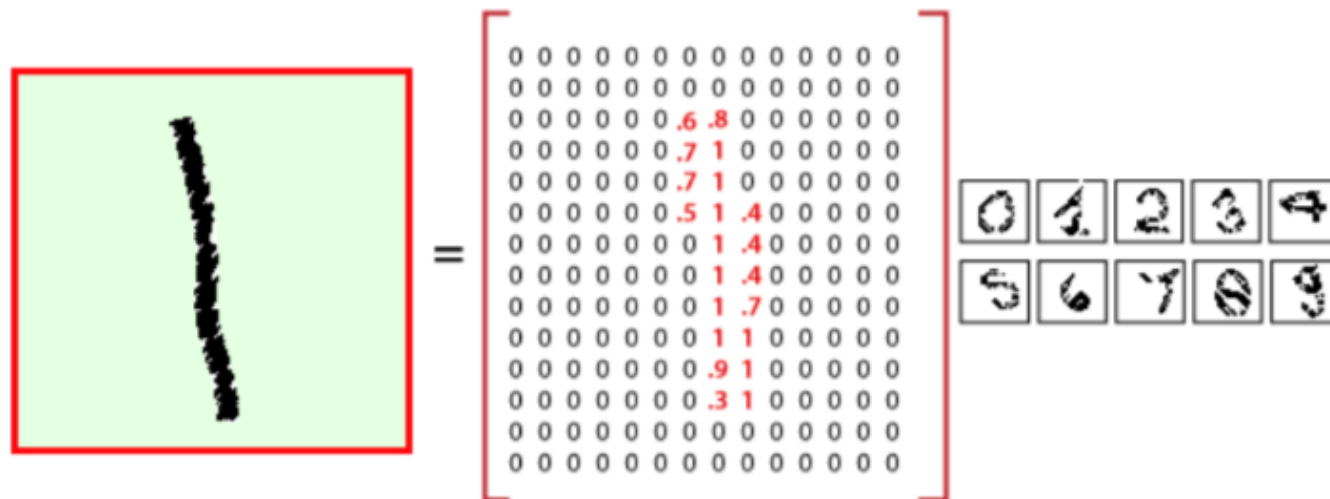
Текст

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Изображения

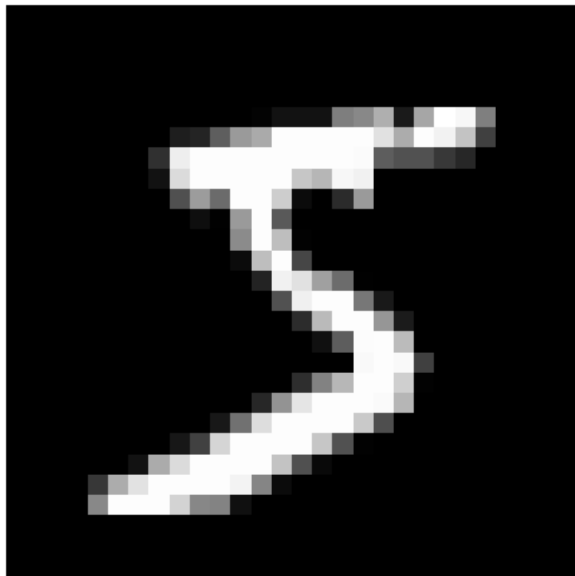
Изображения представляются в цифровой форме с использованием **пикселей**. Каждый пиксель имеет определенный цвет, который может быть представлен в формате ЧБ (черно-белое изображение) или RGB (красный, зеленый, синий) или в других цветовых пространствах. Значения цветовых каналов для каждого пикселя могут быть использованы как признаки для обучения моделей.

Размеры изображений могут варьироваться от небольших значений (например, 28x28 пикселей для MNIST) до высоких разрешений для фотографий.

