МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ Н.Э. БАУМАНА

Факультет Информатика и системы управления

Кафедра Системы обработки информации и управления (ИУ5)

Технологии машинного обучения

Отчет по лабораторной работе №2

Выполнил: Торжков Максим Сергеевич

Группа: ИУ5-61Б

Преподаватель: Гапанюк Юрий Евгеньевич

Дата: 22.03.21

Подпись:

Москва, 2021 г.

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 [8]:
```
data = pd.read_csv('vgsales.csv')
```

In [9]:
```
data.head()
```

Out[9]:

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_S
0	1	Wii Sports	Wii	2006.0	Sports	Nintendo	41.49	29.02	3.77	
1	2	Super Mario Bros.	NES	1985.0	Platform	Nintendo	29.08	3.58	6.81	
2	3	Mario Kart Wii	Wii	2008.0	Racing	Nintendo	15.85	12.88	3.79	
3	4	Wii Sports Resort	Wii	2009.0	Sports	Nintendo	15.75	11.01	3.28	
4	5	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	Nintendo	11.27	8.89	10.22	

In [10]:
```
data.dtypes
```

Out[10]:
```
Rank              int64
Name             object
Platform         object
Year            float64
Genre            object
Publisher        object
NA_Sales        float64
EU_Sales        float64
JP_Sales        float64
Other_Sales     float64
Global_Sales    float64
dtype: object
```

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

Out[11]:
```
Rank              0
Name              0
Platform          0
Year            271
Genre             0
Publisher        58
NA_Sales          0
EU_Sales          0
```

```
JP_Sales          0
Other_Sales       0
Global_Sales      0
dtype: int64
```

In [12]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16598 entries, 0 to 16597
Data columns (total 11 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Rank          16598 non-null  int64
 1   Name          16598 non-null  object
 2   Platform      16598 non-null  object
 3   Year          16327 non-null  float64
 4   Genre         16598 non-null  object
 5   Publisher     16540 non-null  object
 6   NA_Sales      16598 non-null  float64
 7   EU_Sales      16598 non-null  float64
 8   JP_Sales      16598 non-null  float64
 9   Other_Sales   16598 non-null  float64
 10  Global_Sales  16598 non-null  float64
dtypes: float64(6), int64(1), object(4)
memory usage: 1.4+ MB
```

Обработка пропусков

In [13]:
```
# Удаляем столбцы, которые не несут значимой информации
data.drop(['Rank','Other_Sales'], axis = 1, inplace = True)
```

In [14]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16598 entries, 0 to 16597
Data columns (total 9 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Name          16598 non-null  object
 1   Platform      16598 non-null  object
 2   Year          16327 non-null  float64
 3   Genre         16598 non-null  object
 4   Publisher     16540 non-null  object
 5   NA_Sales      16598 non-null  float64
 6   EU_Sales      16598 non-null  float64
 7   JP_Sales      16598 non-null  float64
 8   Global_Sales  16598 non-null  float64
dtypes: float64(5), object(4)
memory usage: 1.1+ MB
```

In [15]:
```
# Заполняем отсутствующие значения
data['Year'] = data['Year'].replace(0,np.nan)
data['Year'] = data['Year'].fillna(data['Year'].mean())
```

In [16]:
```
data.head()
```

Out[16]:

	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Global_Sales
0	Wii Sports	Wii	2006.0	Sports	Nintendo	41.49	29.02	3.77	82.74
1	Super Mario Bros.	NES	1985.0	Platform	Nintendo	29.08	3.58	6.81	40.24
2	Mario Kart Wii	Wii	2008.0	Racing	Nintendo	15.85	12.88	3.79	35.82

	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Global_Sales
3	Wii Sports Resort	Wii	2009.0	Sports	Nintendo	15.75	11.01	3.28	33.00
4	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	Nintendo	11.27	8.89	10.22	31.37

In [17]:
```
data.isnull().sum()
# проверим есть ли пропущенные значения
```

Out[17]:
```
Name            0
Platform        0
Year            0
Genre           0
Publisher      58
NA_Sales        0
EU_Sales        0
JP_Sales        0
Global_Sales    0
dtype: int64
```

In [ ]:

