Московский государственный технический университет им. Н.Э. Баумана
Факультет «Информатика и системы управления»
Кафедра «Системы обработки информации и управления»

Отчет по ЛР №3

по курсу «Технологии машинного обучения»

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

ИСПОЛНИТЕЛЬ:
Аушева Л.И.
Группа ИУ5-61Б
_____

"__"_____2020 г.

ПРЕПОДАВАТЕЛЬ:
Гапанюк Ю.Е.
_____

"__"_____2020 г.

Москва 2020

Цель лабораторной работы:

Изучение способов предварительной обработки данных для дальнейшего формирования моделей.

Задание:

1. Выбрать набор данных (датасет: restaurant-scores-lives-standard.csv), 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:
    a. обработку пропусков в данных;
    b. кодирование категориальных признаков;
    c. масштабирование данных.

Выполнение:

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:
```
import os
import zipfile

DATA_PATH = os.path.join('datasets')

def fetch_data(data_path=DATA_PATH):
    os.makedirs(data_path, exist_ok=True)
    zip_path = os.path.join(data_path, 'restaurant-scores-lives-standard.csv.zip')
    data_zip = zipfile.ZipFile(zip_path)
    data_zip.extractall(path=data_path)
    data_zip.close()
```

In [3]:
```
fetch_data()
```

In [4]:
```
def load_data(data_path=DATA_PATH):
    csv_path = os.path.join(data_path, 'restaurant-scores-lives-standard.csv')
    return pd.read_csv(csv_path)
```

In [5]:
```
data = load_data()
data
```

Out[5]:

	business_id	business_name	business_address	business_city	business_state	business_postal_code	business_latitude	business_longitude	business_lo
0	69618	Fancy Wheatfield Bakery	1362 Stockton St	San Francisco	CA	94133	NaN	NaN	
1	97975	BREADBELLY	1408 Clement St	San Francisco	CA	94118	NaN	NaN	
2	69487	Hakkasan San Francisco	1 Kearny St	San Francisco	CA	94108	NaN	NaN	
3	91044	Chopsticks Restaurant	4615 Mission St	San Francisco	CA	94112	NaN	NaN	
4	85987	Tselogs	552 Jones St	San Francisco	CA	94102	NaN	NaN	
...	...	...	...	...	...	...	...	...	
53968	80305	Snowbird Coffee	1352 A 9th Ave	San Francisco	CA	94110	NaN	NaN	
53969	80233	Buffalo Kitchen	107 Leland Ave	San Francisco	CA	94134	NaN	NaN	
53970	100216	BUNN MIKE	300 DE HARO ST	San Francisco	CA	94103	NaN	NaN	
53971	79430	City Discount Meat & Grocery Market	2298 Mission St	San Francisco	CA	94110	NaN	NaN	
53972	77681	Tart To Tart Inc.	641 Irving St	San Francisco	CA	94122	NaN	NaN	

