Рубежный контроль №1

Филатова Анастасия

ИУ5-13М

Вариант 16

Задача №16.

Для набора данных проведите нормализацию для одного (произвольного) числового признака с использованием преобразования Бокса-Кокса (Box-Cox transformation).

Задача №36.

Для набора данных проведите процедуру отбора признаков (feature selection). Используйте класс SelectKBest для 5 лучших признаков, и метод, основанный на взаимной информации.

```
!pip install pandas
!pip install seaborn
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: pandas in /usr/local/lib/python3.9/dist-packages (1.4.4)
Requirement already satisfied: numpy>=1.18.5 in /usr/local/lib/python3.9/dist-packages (from pandas) (1.22.4)
Requirement already satisfied: python-dateutil>=2.8.1 in /usr/local/lib/python3.9/dist-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.9/dist-packages (from pandas) (2022.7.1)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.9/dist-packages (from python-dateutil>=2.8.1->pandas) (1.16.0)
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: seaborn in /usr/local/lib/python3.9/dist-packages (0.12.2)
Requirement already satisfied: matplotlib!=3.6.1,>=3.1 in /usr/local/lib/python3.9/dist-packages (from seaborn) (3.7.1)
Requirement already satisfied: numpy!=1.24.0,>=1.17 in /usr/local/lib/python3.9/dist-packages (from seaborn) (1.22.4)
Requirement already satisfied: pandas>=0.25 in /usr/local/lib/python3.9/dist-packages (from seaborn) (1.4.4)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.9/dist-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (1.0.7)
Requirement already satisfied: importlib-resources>=3.2.0 in /usr/local/lib/python3.9/dist-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (5.12.0)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.9/dist-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (4.39.3)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.9/dist-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (8.4.0)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.9/dist-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (2.8.2)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.9/dist-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (0.11.0)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.9/dist-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (23.0)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.9/dist-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (1.4.4)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.9/dist-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (3.0.9)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.9/dist-packages (from pandas>=0.25->seaborn) (2022.7.1)
Requirement already satisfied: zipp>=3.1.0 in /usr/local/lib/python3.9/dist-packages (from importlib-resources>=3.2.0->matplotlib!=3.6.1,>=3.1->seaborn) (3.15.0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.9/dist-packages (from python-dateutil>=2.7->matplotlib!=3.6.1,>=3.1->seaborn) (1.16.0)
```

```
[ ] df = pd.read_csv('/content/movies.csv')
```

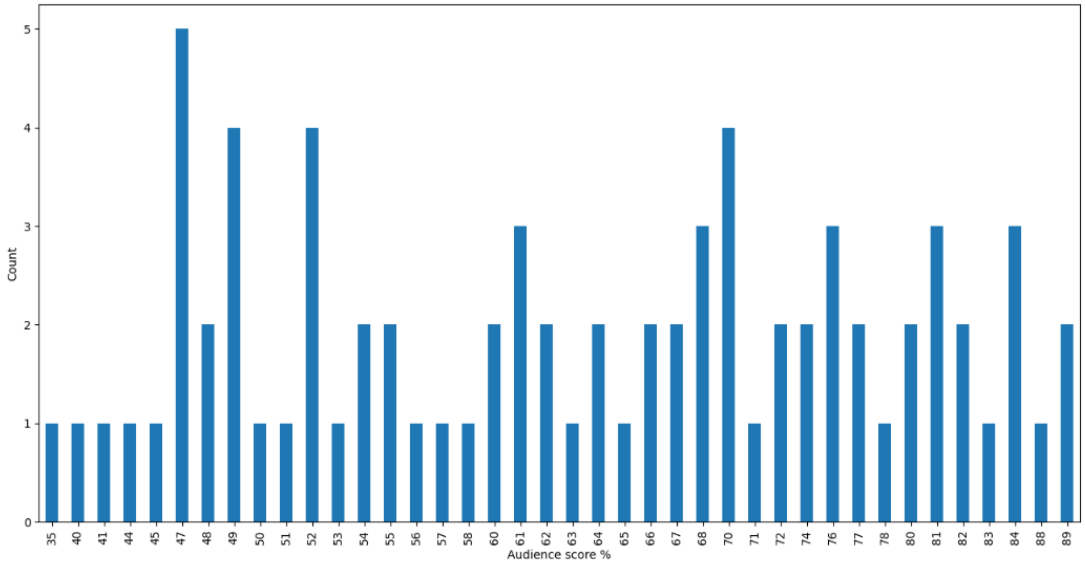
```
df.isna().mean()
```

```
Film                0.0
Genre               0.0
Lead Studio         0.0
Audience score %    0.0
Profitability       0.0
Rotten Tomatoes %   0.0
Worldwide Gross     0.0
Year                0.0
dtype: float64
```

Для набора данных проведите нормализацию для одного (произвольного) числового признака с использованием преобразования Бокса-Кокса (Box-Cox transformation).

```
[ ] df.groupby('Audience score %')['Year'].agg('count').plot(kind = 'bar',figsize=( 16 , 8 ), ylabel = 'Count')
```

