РК1

Вариант 15.

- Студент – Чупис Сергей Александрович
- Группа – ИУ5-21М
- Вариант – 15

Датасет: https://www.kaggle.com/datasets/mylesoneill/world-university-rankings

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

[7]:
```
data = pd.read_csv('cwurData.csv', sep=",")
data.head()
```

[7]:

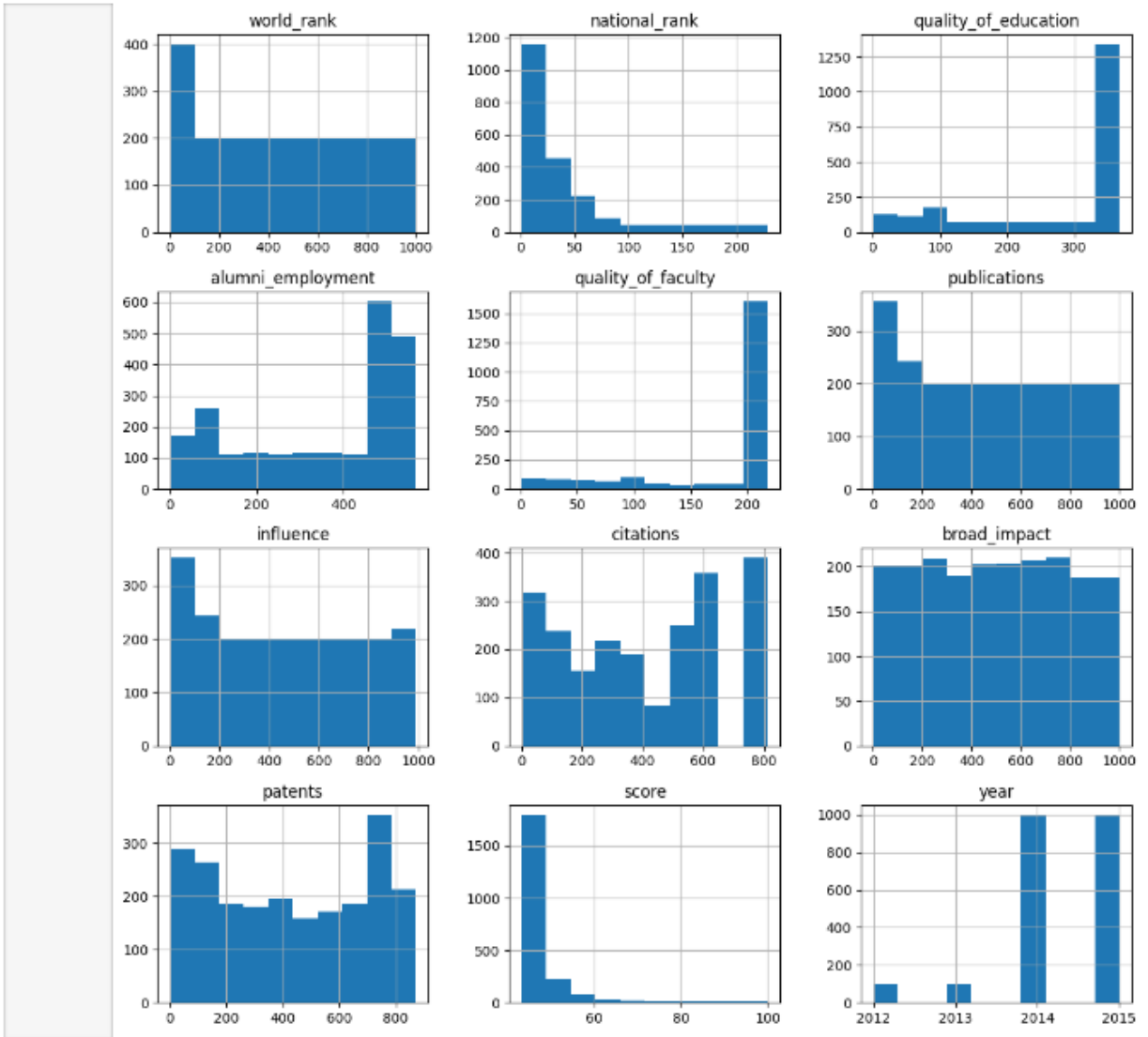
	world_rank	institution	country	national_rank	quality_of_education	alumni_employ
0	1	Harvard University	USA	1	7	
1	2	Massachusetts Institute of Technology	USA	2	9	
2	3	Stanford University	USA	3	17	
3	4	University of Cambridge	United Kingdom	1	10	
4	5	California Institute of Technology	USA	4	2	

```
world rank - мировой рейтинг университета
institution - название университета
country - страна, в которой расположен университет
national rank - рейтинг университета в стране его нахождения
quality of education - рейтинг качества образования
quality of faculty - рейтинг качества процессорско-преподавательского состава
publications - рейтинг публикаций
infuence - рейтинг влияния
citations - количество студентов в университете
broad impact - рейтинг за широкое влияние
patents - рейтинг за патенты
score - обищй балл, используемый для определения мирового рейтинга
year - год рейтинга (с 2012 по 2015 год)
```