53973 rows × 17 columns

Обработка пропусков в данных

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 53973 entries, 0 to 53972
Data columns (total 17 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   business_id            53973 non-null  int64
 1   business_name          53973 non-null  object
 2   business_address       53973 non-null  object
 3   business_city          53973 non-null  object
 4   business_state         53973 non-null  object
 5   business_postal_code   52890 non-null  object
 6   business_latitude      29878 non-null  float64
 7   business_longitude     29878 non-null  float64
 8   business_location      29878 non-null  object
 9   business_phone_number  17434 non-null  float64
 10  inspection_id          53973 non-null  object
 11  inspection_date        53973 non-null  object
 12  inspection_score       39859 non-null  float64
 13  inspection_type        53973 non-null  object
 14  violation_id           40511 non-null  object
 15  violation_description  40511 non-null  object
 16  risk_category          40511 non-null  object
dtypes: float64(4), int64(1), object(12)
memory usage: 7.0+ MB
```

In [7]:
```
data.isnull().sum()
```

Out[7]:
```
business_id                 0
business_name               0
business_address            0
business_city               0
business_state              0
business_postal_code     1083
business_latitude       24095
business_longitude      24095
business_location       24095
business_phone_number   36539
inspection_id               0
inspection_date             0
inspection_score        14114
inspection_type             0
violation_id            13462
violation_description   13462
risk_category           13462
dtype: int64
```

In [8]:
```
sample_incomplete_rows = data[data.isnull().any(axis=1)].head()
sample_incomplete_rows
```

Out[8]:

	business_id	business_name	business_address	business_city	business_state	business_postal_code	business_latitude	business_longitude	business_locatio
0	69618	Fancy Wheatfield Bakery	1362 Stockton St	San Francisco	CA	94133	NaN	NaN	Na
1	97975	BREADBELLY	1408 Clement St	San Francisco	CA	94118	NaN	NaN	Na
2	69487	Hakkasan San Francisco	1 Kearny St	San Francisco	CA	94108	NaN	NaN	Na
3	91044	Chopsticks Restaurant	4615 Mission St	San Francisco	CA	94112	NaN	NaN	Na
4	85987	Tselogs	552 Jones St	San Francisco	CA	94102	NaN	NaN	Na

Удалить строки с пропусками в business_latitude

In [9]:
```
sample_incomplete_rows.dropna(subset=['business_latitude'])
```

Out[9]:

	business_id	business_name	business_address	business_city	business_state	business_postal_code	business_latitude	business_longitude	business_location

Удалить столбцы, у которых есть пропуски

In [10]:
```
sample_incomplete_rows.drop("business_latitude", axis=1)
```

Out[10]:

	business_id	business_name	business_address	business_city	business_state	business_postal_code	business_longitude	business_location	business_phone
0	69618	Fancy Wheatfield Bakery	1362 Stockton St	San Francisco	CA	94133	NaN	NaN	
1	97975	BREADBELLY	1408 Clement St	San Francisco	CA	94118	NaN	NaN	1.41
2	69487	Hakkasan San Francisco	1 Kearny St	San Francisco	CA	94108	NaN	NaN	
3	91044	Chopsticks Restaurant	4615 Mission St	San Francisco	CA	94112	NaN	NaN	
4	85987	Tselogs	552 Jones St	San Francisco	CA	94102	NaN	NaN	

Заменить пропуски средним / медианой / самым частым значением

In [11]:
```
mean_ = data['inspection_score'].mean()
sample_incomplete_rows['inspection_score'].fillna(mean_, inplace=True)
sample_incomplete_rows
```

Out[11]:

	business_id	business_name	business_address	business_city	business_state	business_postal_code	business_latitude	business_longitude	business_locatio
0	69618	Fancy Wheatfield Bakery	1362 Stockton St	San Francisco	CA	94133	NaN	NaN	Na
1	97975	BREADBELLY	1408 Clement St	San Francisco	CA	94118	NaN	NaN	Na
2	69487	Hakkasan San Francisco	1 Kearny St	San Francisco	CA	94108	NaN	NaN	Na
3	91044	Chopsticks Restaurant	4615 Mission St	San Francisco	CA	94112	NaN	NaN	Na
4	85987	Tselogs	552 Jones St	San Francisco	CA	94102	NaN	NaN	Na

In [12]:
```
median = data['inspection_score'].median()
sample_incomplete_rows['inspection_score'].fillna(median, inplace=True)
sample_incomplete_rows
```

Out[12]:

	business_id	business_name	business_address	business_city	business_state	business_postal_code	business_latitude	business_longitude	business_locatio
0	69618	Fancy Wheatfield Bakery	1362 Stockton St	San Francisco	CA	94133	NaN	NaN	Na
1	97975	BREADBELLY	1408 Clement St	San Francisco	CA	94118	NaN	NaN	Na
2	69487	Hakkasan San Francisco	1 Kearny St	San Francisco	CA	94108	NaN	NaN	Na
3	91044	Chopsticks Restaurant	4615 Mission St	San Francisco	CA	94112	NaN	NaN	Na
4	85987	Tselogs	552 Jones St	San Francisco	CA	94102	NaN	NaN	Na

In [13]:
```
data['inspection_type'].value_counts()
```

Out[13]:
```
Routine - Unscheduled                39956
Reinspection/Followup                 6695
Complaint                             2379
New Ownership                         1787
New Construction                       905
