Занятие 3

ОБРАБОТКА И АНАЛИЗ ДАННЫХ

ПРЕДУСЛОВИЕ

- Сохраните себе на гугл диск следующий файл: https://clck.ru/3NAL8w

МОДУЛЬ 1: РАБОТА С ПРОПУЩЕННЫМИ ЗНАЧЕНИЯМИ

Пропущенные значения — это пустые ячейки в таблице. Если таких данных много, это мешает анализу и обучению модели.

Машинные алгоритмы не работают с NaN, поэтому нужно их заполнять или удалять.

МОДУЛЬ 1: РАБОТА С ПРОПУЩЕННЫМИ ЗНАЧЕНИЯМИ

Метод	Что делает	Пример в Pandas
isnull().sum()	Показывает количество пропусков в каждом столбце	df.isnull().sum()
dropna()	Удаляет строки или столбцы с пропусками	df.dropna()
fillna(значение)	Заполняет пропуски заданным значением	df['age'].fillna(df['age'].mean())
df.copy()	Копирует DataFrame для безопасной обработки	df_cleaned = df.copy()

МОДУЛЬ 2: ОЧИСТКА ОТ ВЫБРОСОВ (АНАЛИЗ ЭКСТРЕМАЛЬНЫХ ЗНАЧЕНИЙ)

Что это:
Выбросы — это значения, которые сильно отличаются от большинства данных.

Зачем убирать:
Они могут сильно испортить результат модели, особенно в задачах регрессии.

МОДУЛЬ 2: ОЧИСТКА ОТ ВЫБРОСОВ (АНАЛИЗ ЭКСТРЕМАЛЬНЫХ ЗНАЧЕНИЙ)

Метод	Название	Описание	Формула / Принцип
IQR	Межквартильный размах	Классический способ обнаружения выбросов на основе 1.5 × IQR	IQR = Q3 − Q1Выбросы: < Q1 − 1.5×IQR или > Q3 + 1.5×IQR
Z-score	Z-оценка	Оценивает, насколько далеко значение от среднего	Z = (x − μ) / σВыбросы:
Modified Z-score	Модифицированная Z-оценка	Лучше работает при наличии выбросов, использует медиану	Z = 0.6745 × (x − median) / MADВыбросы:

МЕЖКВАРТАЛЬНЫЙ ИНТЕРВАЛ

- Межквартильный интервал, также известный как интерквартильный размах (IQR), это мера рассеяния, которая показывает разброс значений в центральной части выборки. Он представляет собой разницу между третьим (Q3) и первым (Q1) квартилями набора данных. Другими словами, это диапазон значений, в который попадают 50% данных, расположенных в середине упорядоченного ряда.

Z-ОЦЕНКА

- Стандартизованная величина, которая служит мерой силы выброса, то есть степени отличия конкретной оценки от типичного значения.

$$Z = \frac{\bar{x} - \mu}{\frac{\sigma}{\sqrt{n}}}$$

Где:

- $\bar{x}$ - среднее значение выборки,
- μ - среднее значение генеральной совокупности (предполагаемое),
- σ - стандартное отклонение генеральной совокупности,
- n - размер выборки.

МОДУЛЬ 3: МАСШТАБИРОВАНИЕ ДАННЫХ (НОРМАЛИЗАЦИЯ)

Что это:
процесс преобразования числовых признаков в наборе данных к одному масштабу или диапазону, чтобы избежать доминирования одних признаков над другими из-за их начальных значений

Почему важно:
Многие модели (например, линейная регрессия) чувствительны к разным шкалам.

МОДУЛЬ 3: МАСШТАБИРОВАНИЕ ДАННЫХ (НОРМАЛИЗАЦИЯ)

Метод	Название	Описание	Формула
Min-Max Scaling	Масштабирование до [0, 1]	Приводит значения к диапазону [0, 1]	$X' = (X - \min) / (\max - \min)$
StandardScaler	Z-нормализация (стандартизация)	Приводит данные к среднему 0 и стандартному отклонению 1	$X' = (X - \mu) / \sigma$
RobustScaler	Устойчивое масштабирование	Масштабирует данные, устойчиво к выбросам, использует медиану и IQR	$X' = (X - \text{median}) / \text{IQR}$
Normalizer	Нормализация по вектору	Делает длину каждого вектора (строки) равной 1 — полезно для текстов или направлений	$X' = X /$

МОДУЛЬ 4: КОДИРОВАНИЕ КАТЕГОРИАЛЬНЫХ ПРИЗНАКОВ

Что это:
Категориальные данные — это текстовые значения ("male", "female", "C", "S").

