Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

Факультет «Информатика и системы управления»

Кафедра «Системы обработки информации и управления»

Отчет по лабораторной работе №2

«Обработка пропусков в данных, кодирование категориальных признаков, масштабирование данных»

по дисциплине «Технологии машинного обучения»

Выполнил:
студент группы ИУ5Ц-84Б
Папин А.В.
подпись, дата

Проверил:
к.т.н., доц., Ю.Е. Гапанюк
подпись, дата

2024 г.

СОДЕРЖАНИЕ ОТЧЕТА

1. Цель лабораторной работы:

Изучение способов предварительной обработки данных для дальнейшего формирования моделей.

2. Описание задание

1. Выбрать набор данных (датасет), содержащий категориальные признаки и пропуски в данных. Для выполнения следующих пунктов можно использовать несколько различных наборов данных (один для обработки пропусков, другой для категориальных признаков и т.д.)
2. Для выбранного датасета (датасетов) на основе материалов лекции решить следующие задачи:
   - обработку пропусков в данных;
   - кодирование категориальных признаков;
   - масштабирование данных.

3. Основные характеристики датасета

Название датасета: Hotels Dataset (Датасет отелей)

Ссылка: https://www.kaggle.com/datasets/raj713335/tbo-hotels-dataset/data

О датасетах

Этот датасет содержит информацию о 1 000 000+ отелях из различных стран и регионов, таких как их тарифы, отзывы, удобства, местоположение и звездный рейтинг. Данные были собраны из различных источников, таких как веб-сайты отелей, онлайн-агентства по бронированию и платформы отзывов. Датасет может использоваться для различных целей, таких как:

- Исследовательский анализ данных для понимания характеристик и распределения отелей по разным рынкам и сегментам.
- Анализ настроений для извлечения идей из отзывов и рейтингов гостей отеля и выявленидаций для предоставления персонализированных

предложений по бронированию отелей на основе предпочтений и поведения пользователя.

- Прогнозирование цен для оценки оптимальных тарифов для отелей на основе спроса, сезонности и конкуренции.
- Классификация для определения типа и категории отелей на основе их характеристик и атрибутов.

Структура данных

Датасет состоит из 16 столбцов и 1 000 000+ строк, где каждая строка представляет собой отель. Столбцы включают в себя:

countyCode: Код страны, к которой принадлежит отель.

countyName: Название страны, к которой принадлежит отель.

cityCode: Код города, где расположен отель.

cityName: Город, где расположен отель.

HotelCode: Уникальный идентификатор каждого отеля.

hotel_name: Название отеля.

HotelRating: Звездный рейтинг отеля от 1 до 5.

Address: Адрес отеля.

Attractions: Достопримечательности рядом с отелем.

Description: Подробное описание отеля.

FaxNumber: Номер факса отеля.

HotelFacilities: Доступные в отеле удобства.

Map: Местоположение отеля в формате GPS (широта и долгота).

PhoneNumber: Телефонный номер отеля.

PinCode: Почтовый индекс адреса отеля.

HotelWebsiteUrl: Веб-ссылка для бронирования отеля.

4. Изучение данных

```
In [3]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1010033 entries, 0 to 1010032
Data columns (total 16 columns):
 #   Column            Non-Null Count    Dtype
---  ------            --------------    -----
 0   countyCode        1009121 non-null  object
 1    countyName       1010033 non-null  object
 2    cityCode         1010033 non-null  int64
 3    cityName         1010033 non-null  object
 4    HotelCode        1010033 non-null  int64
 5    HotelName        1010033 non-null  object
 6    HotelRating      1010033 non-null  object
 7    Address          1009931 non-null  object
 8    Attractions      484941 non-null   object
 9    Description      963028 non-null   object
 10   FaxNumber        449686 non-null   object
 11   HotelFacilities  959655 non-null   object
 12   Map              1009103 non-null  object
 13   PhoneNumber      682896 non-null   object
 14   PinCode          979062 non-null   object
 15   HotelWebsiteUrl  759915 non-null   object
dtypes: int64(2), object(14)
memory usage: 123.3+ MB
```

```
In [4]: df.columns
```

```
Out[4]: Index(['countyCode', ' countyName', ' cityCode', ' cityName', ' HotelCode',
       ' HotelName', ' HotelRating', ' Address', ' Attractions',
       ' Description', ' FaxNumber', ' HotelFacilities', ' Map',
       ' PhoneNumber', ' PinCode', ' HotelWebsiteUrl'],
      dtype='object')
```

Как и видим, что присуствует лишний пробел перед названием, устраним и приводим их к нижнему регистру

```
In [5]: # Приведение к нижнему регистру и удаление лишних пробелов в названиях столбцов
df.columns = df.columns.str.strip().str.lower()
```