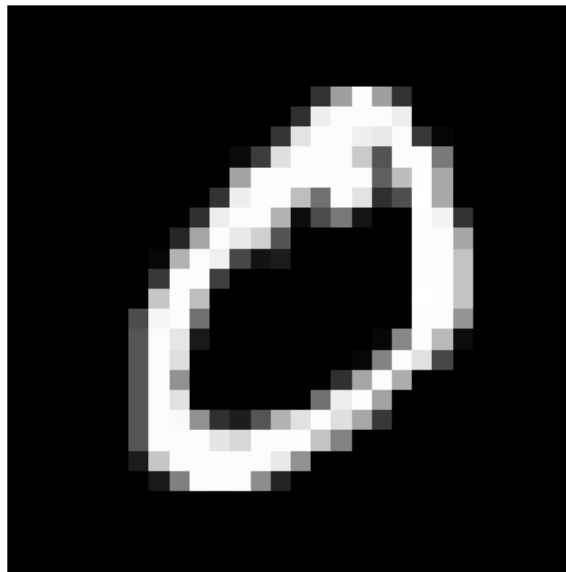


Изображения

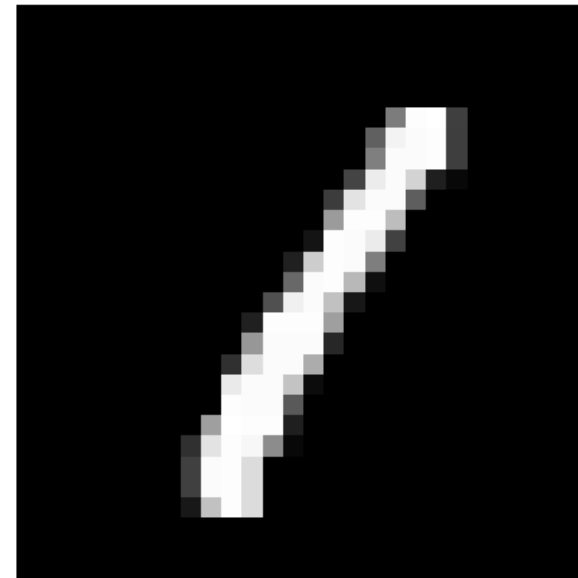
Label: 5



Label: 0



Label: 1

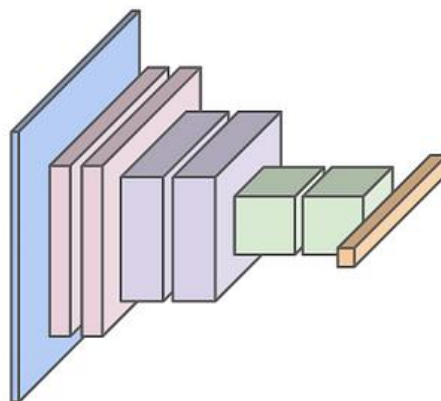


Видео

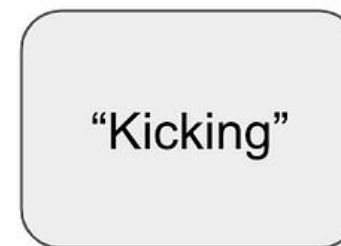