[3]:
```
data.dtypes
```

[3]:
```
world_rank              int64
institution            object
country                object
national_rank           int64
quality_of_education    int64
alumni_employment       int64
quality_of_faculty      int64
publications            int64
influence               int64
citations               int64
broad_impact          float64
patents                 int64
score                 float64
year                    int64
dtype: object
```

[4]:
```
data.hist(figsize=(13,13))
plt.show()
```

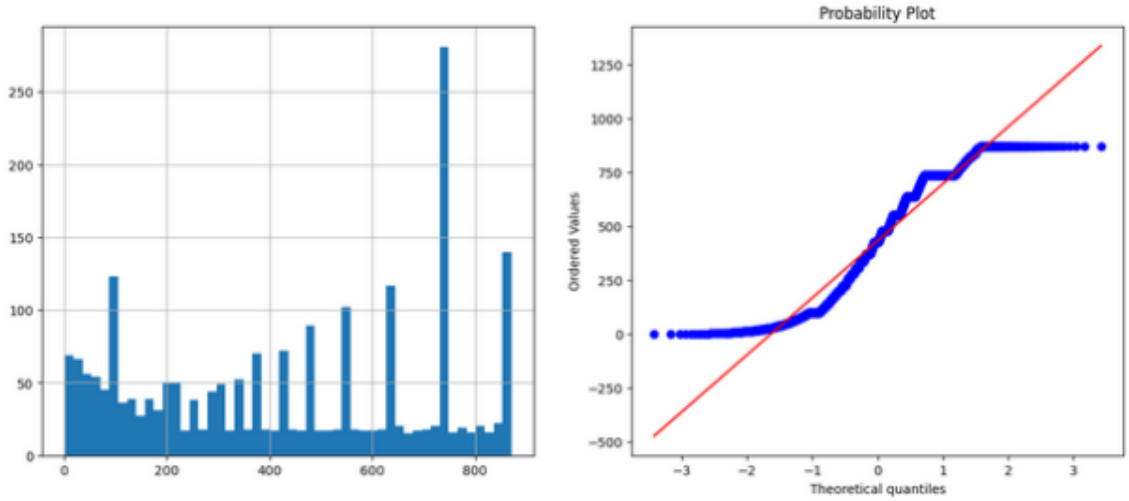


Задача №15.

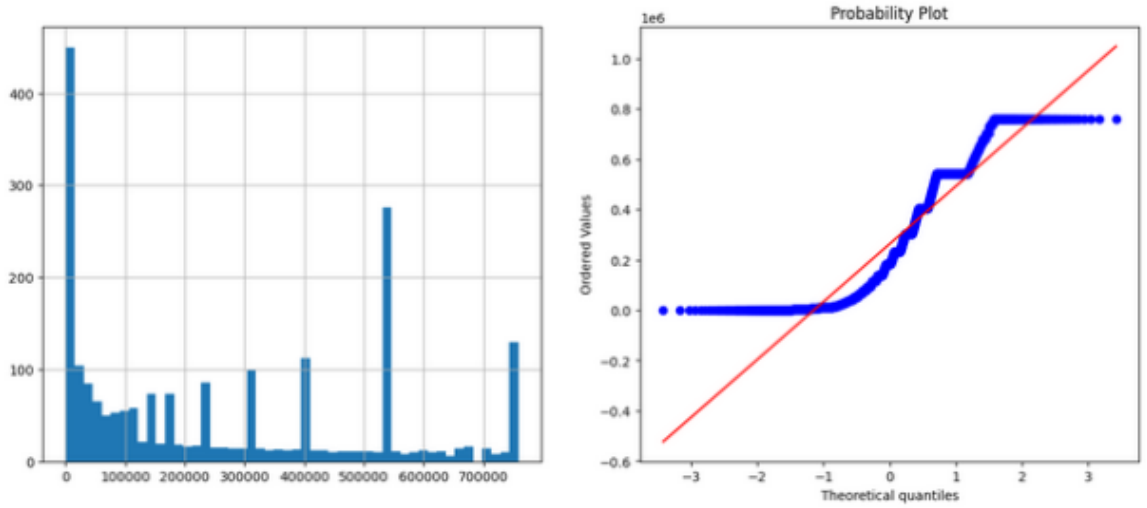
Для набора данных проведите нормализацию для одного (произвольного) числового признака с использованием функции "возведение в степень".

