Предобработка данных

Синтаксис

Переименование столбцов

```
# параметру columns передают словарь,
# в котором ключи — старые названия столбцов, а значения — новые
df = df.rename(
    columns={
        'old_name1': 'new_name1',
        'old_name2': 'new_name2',
        'old_name3': 'new_name3',
    }
)
```

Поиск пропущенных значений

```
df.isna()

# подсчёт количества пропущенных значений
df.isna().sum()
```

Заполнение пропущенных значений

```
# аргумент — значение, которым заполняют пропуски
df = df.fillna(0)
df['column'] = df['column'].fillna(0)
```

Удаление пропущенных значений

```
# удаление всех строк, где есть хотя бы одно пропущенное значение
df = df.dropna()

# удаление строк с пропусками в столбцах, перечисленных в списке subset
df = df.dropna(subset=['a','b','c'])

# удаление всех столбцов, где есть хотя бы одно пропущенное значение
df = df.dropna(axis='columns')
```

Предобработка данных

Поиск дубликатов

```
df.duplicated()

# подсчёт количества дубликатов
df.duplicated().sum()
```

Удаление дубликатов

```
'''Чтобы в индексах не было пропусков при удалении дубликатов,
вместе с drop_duplicates() вызывают метод reset_index().
Аргумент drop=True указывают, чтобы не создавать столбец
со старыми значениями индексов.'''
df = df.drop_duplicates().reset_index(drop=True)
```

Вывод всех уникальных значений в столбце

```
# получить набор всех уникальных значений в столбце
df['column'].unique()

# количество всех уникальных значений в столбце
df['column'].nunique()
```

Замена значений

```
# old_value — старое значение, new_value — новое значение
df = df.replace('old_value', 'new_value')
```

Теория

GIGO (от англ. garbage in, garbage out, буквально «мусор на входе — мусор на выходе»). Принцип, утверждающий, что при неверных входных данных даже правильный алгоритм анализа выдаёт неверные результаты.

Предобработка данных

Пропуски

Пропущенные значения могут выглядеть по-разному:

- ожидаемые, None или NaN;
- тексты-заполнители, например: 0, '?', 'NN', 'n/a';
- произвольные значения, которые выбрали создатели таблицы.

Пропуски можно удалять либо заполнять на основе известных данных.

У обоих способов есть свои плюсы и минусы. Удалив пропуски, можно быть уверенным, что оставшиеся данные отвечают всем требованиям, однако так можно потерять важные данные. Заполнив пропуски, можно сохранить наибольшее количество данных. Однако выбранные значения для заполнения могут быть некорректными.

Дубликаты

Дубликаты могут быть разными:

- Явные — строки с абсолютно идентичной информацией. Такие дубликаты увеличивают размер таблицы, и её становится труднее и дольше обрабатывать.
- Неявные, например «джаз» и jazz, обозначающие один жанр. Такие дубликаты тяжелее найти, но они могут привести к ошибкам в результатах.

Переименование столбцов

Как должны выглядеть названия столбцов в датафрейме:

- не содержат пробелов в начале, середине или конце;
- слова разделены подчёркиванием;
- написаны на одном языке и в одном регистре;
- из названия понятно, что за данные хранятся в столбце.