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
```

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

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

Out[3]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	bachelor's degree	standard	none	72	72	74
1	female	group C	some college	standard	completed	69	90	88
2	female	group B	master's degree	standard	none	90	95	93
3	male	group A	associate's degree	free/reduced	none	47	57	44
4	male	group C	some college	standard	none	76	78	75

In [4]:
```
data['race/ethnicity'].value_counts()
```

Out[4]:
```
group C    319
group D    262
group B    190
group E    140
group A     89
Name: race/ethnicity, dtype: int64
```

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

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

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

Out[7]:

	parental level of education	test preparation course	math score	reading score	writing score	lunch_free/reduced	lunch_standard	race/ethnici
0	bachelor's degree	none	72	72	74	0	1	
1	some college	completed	69	90	88	0	1	
2	master's degree	none	90	95	93	0	1	

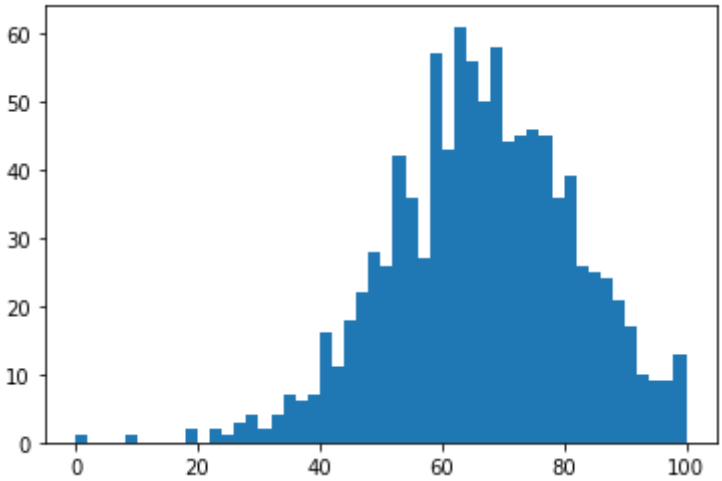
	parental level of education	test preparation course	math score	reading score	writing score	lunch_free/reduced	lunch_standard	race/ethnici
3	associate's degree	none	47	57	44	1	0	
4	some college	none	76	78	75	0	1	

Масштабирование значений

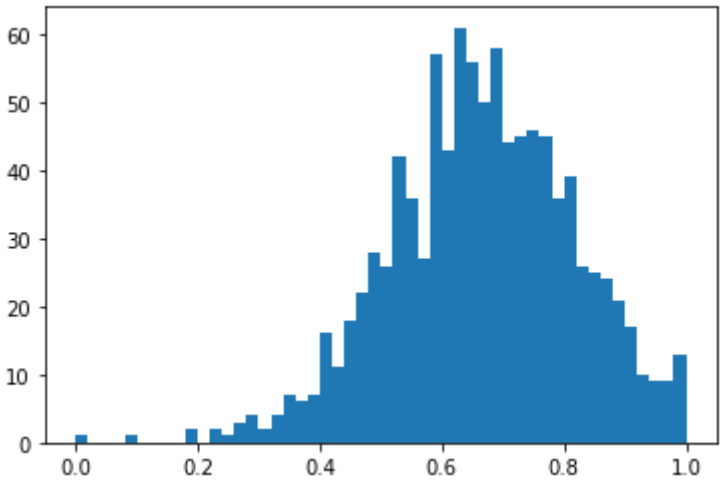
In [8]:
```
from sklearn.preprocessing import StandardScaler, MinMaxScaler, StandardScaler, Norm
```

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

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



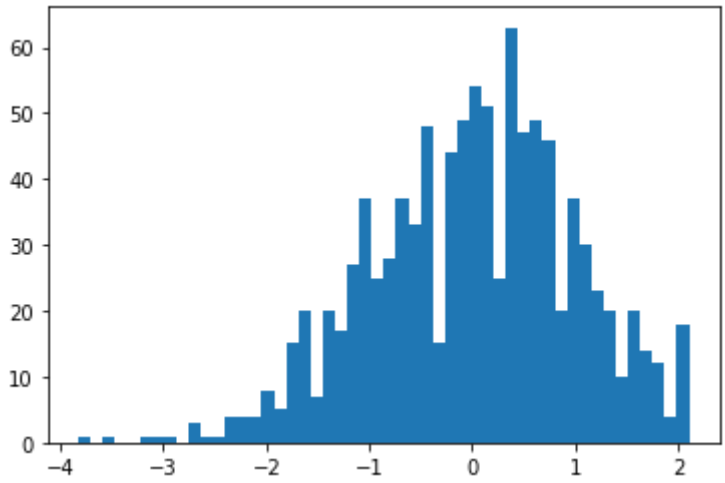
In [11]:
```
plt.hist(sc1_data, 50)
plt.show()
```



In [12]:
```
# Удаляем столбцы, которые не несут значимой информации
data.drop(['test preparation course','parental level of education'], axis = 1, inpla
```

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

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



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

Out[15]:

	math score	reading score	writing score	lunch_free/reduced	lunch_standard	race/ethnicity_group A	race/ethnicity_
0	72	72	74	0	1	0	
1	69	90	88	0	1	0	
2	90	95	93	0	1	0	
3	47	57	44	1	0	1	
4	76	78	75	0	1	0	

In [ ]: