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 вычислительная техника»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Технология машинного обучения»

Отчет по лабораторной работе №2
«Обработка пропусков в данных, кодирование категориальных признаков и масштабирование данных»

Выполнил:
студент группы ИУ5-62Б

Ванина П.В.
Подпись и дата:

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю.Е.
Подпись и дата:

Москва, 2022 г.

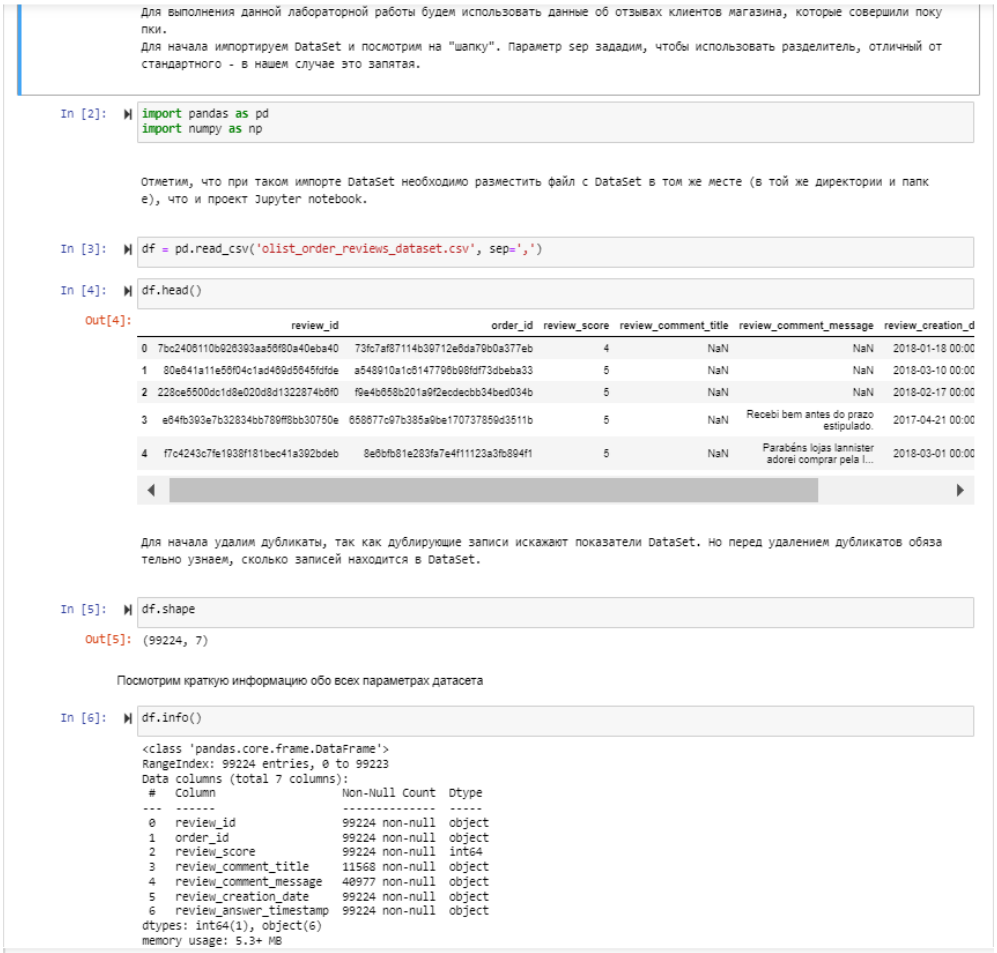
Цель лабораторной работы:

Изучение способов предварительной обработки данных для дальнейшего формирования моделей.

Описание задания:

- Выбрать набор данных (датасет)
- Для выбранного датасета выполнить следующие задачи:
  1. Обработку пропусков в данных
  2. Кодирование категориальных признаков
  3. Масштабирование данных

Выполнение лабораторной работы:



Для выполнения данной лабораторной работы будем использовать данные об отзывах клиентов магазина, которые совершили покупки.
Для начала импортируем DataSet и посмотрим на "шапку". Параметр sep зададим, чтобы использовать разделитель, отличный от стандартного - в нашем случае это запятая.