[5]:
```
def diagnostic_plots(df, variable):
    plt.figure(figsize=(15,6))
    # гистограмма
    plt.subplot(1, 2, 1)
    df[variable].hist(bins=50)
    ## Q-Q plot
    plt.subplot(1, 2, 2)
    stats.probplot(df[variable], plot=plt)
    plt.show()
```

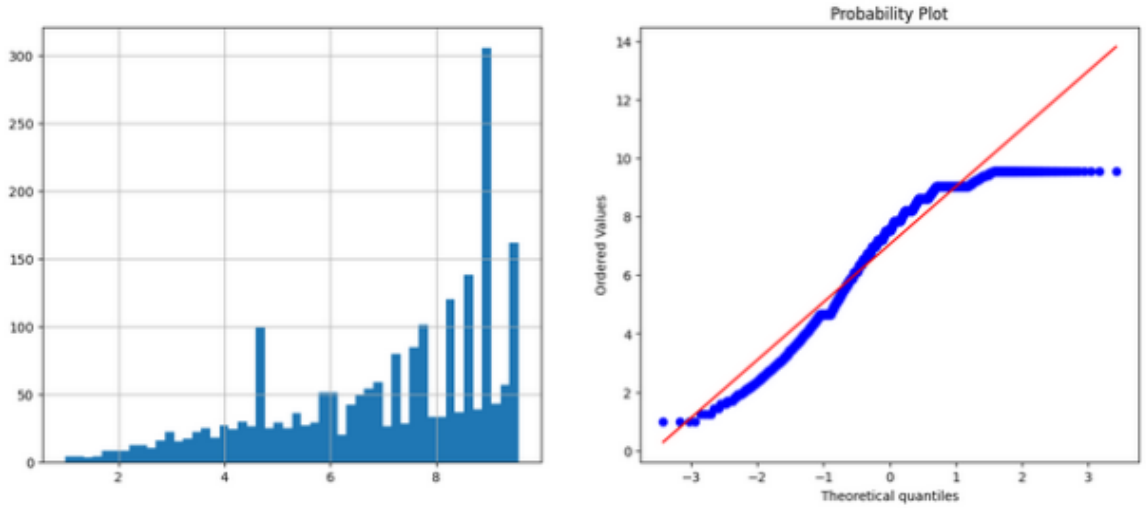
[6]:
```
diagnostic_plots(data, 'patents')
```



[7]:
```
data['patents2'] = data['patents']**(2)
diagnostic_plots(data, 'patents2')
```



[8]:
```
data['patents2'] = data['patents']**(1/3)
diagnostic_plots(data, 'patents2')
```



Таким образом, можно сказать, что данный вид нормализации не очень подходит для конкретного случая, однако, наиболее хорошо он себя показывает при возведении в степень “1/3”.

Задача №35.

Для набора данных проведите процедуру отбора признаков (feature selection). Используйте метод вложений (embedded method). Используйте подход на основе дерева решений.

[11]:
```
half_data = data.copy()
half_data
```

[11]:

	world_rank	institution	country	national_rank	quality_of_education	alumni_emp
0	1	Harvard University	USA	1	7	
1	2	Massachusetts Institute of Technology	USA	2	9	
2	3	Stanford University	USA	3	17	
3	4	University of Cambridge	United Kingdom	1	10	
4	5	California Institute of Technology	USA	4	2	
...	...	...	...	...	...	
2195	996	University of the Algarve	Portugal	7	367	
2196	997	Alexandria University	Egypt	4	236	
2197	998	Federal University of Ceará	Brazil	18	367	
2198	999	University of A Coruña	Spain	40	367	
2199	1000	China Pharmaceutical University	China	83	367	

2200 rows × 14 columns

Удалим текстовые столбцы (от этого может потеряться смысл процедуры отбора признаков, однако считаю необходимым их удалить для упрощения выполнения задания). Также удалим строки с пустыми значениями.

