Тема 6. Предобработка данных

Синтаксис

Метод `set_axis()` для изменения названий столбцов

```
# аргументы - список новых названий столбцов,
# axis со значением columns для изменений в столбцах,
# inplace со значением True для изменения структуры данных
df.set_axis(['a','b','c'],axis = 'columns',inplace = True)
```

Методы `isnull()` и `isna()` для определения пропущенных значений. В сочетании с методом `sum()` — подсчёт пропущенных значений.

```
df.isnull().sum()
```

Метод `fillna()` для заполнения пропущенных значений.

```
df = df.fillna(0) # аргумент - значение, на которое будут заменены
                  # пропущенные значения
```

Метод `dropna()` для удаления пропущенных значений

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

или

```
# аргумент subset - названия столбцов, где нужно искать пропуски
df.dropna(subset = ['a','b','c'], inplace = True)
```

или

```
# аргумент axis со значением 'columns' для удаления столбцов с
# хотя бы одним пропущенным значением
df.dropna(axis = 'columns', inplace = True)
```

Метод `duplicated()` для нахождения дубликатов. В сочетании с методом `sum()` - возвращает количество дубликатов.

```
df.duplicated().sum()
```

Метод `drop_duplicates()` для удаления дубликатов. При вызове метода `drop_duplicates()` вместе с повторяющимися строками удаляются их индексы. Поэтому используется с методом `reset_index()`.

```
# аргумент drop со значением True,
# чтобы не создавать столбец со старыми значениями индексов
df.drop_duplicates().reset_index(drop = True)
```

Метод `unique()` для просмотра всех уникальных значений в столбце

```
df['column'].unique()
```

Метод `replace()` для замены значений в таблице или столбце.

```
# первый аргумент - текущее значение
# второй аргмуент - новое значение
df.replace('first_value', 'second_value')
```

Словарь

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

Предобработка — процесс подготовки данных для дальнейшего анализа. Заключается она в поиске проблем, которые могут быть в данных, и в устранении этих проблем.

Таблица, удобная для анализа данных:

- в каждом столбце хранятся значения одной переменной;
- каждая строка содержит одно наблюдение, к которому привязаны значения разных переменных.

Названия столбцов:

- Без пробелов в начале, середине и конце.
- Несколько слов разделяются нижним подчеркиваем.
- На одном языке и в одном регистре.
- Отражают в краткой форме, какого рода информация содержится в каждом столбце.

Пропущенные значения бывают разные:

- чаще всего это None или NaN;
- плейсхолдеры (тексты-заполнители) какого-нибудь общепринятого стандарта, иногда неизвестного вам, но которого придерживаются составители. Чаще всего это **n/a**, **na**, **NA**, и **N.N.** либо **NN**.
- произвольное значение, которое по договорённости между собой используют создатели исходной таблицы данных

Пропущенные значения можно как удалять, так и заполнять на основе известных данных.

- Плюс удаления данных в том, что это простой процесс. Также можно быть уверенным, что те данные, которые остались, хорошие и отвечают всем требованиям. Потенциальные минусы: потеря важной информации и снижение точности.
- Заполнение позволяет сохранить наибольшее количество данных. Очевидный минус — могут получиться плохие результаты на основе уже существующих данных.

Дубликаты (дублированные записи) могут быть следующего вида:

- две и более одинаковых строки с идентичной информацией. Большое количество повторов раздувает размер таблицы, а значит, увеличивает время обработки данных;
- одинаковые по смыслу категории с разными названиями, например, «Политика» и «Политическая ситуация». Замаскированные повторы — источник серьёзных и с трудом обнаруживаемых ошибок в анализе.