5. Описательная статистика

```
In [7]: df.describe()
```

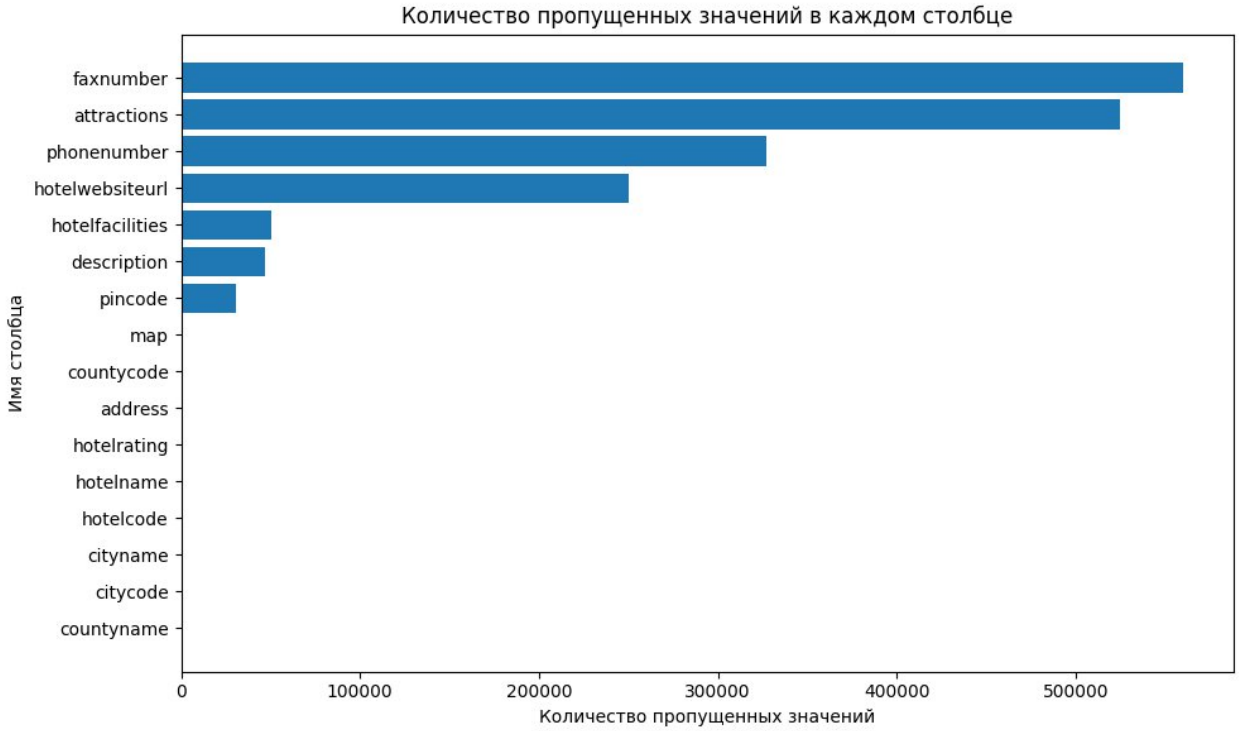
Out[7]:

	citycode	hotelcode
count	1.010033e+06	1.010033e+06
mean	1.274184e+05	2.850143e+06
std	1.353500e+04	1.991153e+06
min	1.000010e+05	1.000000e+06
25%	1.159360e+05	1.310642e+06
50%	1.274950e+05	1.641121e+06
75%	1.388070e+05	5.337335e+06
max	1.518080e+05	6.194373e+06

В датасете слишком много категориальных признаков, поэтому в описательной статистике мало дает информацию. Можно сделать кодирование признаков, т.е. OHE или OH, или попроще - get_dummies()

6. Предобработка данных

6.1. Пропущенные значения



```
In [9]: columns_isnull = [col for col, count in zip(sorted_columns, sorted_missing_counts) if count > 0]
        print(f'Названий столбцов, у которых пропуски:')
        for col in columns_isnull:
            print('\t' + col)
```

```
Названий столбцов, у которых пропуски:
        address
        countycode
        map
        pincode
        description
        hotelfacilities
        hotelwebsiteurl
        phonenumber
        attractions
        faxnumber
```

Пропущенные значения колонки в основном категориальные, поэтому не получится заполнить их медианой. Можно их удалять, но лучше всего заполнить заглушку как - unknown. Если устранить их, то более 50% данных мы потеряем, а это не очень хорошо