Изображения + Временные ряды



Input Video

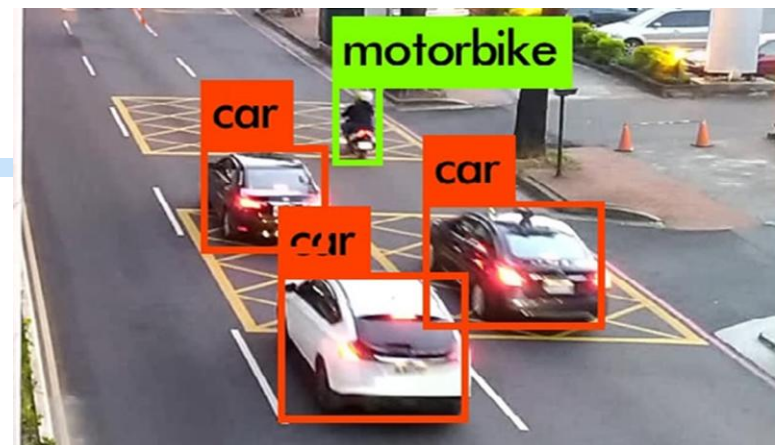


Deep Network



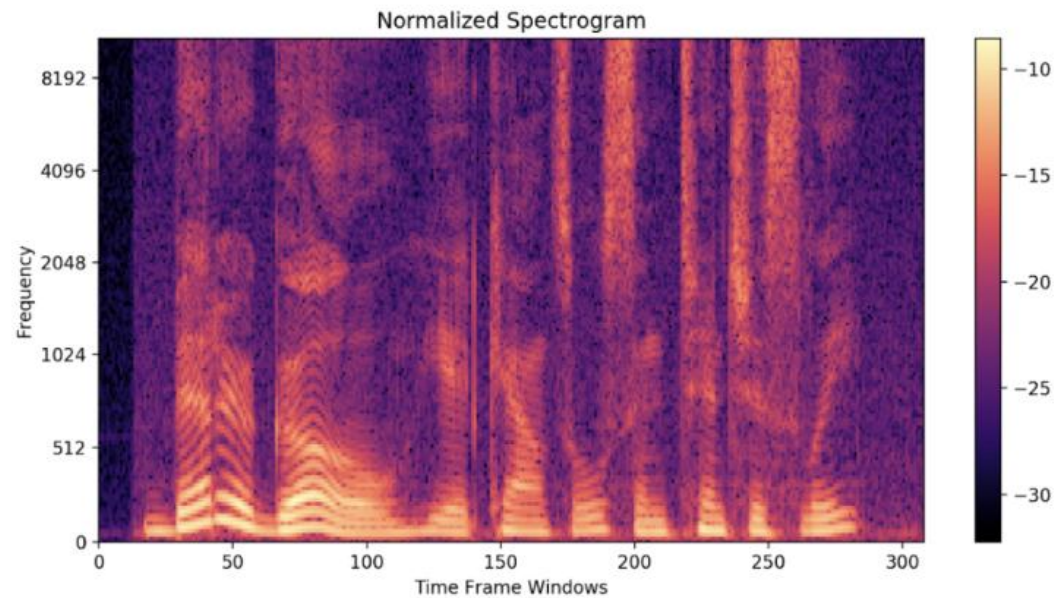
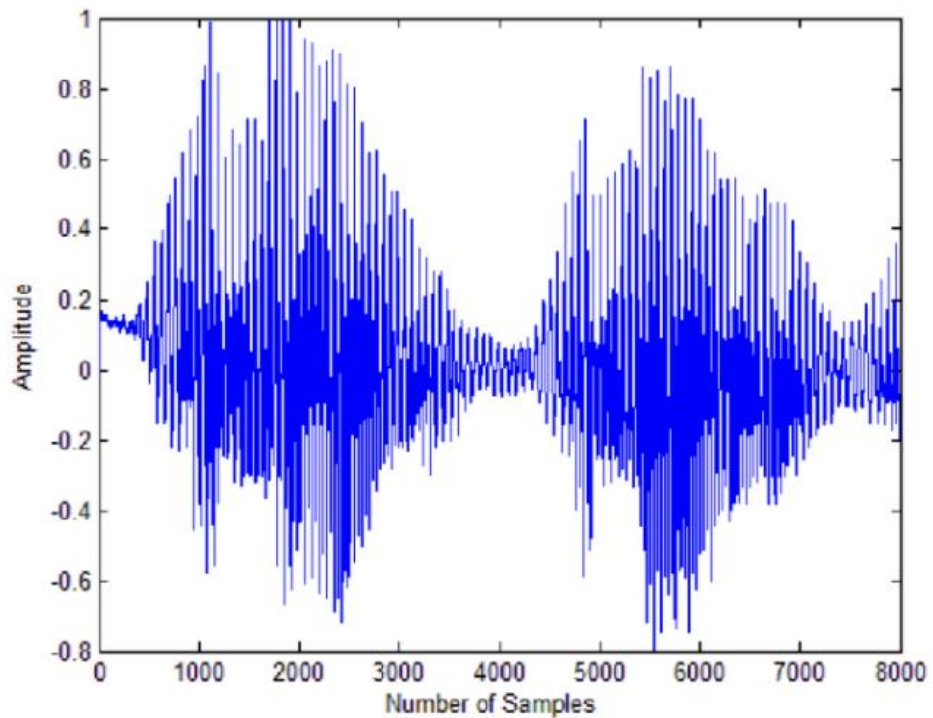
Semantic Classification