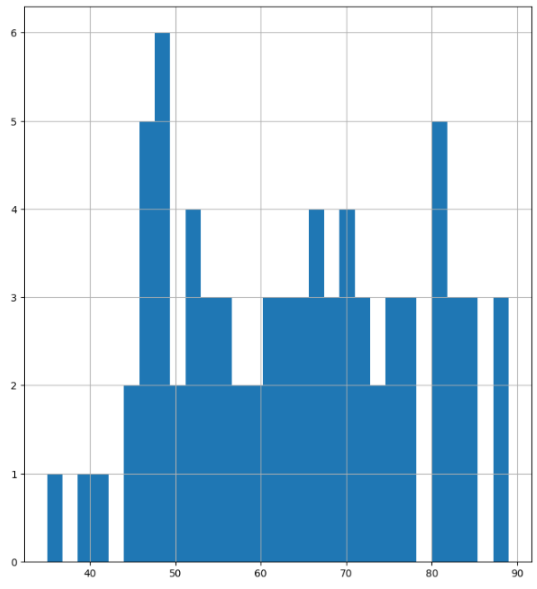
```
[ ] <Axes: xlabel='Audience score %', ylabel='Count'>
```

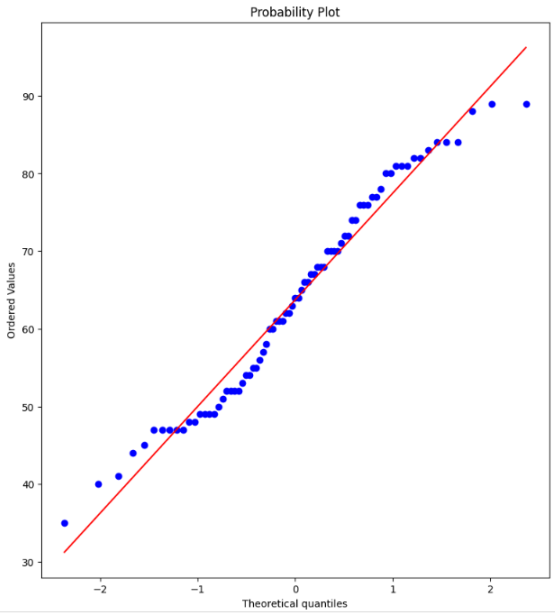



```
[ ] import scipy.stats as stats
    def diagnostic_plots(df, variable):
        plt.figure(figsize=(20,10))
        # гистограмма
        plt.subplot(1, 2, 1)
        df[variable].hist(bins=30)
        ## Q-Q plot
        plt.subplot(1, 2, 2)
        stats.probplot(df[variable], dist="norm", plot=plt)
        plt.show()
```

```
[ ]
    df['Rating_boxcox'], param = stats.boxcox(df['Audience score %'])
    print('Оптимальное значение λ = {}'.format(param))
    diagnostic_plots(df, 'Audience score %')

    Оптимальное значение λ = 0.6849714080298129
```





Для набора данных проведите процедуру отбора признаков (feature selection). Используйте класс SelectKBest для 5 лучших признаков, и метод, основанный на взаимной информации.

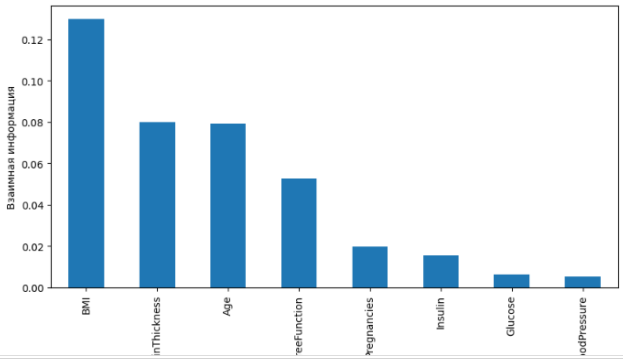
```
[ ] from sklearn.feature_selection import mutual_info_classif, mutual_info_regression, f_regression
    from sklearn.feature_selection import SelectKBest, SelectPercentile
```

```
[ ] df2= pd.read_csv('/content/diabetes.csv')
```

```
[ ] dfX=df2[['Pregnancies','BloodPressure','SkinThickness','BMI','DiabetesPedigreeFunction','Glucose', 'Insulin', 'Age']]
    dfY=df2[['Outcome']]
    df3=df2.drop(columns= 'Outcome')
    df2_feature_names= list(df3.columns)
```

```
[ ] mi = mutual_info_regression(dfX, dfY)
    mi = pd.Series(mi)
    mi.index = df2_feature_names
    mi.sort_values(ascending=False).plot.bar(figsize=(10,5))
    plt.ylabel('Взаимная информация')
```

```
/usr/local/lib/python3.9/dist-packages/sklearn/utils/validation.py:1143: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example
  y = column_or_1d(y, warn=True)
Text(0, 0.5, 'Взаимная информация')
```




```
[ ] sel_mi = SelectKBest(mutual_info_regression, k=5).fit(dfX, dfY)
    list(zip(df2, sel_mi.get_support()))
```

```
/usr/local/lib/python3.9/dist-packages/sklearn/utils/validation.py:1143: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
[('Pregnancies', True),
 ('Glucose', False),
 ('BloodPressure', True),
 ('SkinThickness', True),
 ('Insulin', False),
 ('BMI', True),
 ('DiabetesPedigreeFunction', False),
 ('Age', True)]
```