In [2]:
```
import pandas as pd
import numpy as np
```

Отметим, что при таком импорте DataSet необходимо разместить файл с DataSet в том же месте (в той же директории и папке), что и проект Jupyter notebook.

In [3]:
```
df = pd.read_csv('olist_order_reviews_dataset.csv', sep=',')
```

In [4]:
```
df.head()
```

Out[4]:

	review_id	order_id	review_score	review_comment_title	review_comment_message	review_creation_d
0	7bc2406110b926393aa56f80a40eba40	73fc7af87114b39712e6da79b0a377eb	4	NaN	NaN	2018-01-18 00:00
1	80e641a11e56f04c1ad469d5645fdfde	a548910a1c6147796b98fdf73dbeba33	5	NaN	NaN	2018-03-10 00:00
2	228ce5500dc1d8e020d8d1322874b6f0	f9e4b658b201a9f2ecdecbb34bed034b	5	NaN	NaN	2018-02-17 00:00
3	e64fb393e7b32834bb789ff8bb30750e	658677c97b385a9be170737859d3511b	5	NaN	Recebi bem antes do prazo estipulado.	2017-04-21 00:00
4	f7c4243c7fe1938f181bec41a392bdeb	8e6bfb81e283fa7e4f11123a3fb894f1	5	NaN	Parabéns lojas lannister adorei comprar pela I...	2018-03-01 00:00

Для начала удалим дубликаты, так как дублирующие записи искажают показатели DataSet. Но перед удалением дубликатов обязательно узнаем, сколько записей находится в DataSet.

In [5]:
```
df.shape
```

Out[5]: (99224, 7)

Посмотрим краткую информацию обо всех параметрах датасета

In [6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 99224 entries, 0 to 99223
Data columns (total 7 columns):
 #   Column                  Non-Null Count  Dtype 
---  ------                  --------------  ----- 
 0   review_id               99224 non-null  object
 1   order_id                99224 non-null  object
 2   review_score            99224 non-null  int64 
 3   review_comment_title    11568 non-null  object
 4   review_comment_message  40977 non-null  object
 5   review_creation_date    99224 non-null  object
 6   review_answer_timestamp  99224 non-null  object
dtypes: int64(1), object(6)
memory usage: 5.3+ MB
```

В выбранном датасете находятся преимущественно данные строкового типа, а также целочисленного. Проверим датасет на наличие дубликатов

In [7]:
```
df = df.drop_duplicates()
df.shape
```

Out[7]: (99224, 7)

Видно, что Pandas не нашел ни одного дубликата

Обработка пропусков!
Помним,что если у признака более 70% пропусков, то такой признак удаляют. Поэтому проверим, насколько наши признаки полны.

In [8]:
```
column_values = df[['order_id', 'review_score', 'review_comment_title', 'review_comment_message', 'review_creation_date']].va
unique_values =  pd.unique(column_values)
print(unique_values)
```

```
['73fc7af87114b39712e6da79b0a377eb' 4 nan ...
 '7725825d039fc1f0ceb7635e3f7d9206' '90531360ecb1eec2a1fbb265a0db0508'
 'meu produto chegou e ja tenho que devolver, pois está com defeito , não segurar carga']
```

Существует несколько способов обозначить пропуски, и зачастую создатели датасета не описывают данные в достаточной мере, и определять, как обозначены пропуски, приходится вручную.
Например:

1) NaN / NaT (упрощенно: "не число" / "не время")
2) Пустая ячейка
3) Для числовых признаков – радикальный выброс. К примеру, для столбца "День" это число 999.
4) Маркер или нестандартный символ

Встроенные методы Pandas позволяют с легкостью справиться с первыми двумя разновидностями таких пробелов. Разберемся для начала с категориальными переменными, объединив их в один список.

In [9]:
```
df = df.replace({"День": 999,
                 "order_id": "Неизвестно",
                 "review_score": "Неизвестно",
                 "review_comment_title": "Неизвестно",
                 "review_comment_message": "Неизвестно",
                 "review_creation_date": "Неизвестно",
                 }, np.nan)