Bounding boxes



Аудио и речь

Преобразование в изображение (преобразование Фурье)



Проверка качества данных: NaN

Пропущенные значения – NaN (np.nan)

NaN превращает любую операцию с собой в NaN

	Name	Age	Gender	Seat Class	Ticket Price
0	John Doe	32.0	Male	Business	1000
1	Jane Smith	45.0	Female	Economy	500
2	Bob Johnson	NaN	Male	Economy	450
3	Susan Williams	28.0	Female	NaN	600

Проверка качества данных:NaN

Пути решения:

- **Удаление строк или столбцов:** В Pandas, для удаления строк с отсутствующими значениями, можно использовать `df.dropna()`, а для удаления столбцов - `df.dropna(axis=1)`.
- **Заполнение средними значениями:** Пропущенные значения можно заменить средними или медианными значениями из соответствующего столбца. В Pandas, для заполнения отсутствующих значений средними значениями можно использовать `df.fillna(df.mean())`.
- **Интерполяция:** Для временных рядов и числовых данных вы можете использовать метод интерполяции для заполнения отсутствующих значений на основе соседних значений.
- **В случае самостоятельного сбора данных – Валидация.**

Высокоуровневая работа с данными

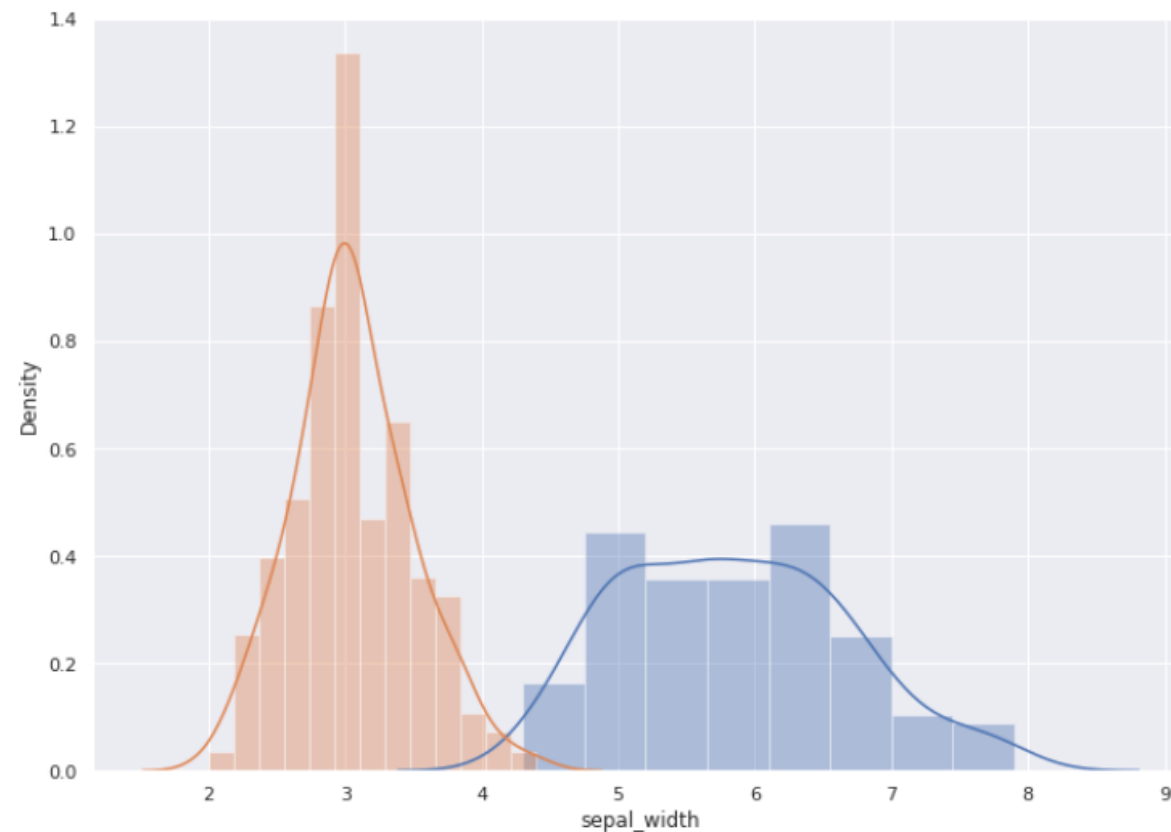
- **Агрегация**
- **Деагрегация**
- **Обогащение**