Non-inspection site visit              843
New Ownership - Followup               512
Structural Inspection                  360
Complaint Reinspection/Followup        232
Foodborne Illness Investigation        217
Routine - Scheduled                     76
Special Event                            6
Multi-agency Investigation               3
Administrative or Document Review        2
Name: inspection_type, dtype: int64
```

In [14]:
```
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
```

Кодирование категорий целочисленными значениями - label encoding

In [15]:
```
le = LabelEncoder()
cat_enc_le = le.fit_transform(data['inspection_type'])
cat_enc_le
```

Out[15]:
```
array([ 1, 11, 11, ..., 11, 11,  9])
```

In [16]:
```
np.unique(cat_enc_le)
```

Out[16]:
```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13])
```

In [17]:
```
le.inverse_transform([0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13])
```

Out[17]:
```
array(['Administrative or Document Review', 'Complaint',
       'Complaint Reinspection/Followup',
       'Foodborne Illness Investigation', 'Multi-agency Investigation',
       'New Construction', 'New Ownership', 'New Ownership - Followup',
       'Non-inspection site visit', 'Reinspection/Followup',
       'Routine - Scheduled', 'Routine - Unscheduled', 'Special Event',
       'Structural Inspection'], dtype=object)
```

In [18]:
```
data['inspection_type_le'] = cat_enc_le
data
```

Out[18]:

	business_id	business_name	business_address	business_city	business_state	business_postal_code	business_latitude	business_longitude	business_lo
0	69618	Fancy Wheatfield Bakery	1362 Stockton St	San Francisco	CA	94133	NaN	NaN	
1	97975	BREADBELLY	1408 Clement St	San Francisco	CA	94118	NaN	NaN	
2	69487	Hakkasan San Francisco	1 Kearny St	San Francisco	CA	94108	NaN	NaN	
3	91044	Chopsticks Restaurant	4615 Mission St	San Francisco	CA	94112	NaN	NaN	
4	85987	Tselogs	552 Jones St	San Francisco	CA	94102	NaN	NaN	
...	...	...	...	...	...	...	...	...	
53968	80305	Snowbird Coffee	1352 A 9th Ave	San Francisco	CA	94110	NaN	NaN	
53969	80233	Buffalo Kitchen	107 Leland Ave	San Francisco	CA	94134	NaN	NaN	
53970	100216	BUNN MIKE	300 DE HARO ST	San Francisco	CA	94103	NaN	NaN	
53971	79430	City Discount Meat & Grocery Market	2298 Mission St	San Francisco	CA	94110	NaN	NaN	
53972	77681	Tart To Tart Inc.	641 Irving St	San Francisco	CA	94122	NaN	NaN	

53973 rows × 18 columns

Кодирование категорий наборами бинарных значений - one-hot encoding

In [19]:
```
ohe = OneHotEncoder()
cat_ohe = ohe.fit_transform(data[['inspection_type']])
```

In [20]:
```
ohe.categories_
```

Out[20]:
```
[array(['Administrative or Document Review', 'Complaint',
        'Complaint Reinspection/Followup',
        'Foodborne Illness Investigation', 'Multi-agency Investigation',
        'New Construction', 'New Ownership', 'New Ownership - Followup',
        'Non-inspection site visit', 'Reinspection/Followup',
        'Routine - Scheduled', 'Routine - Unscheduled', 'Special Event',
        'Structural Inspection'], dtype=object)]
```

In [21]:
```
cat_ohe.toarray()
```

Out[21]:
```
array([[0., 1., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 1., 0., 0.],
       [0., 0., 0., ..., 1., 0., 0.],
       ...,
       [0., 0., 0., ..., 1., 0., 0.],
       [0., 0., 0., ..., 1., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.]])
```

Pandas get_dummies - быстрый вариант one-hot кодирования

In [22]:
```
pd.get_dummies(data['inspection_type'])
```

Out[22]:

	Administrative or Document Review	Complaint	Complaint Reinspection/Followup	Foodborne Illness Investigation	Multi-agency Investigation	New Construction	New Ownership	New Ownership - Followup	Non-inspection site visit	Reinspection/Followup	S
0	0	1	0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	0	1	0	
4	0	0	0	0	0	0	0	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	
53968	0	0	0	0	0	0	0	0	0	0	
53969	0	0	0	0	0	0	0	0	0	0	
53970	0	0	0	0	0	0	0	0	0	0	
53971	0	0	0	0	0	0	0	0	0	0	
53972	0	0	0	0	0	0	0	0	0	1	

53973 rows × 14 columns

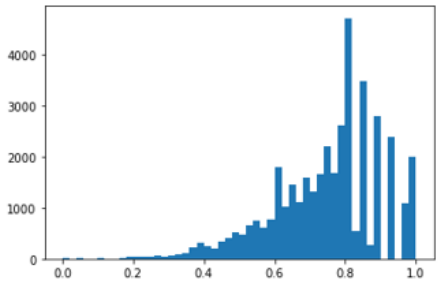
In [23]:
```
from sklearn.preprocessing import MinMaxScaler, StandardScaler, Normalizer
```

MinMax масштабирование

In [24]:
```
d_1 = data.dropna(subset=['inspection_score'])
```

In [25]:
```
sc1 = MinMaxScaler()
sc1_data = sc1.fit_transform(data[['inspection_score']])
```

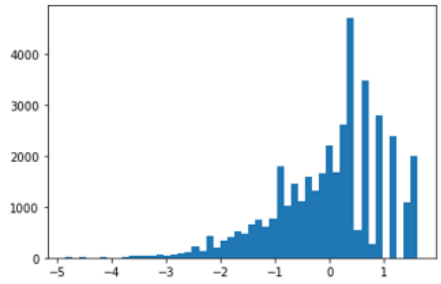
In [30]:
```
plt.hist(sc1_data, 50)
plt.show()
```



Масштабирование данных на основе Z-оценки - StandardScaler

In [31]:
```
sc2 = StandardScaler()
sc2_data = sc2.fit_transform(data[['inspection_score']])
```

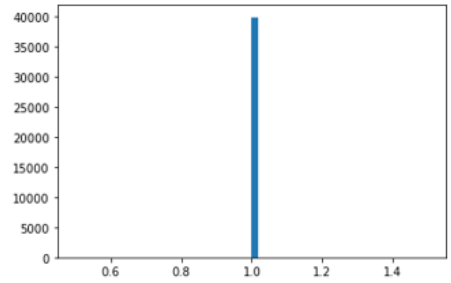
In [33]:
```
plt.hist(sc2_data, 50)
plt.show()
```



Нормализация данных

In [34]:
```
sc3 = Normalizer()
sc3_data = sc3.fit_transform(d_1[['inspection_score']])
```

In [35]:
```
plt.hist(sc3_data, 60)
plt.show()
```



Вывод:

Изучила способы предварительной обработки данных для дальнейшего формирования моделей.