df.head()
```

Out[9]:

	review_id	order_id	review_score	review_comment_title	review_comment_message	review_creation_
0	7bc2406110b926393aa56f80a40eba40	73fc7af87114b39712e6da79b0a377eb	4	NaN	NaN	2018-01-18 00:0
1	80e641a11e56f04c1ad469d5645fdfde	a548910a1c6147796b98fdf73dbeba33	5	NaN	NaN	2018-03-10 00:0
2	228ce5500dc1d8e020d8d1322874b6f0	f9e4b658b201a9f2ecdecbb34bed034b	5	NaN	NaN	2018-02-17 00:0
3	e64fb393e7b32834bb789ff8bb30750e	658677c97b385a9be170737859d3511b	5	NaN	Recebi bem antes do prazo estipulado.	2017-04-21 00:0

Среди всех признаков 88% пропусков находится у признака "review_comment_title" и почти 59% - у "review_comment_message", первый признак поделжит удалению.

In [10]:
```
df.isnull().mean() * 100
```

Out[10]:
```
review_id                 0.000000
order_id                  0.000000
review_score              0.000000
review_comment_title     88.341530
review_comment_message   58.702532
review_creation_date      0.000000
review_answer_timestamp   0.000000
dtype: float64
```

In [11]:
```
df = df.drop(columns=['review_comment_title'])
df.head()
```

Out[11]:

	review_id	order_id	review_score	review_comment_message	review_creation_date	review_answer_ti
0	7bc2406110b926393aa56f80a40eba40	73fc7af87114b39712e6da79b0a377eb	4	NaN	2018-01-18 00:00:00	2018-01-18
1	80e641a11e56f04c1ad469d5645fdfde	a548910a1c6147796b98fdf73dbeba33	5	NaN	2018-03-10 00:00:00	2018-03-11
2	228ce5500dc1d8e020d8d1322874b6f0	f9e4b658b201a9f2ecdecbb34bed034b	5	NaN	2018-02-17 00:00:00	2018-02-18
3	e64fb393e7b32834bb789ff8bb30750e	658677c97b385a9be170737859d3511b	5	Recebi bem antes do prazo estipulado.	2017-04-21 00:00:00	2017-04-21
4	f7c4243c7fe1938f181bec41a392bdeb	8e6bfb81e283fa7e4f11123a3fb894f1	5	Parabéns lojas lannister adorei comprar pela I...	2018-03-01 00:00:00	2018-03-02

Процесс обработки пропусков, к счастью, можно сократить с помощью sklearn.impute.SimpleImputer. Мы выбираем все категориальные переменные и применяем стратегию "[вставить вместо пропуска] самое распространенное значение":

In [12]:
```
from sklearn.impute import SimpleImputer

imputer = SimpleImputer(missing_values = np.nan, strategy = 'most_frequent')

df["order_id"] = imputer.fit_transform(df["order_id"].values.reshape(-1,1))[:,0]

df["review_comment_message"] = imputer.fit_transform(df["review_comment_message"].values.reshape(-1,1))[:,0]

df["review_creation_date"] = imputer.fit_transform(df["review_creation_date"].values.reshape(-1,1))[:,0]

df["review_answer_timestamp"] = imputer.fit_transform(df["review_answer_timestamp"].values.reshape(-1,1))[:,0]
```

Подобным образом заполняются пустоты в числовых переменных, только стратегия теперь – "вставить среднее значение".

In [13]:
```
imputer = SimpleImputer(missing_values = np.nan, strategy = 'mean')

