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

Отчёт по лабораторной работе №2

По дисциплине:
«Технологии машинного обучения»

Выполнил:

Студент группы ИУ5-65Б **Петренко С.С**

(Подпись, дата) (Фамилия И.О.)

Проверил:

Гапанюк Ю. Е.

(Подпись, дата) (Фамилия И.О.)

Москва, 2021

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.

ЛР №2

Импорт библиотек

In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from pandas.plotting import scatter_matrix
import warnings
warnings.filterwarnings('ignore')
sns.set(style="ticks")
%matplotlib inline
```

In [2]:
```
data = pd.read_csv('country_vaccinations.csv')
```

In [3]:
```
data.head()
```

Out[3]:

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated
0	Albania	ALB	2021-01-10	0.0	0.0	NaN
1	Albania	ALB	2021-01-11	NaN	NaN	NaN
2	Albania	ALB	2021-01-12	128.0	128.0	NaN
3	Albania	ALB	2021-01-13	188.0	188.0	NaN
4	Albania	ALB	2021-01-14	266.0	266.0	NaN

In [4]:
```
data.dtypes
```

Out[4]:
```
country                              object
iso_code                             object
date                                 object
total_vaccinations                  float64
people_vaccinated                   float64
people_fully_vaccinated             float64
daily_vaccinations_raw              float64
daily_vaccinations                  float64
total_vaccinations_per_hundred      float64
people_vaccinated_per_hundred       float64
people_fully_vaccinated_per_hundred float64
daily_vaccinations_per_million      float64
vaccines                             object
source_name                          object
source_website                       object
dtype: object
```

In [5]:
```
data.isnull().sum()
# проверим есть ли пропущенные значения
```

Out[5]:
```
country                                 0
iso_code                              272
date                                    0
total_vaccinations                   1214
people_vaccinated                    1615
people_fully_vaccinated              2277
daily_vaccinations_raw               1583
daily_vaccinations                    135
total_vaccinations_per_hundred       1214
people_vaccinated_per_hundred        1615
people_fully_vaccinated_per_hundred  2277
daily_vaccinations_per_million        135
vaccines                                0
source_name                             0
source_website                          0
dtype: int64
```

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3555 entries, 0 to 3554
Data columns (total 15 columns):
 #   Column                                Non-Null Count  Dtype  
---  ------                                --------------  -----  
 0   country                               3555 non-null   object 
 1   iso_code                              3283 non-null   object 
 2   date                                  3555 non-null   object 
 3   total_vaccinations                    2341 non-null   float64
 4   people_vaccinated                     1940 non-null   float64
 5   people_fully_vaccinated               1278 non-null   float64
 6   daily_vaccinations_raw                1972 non-null   float64
 7   daily_vaccinations                    3420 non-null   float64
 8   total_vaccinations_per_hundred        2341 non-null   float64
 9   people_vaccinated_per_hundred         1940 non-null   float64
 10  people_fully_vaccinated_per_hundred   1278 non-null   float64
 11  daily_vaccinations_per_million        3420 non-null   float64
 12  vaccines                              3555 non-null   object 
 13  source_name                           3555 non-null   object 
 14  source_website                        3555 non-null   object 
dtypes: float64(9), object(6)
memory usage: 416.7+ KB
```

Обработка пропусков

In [7]:
```
# Удаляем столбцы, которые не несут значимой информации
data.drop(['source_name','source_website'], axis = 1, inplace = True)
```

In [8]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3555 entries, 0 to 3554
Data columns (total 13 columns):
 #   Column                               Non-Null Count  Dtype  
---  ------                               --------------  -----  
 0   country                              3555 non-null   object 
 1   iso_code                             3283 non-null   object 
 2   date                                 3555 non-null   object 
 3   total_vaccinations                   2341 non-null   float64
 4   people_vaccinated                    1940 non-null   float64
 5   people_fully_vaccinated              1278 non-null   float64
 6   daily_vaccinations_raw               1972 non-null   float64
 7   daily_vaccinations                   3420 non-null   float64
 8   total_vaccinations_per_hundred       2341 non-null   float64
 9   people_vaccinated_per_hundred        1940 non-null   float64
 10  people_fully_vaccinated_per_hundred  1278 non-null   float64
 11  daily_vaccinations_per_million       3420 non-null   float64
 12  vaccines                             3555 non-null   object 
dtypes: float64(9), object(4)
memory usage: 361.2+ KB
```

In [9]:
```
# Заполняем отсутствующие значения
data['total_vaccinations'] = data['total_vaccinations'].replace(0,np.nan)
data['total_vaccinations'] = data['total_vaccinations'].fillna(data['total_
```

In [10]:
```
data.head()
```

Out[10]:

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated
0	Albania	ALB	2021-01-10	1.508878e+06	0.0	NaN
1	Albania	ALB	2021-01-11	1.508878e+06	NaN	NaN
2	Albania	ALB	2021-01-12	1.280000e+02	128.0	NaN
3	Albania	ALB	2021-01-13	1.880000e+02	188.0	NaN
4	Albania	ALB	2021-01-14	2.660000e+02	266.0	NaN

In [11]:
```
data.isnull().sum()
# проверим есть ли пропущенные значения
```

Out[11]:
```
country                                0
iso_code                             272
date                                   0
total_vaccinations                     0
people_vaccinated                   1615
people_fully_vaccinated             2277
daily_vaccinations_raw              1583
daily_vaccinations                   135
total_vaccinations_per_hundred      1214
people_vaccinated_per_hundred       1615
people_fully_vaccinated_per_hundred 2277
daily_vaccinations_per_million       135
vaccines                               0
dtype: int64
```

ЛР №2

Импорт библиотек

In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn.impute import SimpleImputer
from sklearn.model_selection import train_test_split
```