Классический пример - добавление географических данных:

Если у вас есть данные о клиентах и их почтовых адресах, вы можете обогатить эти данные, добавив информацию о географических координатах, чтобы знать, где находятся ваши клиенты.

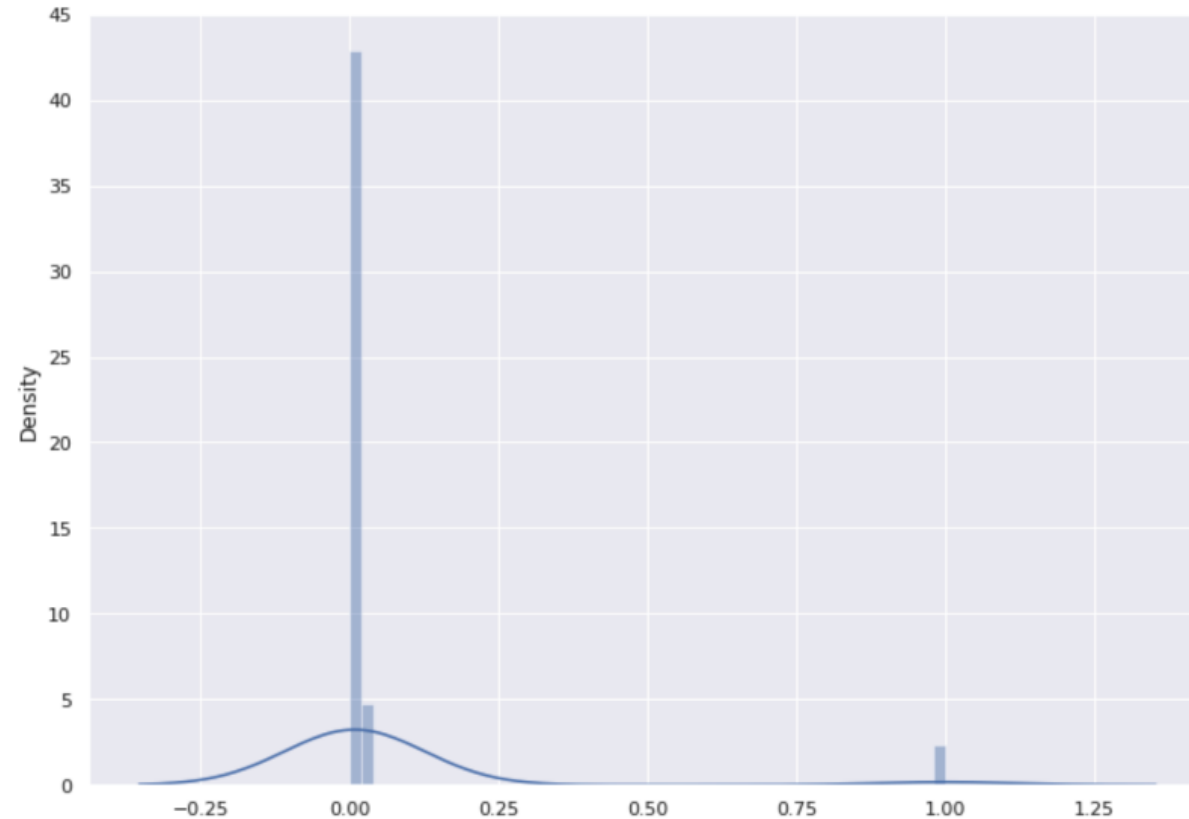
Визуализация данных: числовые данные

- **Гистограммы:** показывают частоту встречаемости значений



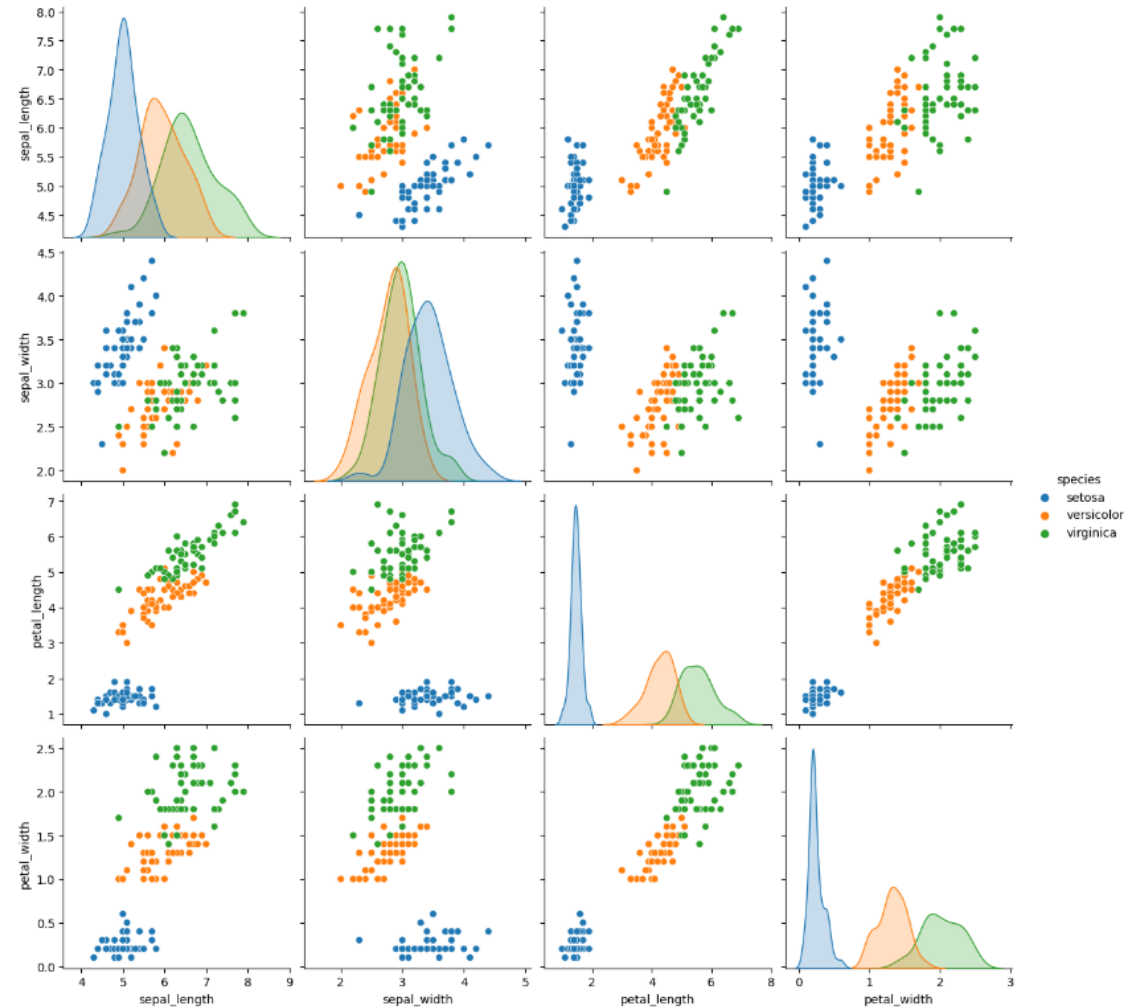
Визуализация данных: числовые данные

- **Выбросы и аномалии:** их можно обработать отдельно



Визуализация данных: числовые данные

- **Pairplot:**
зависимость признаков друг от друга, гистограммы для каждого признака



