Основные понятия

Регрессия — вид задачи машинного обучения по прогнозу значений одного непрерывного количественного признака по значениям других. Пример задачи регрессии — предсказание стоимости квартиры по количеству комнат, метражу, удалённости от центра города и другим параметрам.

Признаки — свойства или характеристики объекта, среди которых модель ищет закономерности для предсказания.

Объекты, наблюдения — отдельные предметы, существа или характеристики, информация о которых хранится в строках датасета. Модели МО предсказывают свойства новых объектов, опираясь на информацию о старых.

	Признак				
Объект	Порода	Окрас	Пол	Возраст	Вес
Том	Персидская	Чёрный	Кот	7	6.2
Берт	Мейн-кун	Белый	Кот	5	10.9
Ася	Бенгальская	Коричневый	Кошка	9	4.6
Феликс	Британская	Голубой	Кот	2	3.8
Китти	Сиамская	Лиловый	Кошка	4	5.0

Целевой признак — признак, значение которого нужно спрогнозировать.

Входные признаки — все признаки, кроме целевого, среди них модель ищет зависимости для предсказания.

Признаковое описание — совокупность всех значений входных признаков одного объекта.

Первая модель и библиотека sklearn

Входные признаки

Целевой признак

	Порода	Окрас	Пол	Возраст	Вес
Объект → Том	Персидская	Чёрный	Кот	7	6.2
Берт	Мейн-кун	Белый	Кот	5	10.9
Ася	Бенгальская	Коричневый	Кошка	9	4.6
Феликс	Британская	Голубой	Кот	2	3.8
Китти	Сиамская	Лиловый	Кошка	4	5.0

Переменную с входными признаками принято называть большим X, а переменную с целевым признаком — маленьким y:

```
# сохранение входных признаков в переменную X
X = df.drop('Целевой признак', axis=1)

# сохранение целевого признака в переменную y
y = df['Целевой признак']
```

Библиотека sklearn

scikit-learn — одна из самых популярных библиотек в машинном обучении и анализе данных. Её название часто сокращают до sklearn. У неё открытый исходный код, поэтому её свободно используют в личных и коммерческих целях. Она поможет:

- выполнить предобработку данных,
- решить задачи регрессии, классификации и кластеризации,
- подобрать модель МО под задачу.

Выборки данных

Выборки — наборы объектов из исходного датасета, которые нужны для разных целей:

- На **тренировочной выборке** модель МО обучают.
- На **валидационной выборке** подбирают параметры модели и исправляют ошибки обучения. Не все базовые модели требуют донастройки, поэтому валидационную выборку выделяют не всегда.
- На **тестовой выборке** проверяют итоговое качество модели.

```
from sklearn.model_selection import train_test_split

# разделение на тренировочную и тестовую выборки
X_train, X_test, y_train, y_test = train_test_split(
    X, # входные признаки
    y, # целевой признак
    test_size=0.25, # размер тестовой выборки
    random_state=RANDOM_STATE # фиксирование случайности
)
```

Обучение линейной регрессии

```
from sklearn.linear_model import LinearRegression

# объявляем модель
model_lr = LinearRegression()

model_lr.fit(X_train, y_train) # обучаем модель

predictions = model_lr.predict(X_test_scalled) # получаем предсказания
```