[12]:
```
half_data = data.dropna(axis=0,how='any')
half_data.pop('institution')
half_data.pop('country')
(data.shape, half_data.shape)
```

[12]:
```
((2200, 14), (2000, 12))
```

[13]:
```
half_data.dtypes
```

[13]:
```
world_rank               int64
national_rank            int64
quality_of_education     int64
alumni_employment        int64
quality_of_faculty       int64
publications             int64
influence                int64
citations                int64
broad_impact           float64
patents                  int64
score                  float64
year                     int64
dtype: object
```

все данные представлены в числовом виде, теперь можно производить процедуру отбора признаков. определим целевой признак "world_rank"

[14]:
```
x = half_data.copy()
x = x.drop(columns='world_rank')
x
```

[14]:

	national_rank	quality_of_education	alumni_employment	quality_of_faculty	publicati
200	1	1	1	1	
201	2	11	2	4	
202	3	3	11	2	
203	1	2	10	5	
204	2	7	12	10	
...	...	...	...	...	
2195	7	367	567	218	9
2196	4	236	566	218	9
2197	18	367	549	218	8

[15]:
```
y = half_data['world_rank']
y
```

[15]:
```
200        1
201        2
202        3
203        4
204        5
         ...
2195     996
2196     997
2197     998
2198     999
2199    1000
Name: world_rank, Length: 2000, dtype: int64
```

[15]:
```
dtc1 = DecisionTreeRegressor()
dtc1.fit(x,y)
dtc1.feature_importances_, sum(dtc1.feature_importances_)
```

[15]:
```
(array([9.90871501e-05, 2.69901988e-04, 4.43104786e-03, 1.16170347e-03,
        5.00205881e-04, 7.90119932e-05, 3.22699586e-04, 1.07505571e-01,
        1.73197383e-03, 8.79763045e-01, 3.87762846e-03, 2.58123120e-04]),
 1.0)
```

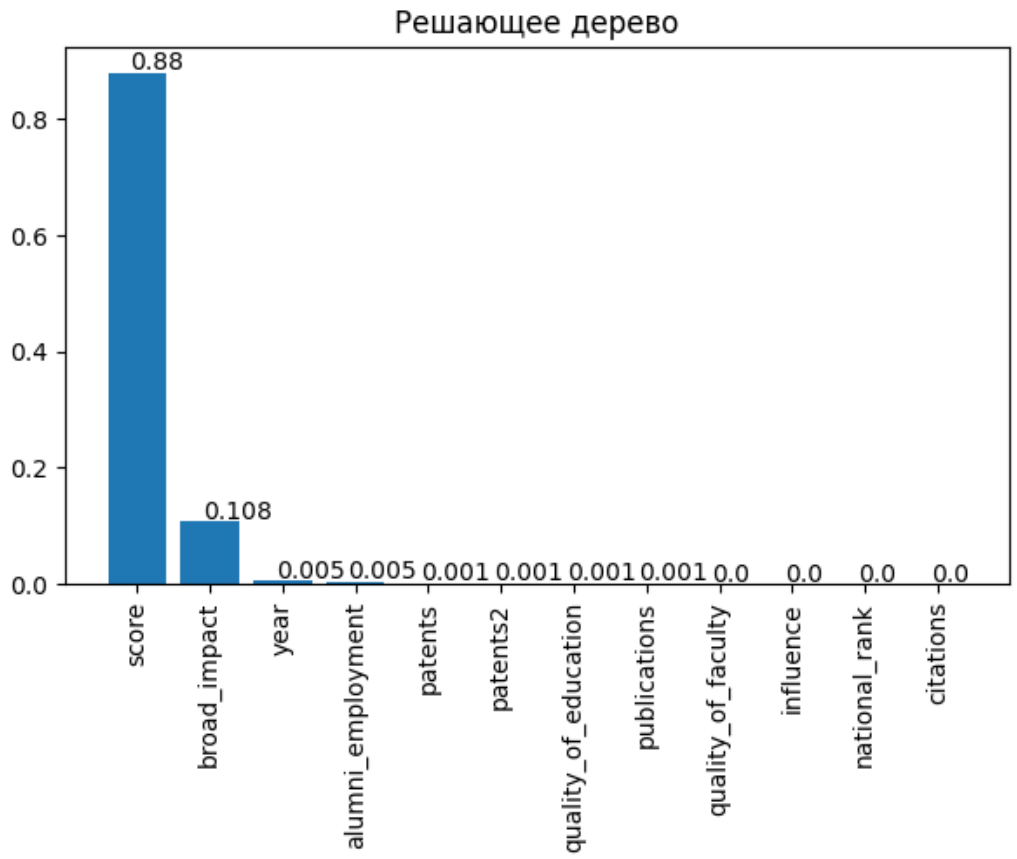
[16]:
```
dtc1 = DecisionTreeRegressor()
dtc1.fit(x,y)
dtc1.feature_importances_, sum(dtc1.feature_importances_)
```