In [2]:
```
data = pd.read_csv('train.csv')
```

In [3]:
```
data.head()
```

Out[3]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Far
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.250
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.283
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.925
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.100
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.050

In [4]:
```
data['Embarked'].value_counts()
```

Out[4]:
```
S    644
C    168
Q     77
Name: Embarked, dtype: int64
```

In [5]:
```
# Кодируем признаки Pclass и Embarked в отдельные столбцы
data = pd.get_dummies(data, columns=['Pclass','Embarked'])
```

In [6]:
```
# Пол кодируем в 1/0
data['IsMale']=data.Sex.replace({'female':0,'male':1})
data.drop('Sex', axis = 1, inplace = True)
```

In [7]:
```
data.head()
```

Out[7]:

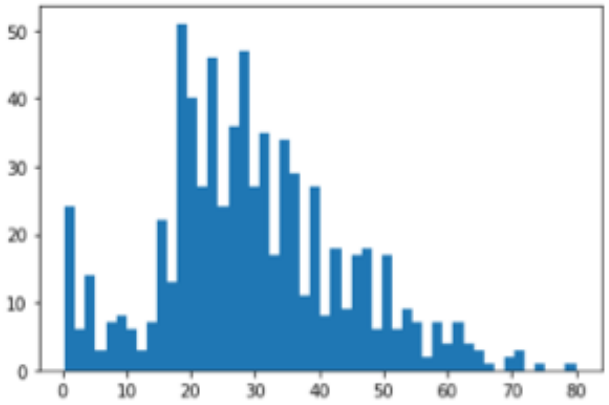
	PassengerId	Survived	Name	Age	SibSp	Parch	Ticket	Fare	Cabin	Pclass
0	1	0	Braund, Mr. Owen Harris	22.0	1	0	A/5 21171	7.2500	NaN	
1	2	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	38.0	1	0	PC 17599	71.2833	C85	
2	3	1	Heikkinen, Miss. Laina	26.0	0	0	STON/O2. 3101282	7.9250	NaN	
3	4	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	35.0	1	0	113803	53.1000	C123	
4	5	0	Allen, Mr. William Henry	35.0	0	0	373450	8.0500	NaN	

Масштабирование значений

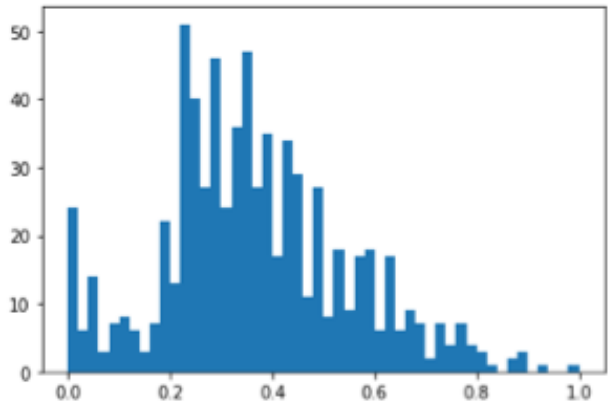
In [8]:
```
from sklearn.preprocessing import StandardScaler, MinMaxScaler, StandardSc
```

In [9]:
```
sc1 = MinMaxScaler()
sc1_data = sc1.fit_transform(data[['Age']])
```

In [10]:
```
plt.hist(data['Age'], 50)
plt.show()
```



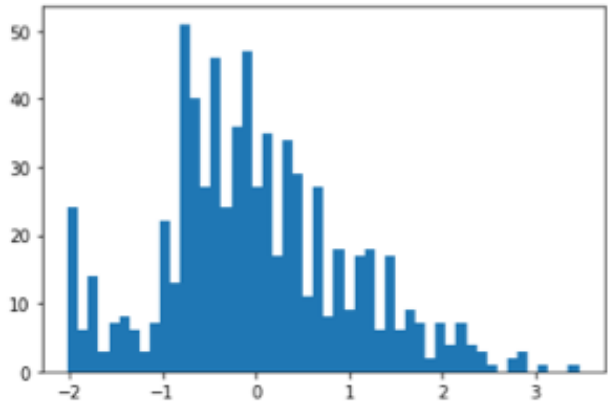
In [11]:
```
plt.hist(sc1_data, 50)
plt.show()
```



In [12]:
```
# Удаляем столбцы, которые не несут значимой информации
data.drop(['Cabin','Name','Ticket'], axis = 1, inplace = True)
```

In [13]:
```
sc2 = StandardScaler()
sc2_data = sc2.fit_transform(data[['Age']])
```

In [14]:
```
plt.hist(sc2_data, 50)
plt.show()
```



In [15]:
```
data.head()
```

Out[15]:

	PassengerId	Survived	Age	SibSp	Parch	Fare	Pclass_1	Pclass_2	Pclass_3	Emba
0	1	0	22.0	1	0	7.2500	0	0	1	
1	2	1	38.0	1	0	71.2833	1	0	0	
2	3	1	26.0	0	0	7.9250	0	0	1	
3	4	1	35.0	1	0	53.1000	1	0	0	
4	5	0	35.0	0	0	8.0500	0	0	1	