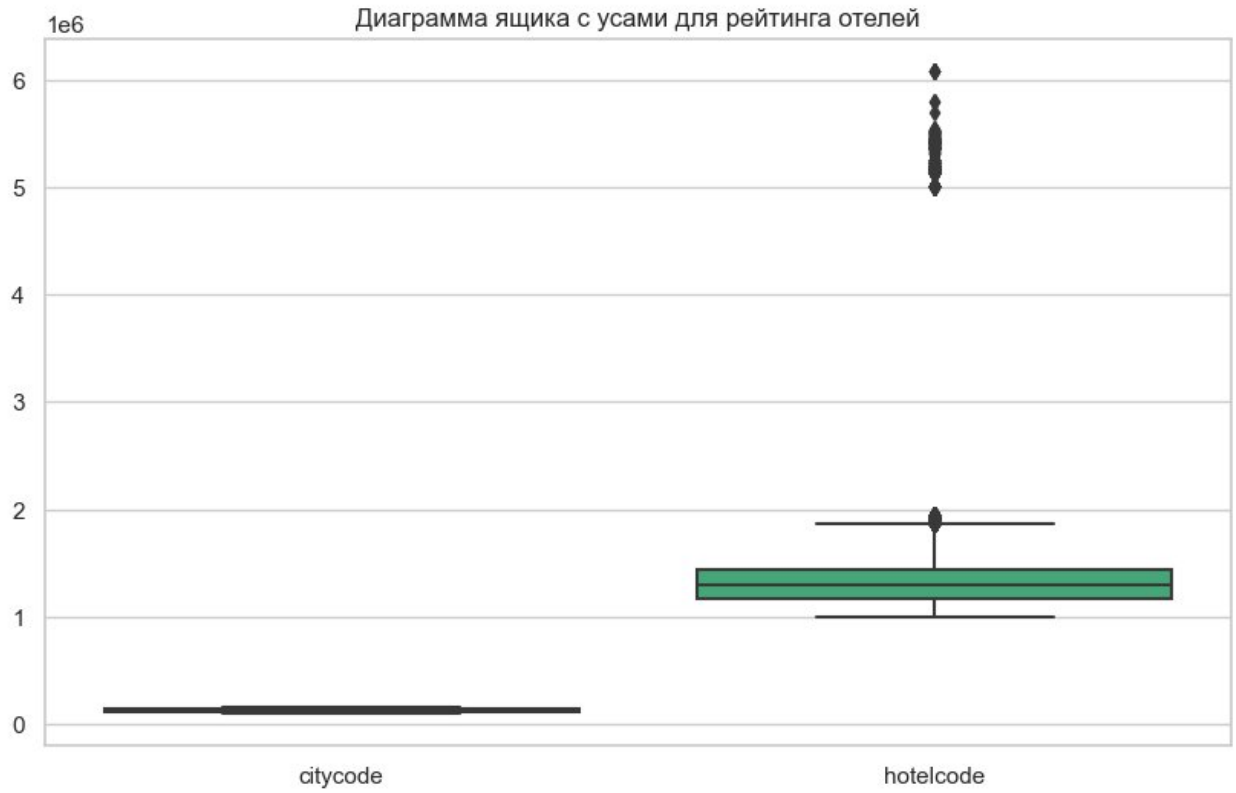
```
In [10]: # Устраняем пропуски заглушками
         # df[columns_isnull] = df[columns_isnull].fillna('unknown')
         df.dropna(inplace=True)
```

6.2.Дубликаты

```
In [11]: # Кол-во дублирующие значения
         df.duplicated().sum()
```

```
Out[11]: 0
```

6.3.Выбросы - Ящик с усами



6.4.Создание новых признаков - высота и ширина

Как и видим, что можно разделить на высоту и ширины с колонки map

In [13]:
```
df[['map']].head()
```

Out[13]:

	map
8	41.34106|19.83108
9	41.32547|19.82503
11	41.33054|19.82281
17	41.31723|19.82361
26	41.332199|19.818794

In [14]:
```
# Разделение столбца 'map' на два отдельных столбца
df[['latitude', 'longitude']] = df['map'].str.split('|', expand=True)

# Приведение к числовому формату
df['latitude'] = pd.to_numeric(df['latitude'])
df['longitude'] = pd.to_numeric(df['longitude'])

# Удаление столбца 'map
df = df.drop('map', axis=1)
```

In [15]:
```
df[['latitude', 'longitude']].head()
```

Out[15]:

	latitude	longitude
8	41.341060	19.831080
9	41.325470	19.825030
11	41.330540	19.822810
17	41.317230	19.823610
26	41.332199	19.818794

6.5.Преобразование категорий в числа

In [16]:
```
# Создаем словарь для соответствия категорий и их кодов
rating_mapping = {
    'OneStar': 1,
    'TwoStar': 2,
    'ThreeStar': 3,
    'FourStar': 4,
    'FiveStar': 5,
    'All': 6
}

# Присваиваем числовые коды
df['hotelrating_encoded'] = df['hotelrating'].map(rating_mapping)
```

7. Итог

7.1.Предобработка данных

- Присутствовали огромные пропуски, было решено устранять их. Была мысль заполнять их заглушками, но это можно реализовать, если потребуется
- Дубликатов нет
- Присуствует значительное кол-во выбросов - код отеля
- Создали новые признаки - долгота и ширина, разбив с колонки map
- В ходе предобработки данных было вявлено, что есть колонка, которая дает информацию ближайщих достопримчательности отелей, но однако информация носит HTML формата, что на парсинг уходит

много времени и необходимо сопровождать кода. Для тщательного исследования будет полезно

- Также много категориальных признаков, можно преобразовать их в численным через ОНЕ или ОН