

1.4 категоризация

е.g. доход, возраст

1 индикаторы категорий для ускорения обработки

2 вш словарь - мб для биг даты имеет смысл

pivot merge dict vs двойной индекс в pivot

3 мал группы - выводы не сделать

4 pd.cut vs custom func + apply

5 катег-ция по неск столбцам

Если уже произведена:

Нужен чек, что индикация категорий без ошибок

0 переименование столбцов

`.set_axis(['old_name', 'old_name', 'new_name'], axis = 'columns', inplace=True)`

vs

`.rename`

1 проблемы длинных назв-ий категорий

-визуально хуже

-дольше обработка

-сложнее срезы делать

-дольше редачить категории

достаточный фикс - **индикаторы категорий** ввести

2 зачем отдельный словарь с индикаторами категорий?

ненужная фигня, но мб для биг даты имеет смысл

в проекте pivot merge dict

vs двойной индекс в pivot

без разницы

когда много повторов строк

3 Категоризация и мал группы

если группа мало представлена по ней не сделать вывод

4 Используем функцию для категоризации - f(value)

```
df.column.apply(func)
```

vs

```
pd.cut (karpov)
```

5 Функция для катег-ции по неск столбцам - f(row)

(можно 2 столбца как в прош пункте - 1 столбец в индекс)

```
row = pd.Series(data=row_values, index=row_columns)
```

```
row(name_col) = value
```

case - конструкция

```
df.apply
```

```
apply(func, axis = 1)
```

case:

оптимизацией работы службы поддержки

по нашей **сегментации клиентов по возрасту**

и занятости

может быть создана

система приоритетов поступающих проблем