Визуализация данных: числовые данные

- **Корреляция, heatmap:**
связанность признаков друг с другом



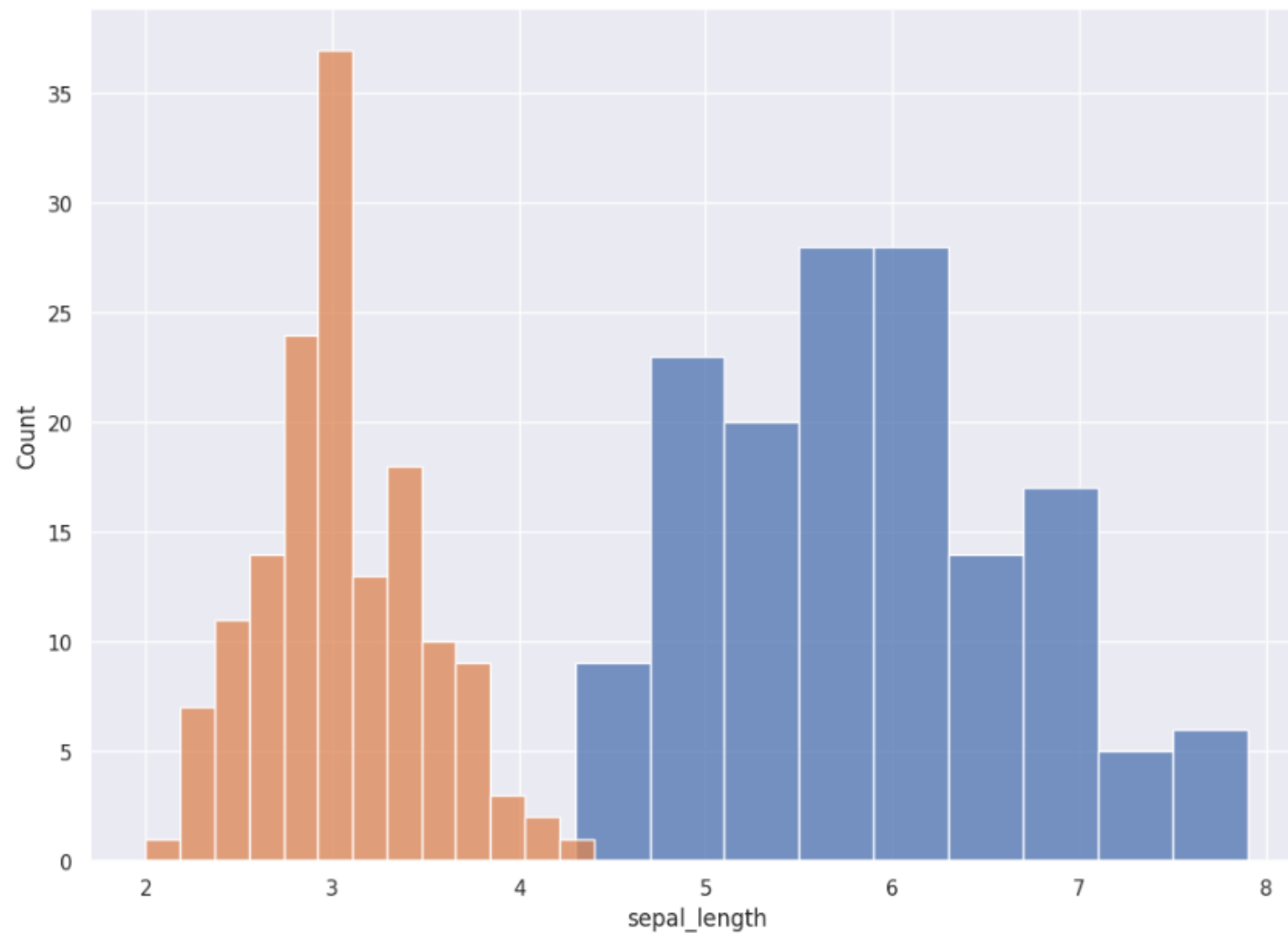
Предобработка данных: числовые данные

- **Разные диапазоны**

	Дом	Площадь (кв. футов)	Цена (\$)	Число спален	Год постройки
0	Дом 1	1500	250000	3	1990
1	Дом 2	2200	450000	4	2005
2	Дом 3	1700	350000	3	1985
3	Дом 4	1200	150000	2	1970