[16]:
```
(array([5.65400814e-05, 6.84475338e-04, 4.55740026e-03, 2.11883370e-04,
        5.31277246e-04, 6.06999677e-05, 4.42892292e-05, 1.07503764e-01,
        1.25357755e-03, 8.79663712e-01, 4.69333057e-03, 7.39050550e-04]),
 0.9999999999999999)
```

[17]:
```python
from operator import itemgetter

def draw_feature_importances(tree_model, X_dataset, title, figsize=(7,4)):
    """
    Вывод важности признаков в виде графика
    """
    # Сортировка значений важности признаков по убыванию
    list_to_sort = list(zip(X_dataset.columns.values, tree_model.feature_importances_))
    sorted_list = sorted(list_to_sort, key=itemgetter(1), reverse = True)
    # Названия признаков
    labels = [x for x,_ in sorted_list]
    # Важности признаков
    data = [x for _,x in sorted_list]
    # Вывод графика
    fig, ax = plt.subplots(figsize=figsize)
    ax.set_title(title)
    ind = np.arange(len(labels))
    plt.bar(ind, data)
    plt.xticks(ind, labels, rotation='vertical')
    # Вывод значений
    for a,b in zip(ind, data):
        plt.text(a-0.1, b+0.005, str(round(b,3)))
    plt.show()
    return labels, data
```

[18]:
```python
_,_=draw_feature_importances(dtc1, x, 'Решающее дерево')
```



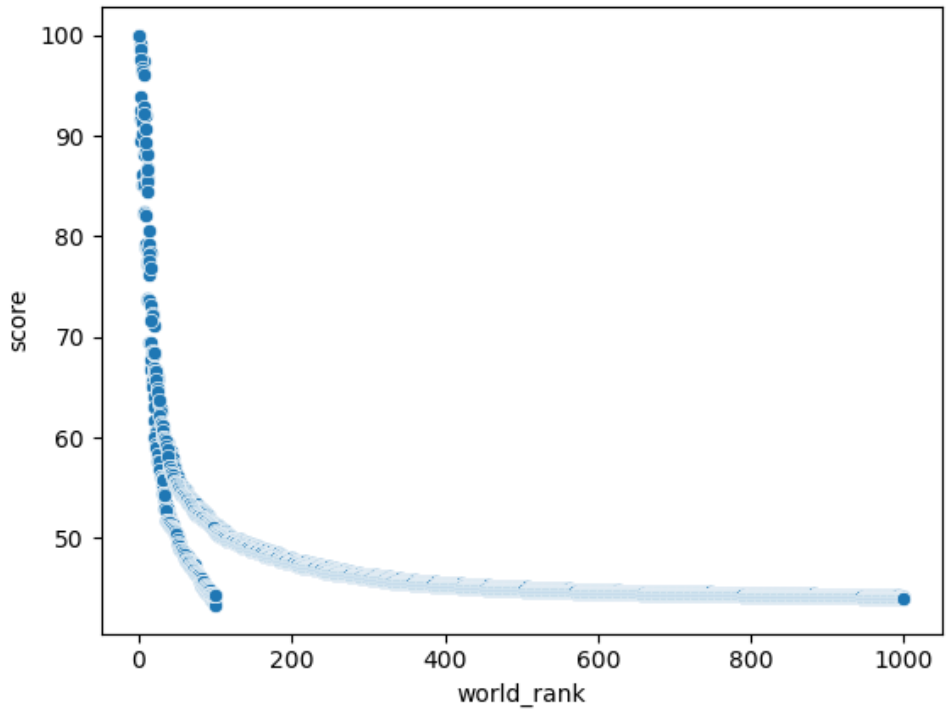
Таким образом, можно сказать, что отбор ведётся по признакам score и broad_impact

Дополнительное задание

Дополнительное задание для группы ИУ5-21М: Для студентов групп ИУ5-21М, ИУ5И-21М, ИУ5Ц-21М - для пары произвольных колонок данных построить график "Диаграмма рассеяния".

[4]:
```
%pip install seaborn
```

[5]:
```
import seaborn as sns
sns.scatterplot(data=data, x="world_rank", y="score")
plt.show()
```



[6]:
```
sns.pairplot(data)
```

[6]:
```
<seaborn.axisgrid.PairGrid at 0x8fc6a78>
```