**EDA-3. Проектирование признаков**

1. Проектирование признаков:

1. создание признаков:
   1. разбор категорий;
   2. разбор даты и времени;
   3. разбор числовых признаков;
   4. разбор текста.
2. преобразование признаков;
3. отбор признаков.

2. Внешние источники данных

**Кодирование признаков:**

import category\_encoders as ce

1 ПОРЯДКОВОЕ КОДИРОВАНИЕ. ORDINAL ENCODING

ord\_encoder = ce.OrdinalEncoder()

data\_bin = ord\_encoder.fit\_transform(clothing[['size']])

2 ОДНОКРАТНОЕ КОДИРОВАНИЕ. ONE-HOT ENCODING

encoder = ce.OneHotEncoder(cols=['type'])

type\_bin = encoder.fit\_transform(clothing['type'])

clothing\_dummies = pd.**get\_dummies**(clothing, columns=['type'])

3 ДВОИЧНОЕ КОДИРОВАНИЕ

bin\_encoder = ce.BinaryEncoder(cols=['type'])

type\_bin = bin\_encoder.fit\_transform(clothing['type'])

**НОРМАЛИЗАЦИЯ:**

1. MINMAXSCALER все значения признаков будут преобразованы в диапазон [0,1]

ЧУВСТВИТЕЛЕН К ВЫБРОСАМ (ведь берем мин и макс), использовать на очищенных данных

1-Из каждого значения признака х вычитается минимальное значение этого признака,

2-Результат вычитания делится на разность между максимумом и минимумом признака.

mm\_scaler = preprocessing.MinMaxScaler()

df\_mm = mm\_scaler.fit\_transform(df)

2. ROBUSTSCALER

RobustScaler вычитает из данных медиану и делит результат на Межквартильный размах

r\_scaler = preprocessing.RobustScaler()

df\_r = r\_scaler.fit\_transform(df)

**СТАНДАРТИЗАЦИЯ** (среднее значений - 0, а стандартное отклонение — 1)

В отличие от нормализации стандартизация предполагает, что признак распределён нормально.

STANDARDSCALER

s\_scaler = preprocessing.StandardScaler()

df\_s = s\_scaler.fit\_transform(df)

Инструкция по преобразованию признаков:

* если признак распределён нормально, то его необходимо стандартизировать;
* если признак распределён ненормально, его необходимо нормализовать;
* если разброс значений небольшой, то можно обойтись без преобразования данных.

**Отбор признаков.**

Отбор признаков — процесс выбора важных признаков, наиболее влияющих на предсказание.

Отбор признаков помогает:

1: Сократить время обучения. Чем меньше данных, тем быстрее обучается модель.

2: Повысить качество предсказания (шум, мультиколлинеарность)

Мультиколлинеарность — сильная корреляционная связь между признаками, отбираемыми для обучения.