$$y = x_1 w_1 + x_2 w_2 + \dots + x_p w_p + b$$

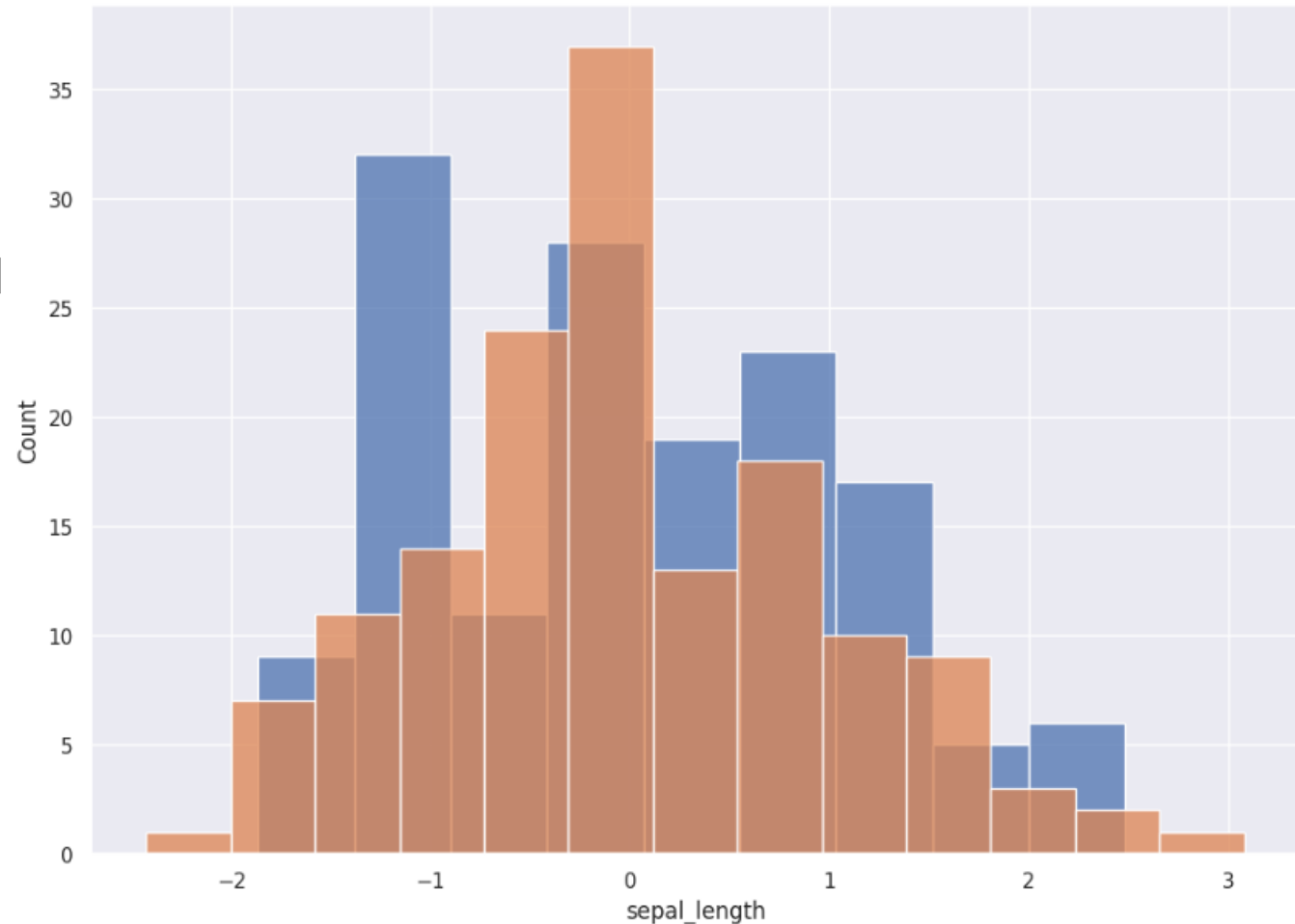
Предобработка данных: числовые данные



Предобработка данных: числовые данные

- **Нормализация:**
распределение
данных центрируется
вокруг нуля

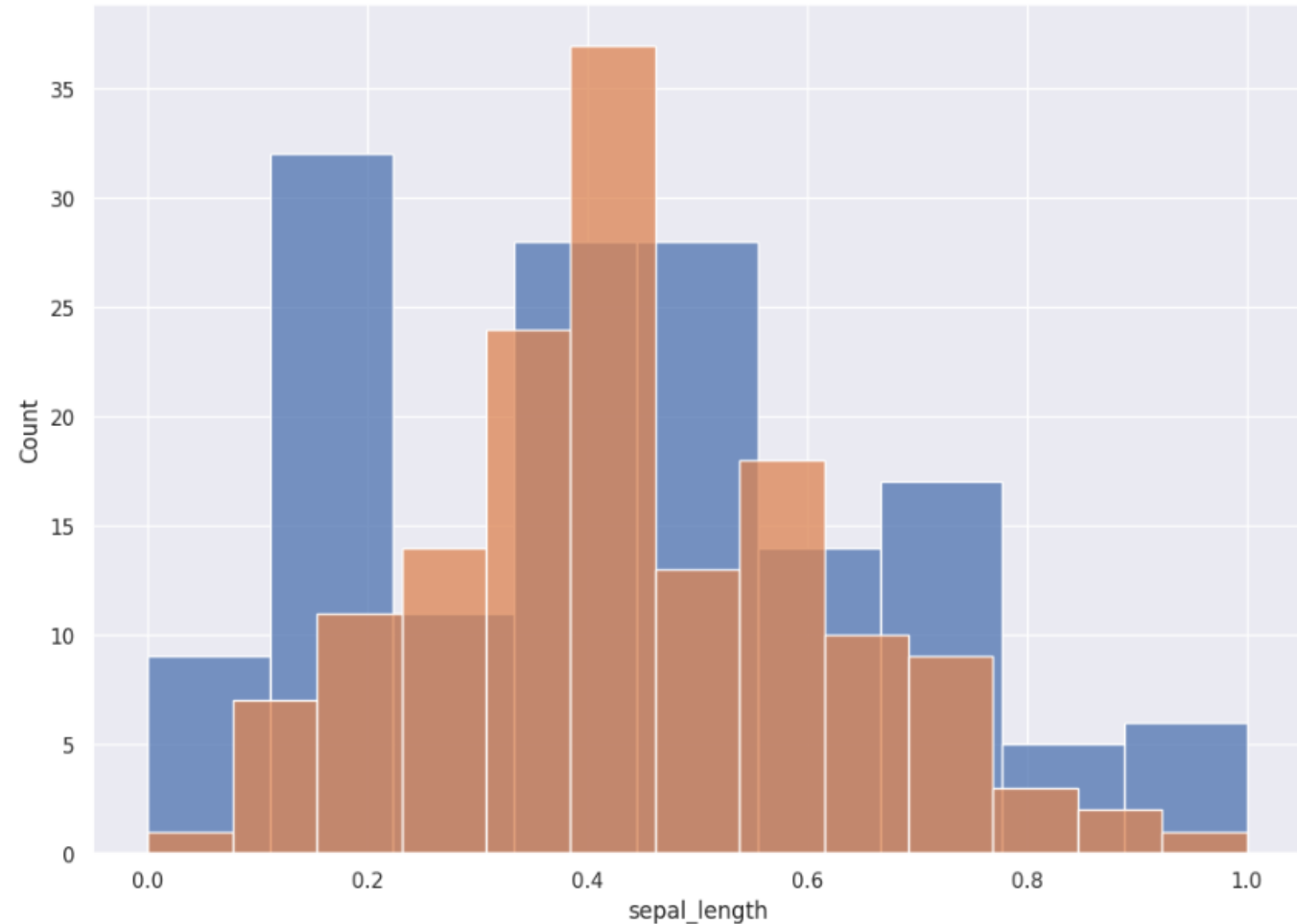
$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$



Предобработка данных: числовые данные

- **Стандартизация:**
распределение
данных центрируется
вокруг нуля и
стандартное
отклонение равно
единице

$$X' = \frac{X - \mu}{\sigma}$$



Предобработка данных: категориальные данные

- **Определить порядок:**
отсортировать значения

XXS < XS < S < M < L < XL < XXL



0 < 1 < 2 < 3 < 4 < 5 < 6



Предобработка данных: категориальные данные

- Как быть с номинальными значениями, для которых нельзя определить отношения порядка?