df["review_score"] = imputer.fit_transform(df["review_score"].values.reshape(-1,1))[:,0]
```

In [14]:
```
df.head()
```

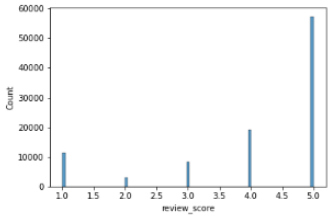
Out[14]:

	review_id	order_id	review_score	review_comment_message	review_creation_date	review_answer_ti
0	7bc2406110b926393aa56f80a40eba40	73fc7af87114b39712e6da79b0a377eb	4.0	Muito bom	2018-01-18 00:00:00	2018-01-18
1	80e641a11e56f04c1ad469d5645fdfde	a548910a1c6147796b98fdf73dbeba33	5.0	Muito bom	2018-03-10 00:00:00	2018-03-11
2	228ce5500dc1d8e020d8d1322874b6f0	f9e4b658b201a9f2ecdecbb34bed034b	5.0	Muito bom	2018-02-17 00:00:00	2018-02-18
3	e64fb393e7b32834bb789ff8bb30750e	658677c97b385a9be170737859d3511b	5.0	Recebi bem antes do prazo estipulado.	2017-04-21 00:00:00	2017-04-21
4	f7c4243c7fe1938f181bec41a392bdeb	8e6bfb81e283fa7e4f11123a3fb894f1	5.0	Parabéns lojas lannister adorei comprar pela I...	2018-03-01 00:00:00	2018-03-02

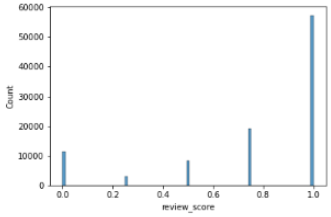
Масштабирование данных

In [15]:
```
import seaborn as sns

sns.histplot(df['review_score'])
```

Out[15]:
```
<AxesSubplot:xlabel='review_score', ylabel='Count'>
```



In [16]:
```
from sklearn.preprocessing import MinMaxScaler, StandardScaler, Normalizer

df['review_score'] = MinMaxScaler().fit_transform(df['review_score'].values.reshape(-1, 1))
sns.histplot(df['review_score']);
```



Кодирование категориальных признаков

In [24]:
```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
le.fit(df.order_id)
df['order_id_le'] = le.transform(df.order_id)
df
```

Out[24]:

review_id	order_id	review_score	review_comment_message	review_creation_date	review_answer_timestamp	Hours	order_id_le
)a40eba40	73fc7af87114b39712e6da79b0a377eb	0.75	Muito bom	2018-01-18 00:00:00	2018-01-18 21:46:59	20	44692
d5645fdfde	a548910a1c6147796b98fdf73dbeba33	1.00	Muito bom	2018-03-10 00:00:00	2018-03-11 03:05:13	20	63412
d22874b6f0	f9e4b658b201a9f2ecdecbb34bed034b	1.00	Muito bom	2018-02-17 00:00:00	2018-02-18 14:36:24	20	96304
9bb30750e	658677c97b385a9be170737859d3511b	1.00	Recebi bem antes do prazo estipulado.	2017-04-21 00:00:00	2017-04-21 22:02:06	20	39089
1a392bdeb	8e6bfb81e283fa7e4f11123a3fb894f1	1.00	Parabéns lojas lannister adorei comprar pela I...	2018-03-01 00:00:00	2018-03-02 10:26:53	20	54600
...	...	...	...	...	...	...	...
ibbf39d7c9	2a8c23fee101d4d5662fa670396eb8da	1.00	Muito bom	2018-07-07 00:00:00	2018-07-14 17:18:30	20	16272
odfdf4ed7a	22ec9f0669f784db00fa86d035cf8602	1.00	Muito bom	2017-12-09 00:00:00	2017-12-11 20:06:42	20	13435
9fd302472	55d4004744368f5571d1f590031933e4	1.00	Excelente mochila, entrega super rápida. Super...	2018-03-22 00:00:00	2018-03-23 09:10:43	20	33099
33eb85149	7725825d039fc1f0ceb7635e3f7d9206	0.75	Muito bom	2018-07-01 00:00:00	2018-07-02 12:59:13	20	45831
ecd4cc548f	90531360ecb1eec2a1fbb265a0db0508	0.00	meu produto chegou e ja tenho que devolver, po...	2017-07-03 00:00:00	2017-07-03 21:01:49	20	55335

In [ ]:
```
```