Проблема:
Модели не понимают текст — им нужны числа.

МОДУЛЬ 4: КОДИРОВАНИЕ КАТЕГОРИАЛЬНЫХ ПРИЗНАКОВ

Метод	Название	Описание	Пример или формула
One-Hot Encoding	Кодирование "один из N"	Создаёт отдельные бинарные столбцы для каждой категории	"Пол": male → [1,0], female → [0,1]
Label Encoding	Числовое кодирование	Преобразует категории в числа (риск: модель может посчитать, что есть порядок)	"Класс": A → 0, B → 1, C → 2
pd.get_dummies()	Быстрая one-hot реализация	Автоматически создаёт бинарные признаки из категориальных колонок	pd.get_dummies(df, columns=['sex'])
Ordinal Encoding	Порядковое кодирование	Используется, если категории имеют порядок (например: низкий, средний, высокий)	"Размер": small → 0, medium → 1, large → 2
Binary Encoding	Бинарное кодирование	Комбинация label + бинарное представление числа	A → 0 → 000, B → 1 → 001 и т.д.
Target Encoding	Целевая кодировка	Заменяет категорию на среднее значение целевой переменной по этой категории	"Город" → средний доход

МОДУЛЬ 5: УДАЛЕНИЕ ДУБЛИКАТОВ

```
df = df.drop_duplicates()
```

Что это:
Повторяющиеся строки в таблице.

Зачем убирать:
Они могут искажать результаты и перегружать модель.

МОДУЛЬ 6: ПРИВЕДЕНИЕ ТИПОВ ДАННЫХ

Что это: Иногда числа записаны как строки, или даты как обычный текст.

Решение: df['age'] = df['age'].astype(int)

Типы: int — целые числа, float — вещественные числа, str — строки, bool — булевы (True/False)

МОДУЛЬ 7: ОБРАБОТКА ТЕКСТОВЫХ ДАННЫХ

Что это:
Когда данные — это тексты (отзывы, комментарии, заголовки).

Проблема:
Машины не понимают текст напрямую — нужно его "очистить" и преобразовать в числа.

МОДУЛЬ 7: ОБРАБОТКА ТЕКСТОВЫХ ДАННЫХ

Этап	**Что делает**	**Python**
Приведение к нижнему регистру	Все буквы — маленькие	text.lower()
Удаление пунктуации	Убираем знаки препинания	str.translate(…)
Токенизация	Делим на слова	nltk.word_tokenize(text)
Удаление стоп-слов	Убираем частые бесполезные слова	word not in stopwords.words('english')
Лемматизация	Приводим слова к начальной форме	lemmatizer.lemmatize('running') → 'run'
Векторизация	Переводим текст в вектор для модели	TfidfVectorizer, CountVectorizer, Word2Vec

ВЕКТОРИЗАЦИЯ???

- Векторизация – это термин, обозначающий классический подход к преобразованию входных данных из их исходного формата *(например, текста)* в векторы действительных чисел, которые понятны моделям машинного обучения.

НА ПРИМЕРЕ

- texts = [
- "Я люблю котов",
- "Я люблю собак"
-]

Шаг 1: Строим словарь (все уникальные слова)

Слово	Индекс
я	0
люблю	1
котов	2
собак	3

ШАГ 2: ПРЕДСТАВЛЯЕМ ПРЕДЛОЖЕНИЯ В ВИДЕ ВЕКТОРОВ

- Смотрим относительно предложения «Я люблю котов»:
- Слова: я, люблю, котов встречаются по 1 разу , следовательно вектор будет [1, 1, 1]
- Слова: я, люблю, собак встречаются по 1, 1 и 0 раз соответственно. Вектором будет [1, 1, 0]

- Это и есть **Bag-of-Words модель**
- Мы **не учитываем порядок слов**, а просто смотрим, какие слова есть в предложении.

АЛЬТЕРНАТИВЫ

Метод	Что делает
CountVectorizer	Считает, сколько раз встречается слово (частотная модель)
TfidfVectorizer	Взвешивает частоту с учетом важности слова (TF * IDF)
Word2Vec / GloVe	Учитывает контекст, слова с похожим значением близки по вектору
BERT	Глубокое понимание смысла предложения в контексте

АНАЛИЗ ТЕКСТА + ПРОСТОЯ GPT МОДЕЛЬ

- https://colab.research.google.com/drive/1Is4t4UhqxBQOg9sAH8iDTjmRgszmvANo#scrollTo=Vhc0PiHnW0U_