	Район
0	Манхэттен
1	Бруклин
2	Квинс
3	Бронкс
4	Стэтен-Айленд



Предобработка данных: категориальные данные

- **One-hot encoding:** превращает классы в вектор с одной единицей (эффективен при сравнительно малом числе классов)

	Район		Район_Бронкс	Район_Бруклин	Район_Квинс	Район_Манхэттен	Район_Статен-Айленд
0	Манхэттен	0	0	0	0	1	0
1	Бруклин	1	0	1	0	0	0
2	Квинс	2	0	0	1	0	0
3	Бронкс	3	1	0	0	0	0
4	Статен-Айленд	4	0	0	0	0	1



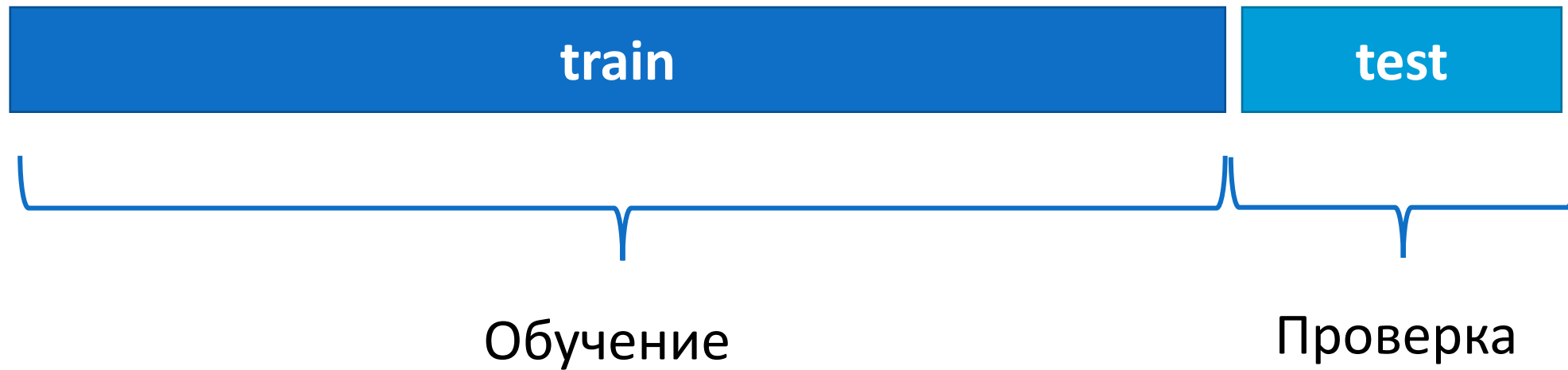
Деление на train/val/test

Почему обучать модель на всём датасете – плохая идея?

train

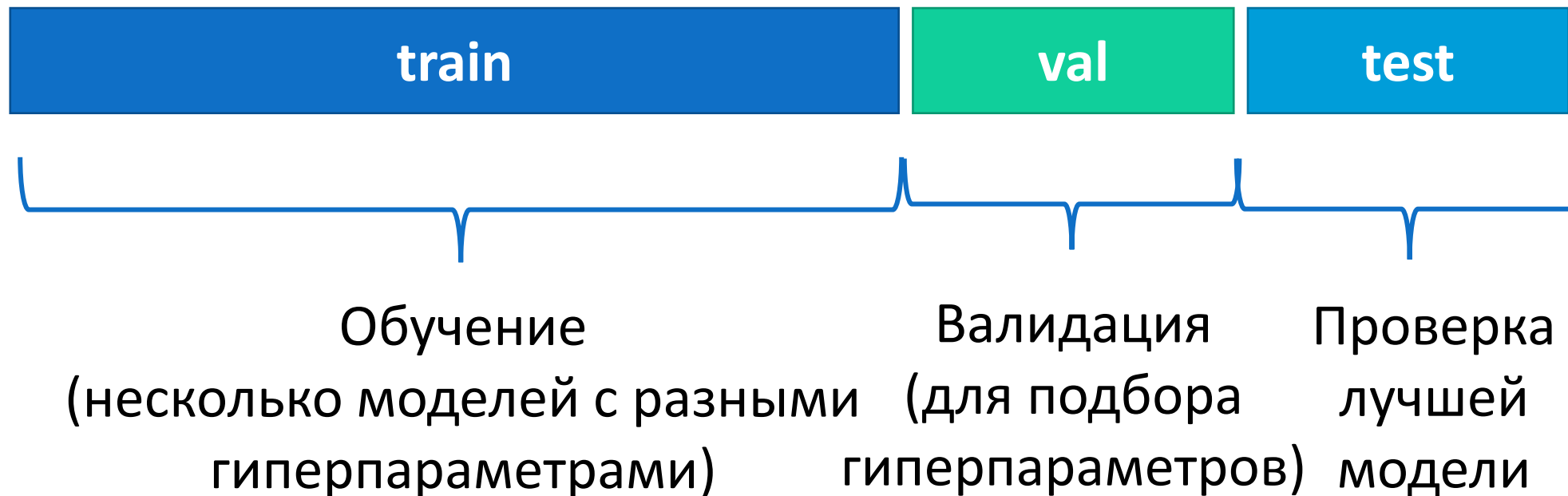
Деление на train/val/test

Обучающие и тестовые данные



Деление на train/val/test

Обучающие, валидационные и тестовые данные



Деление датасета

train_test_split

```
[ ] from sklearn.model_selection import train_test_split
```

```
[ ] X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1)
```

Спасибо за внимание!

Конец Лекции 4