Ткачева Диана ИУ5-22М РК1

Вариант №15

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as stats
import seaborn as sns
from sklearn.tree import DecisionTreeClassifier
from sklearn.datasets import load_iris
```

Задача 15

```
data = pd.read_csv('online_store_customer_data.csv', sep=",")
```

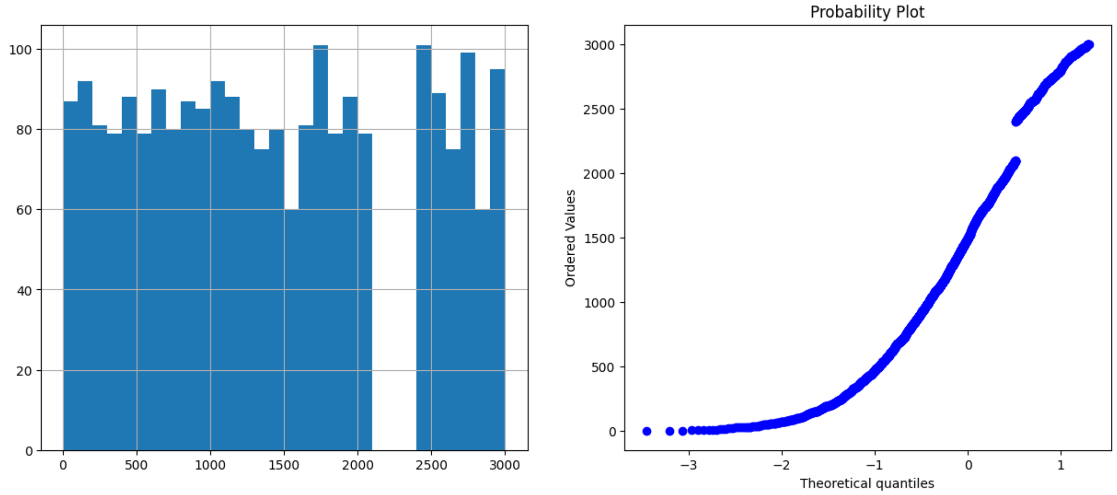
```
data.Amount_spent.unique()
```

```
array([2051.36,  544.04, 1572.6 , ..., 2030.07, 1909.77, 1073.15])
```

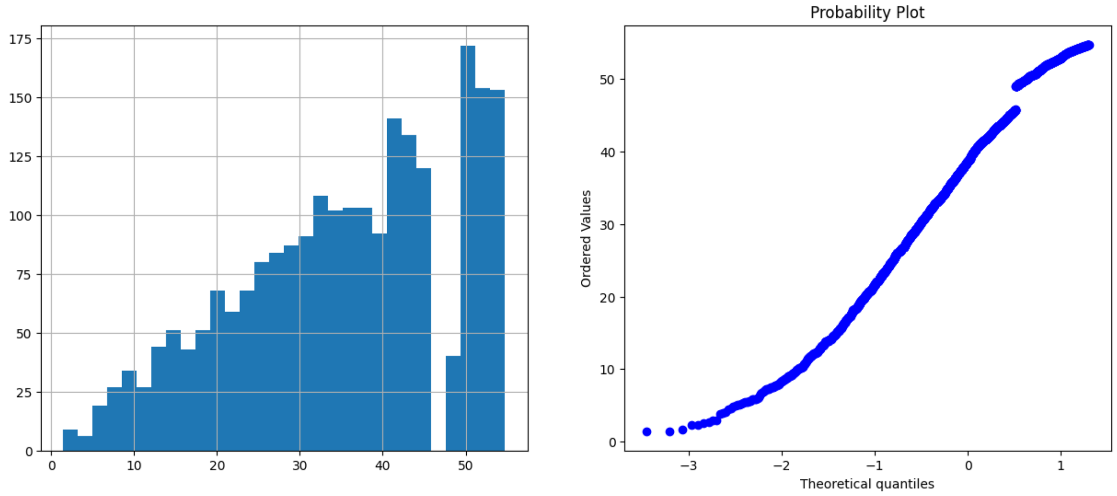
```
data.head()
```

```
  Transaction_date  Transaction_ID  Gender   Age Marital_status  
State_names  \
0         1/1/2019          151200  Female  19.0         Single  
Kansas   
1         1/1/2019          151201    Male  49.0         Single  
Illinois   
2         1/1/2019          151202    Male  63.0        Married   New 
Mexico   
3         1/1/2019          151203     NaN  18.0         Single  
Virginia   
4         1/1/2019          151204    Male  27.0         Single  
Connecticut   

    Segment Employees_status Payment_method  Referal  Amount_spent  
Age_num  
0     Basic     Unemployment          Other      1.0       2051.36  
19.0  
1     Basic    self-employed           Card      0.0        544.04  
49.0  
2     Basic          workers         PayPal      1.0       1572.60  
63.0  
3  Platinum          workers           Card      1.0       1199.79  
18.0  
4     Basic    self-employed           Card      0.0           NaN  
27.0  
```

```
def diagnostic_plots(df, variable):
    plt.figure(figsize=(15,6))
    # 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()

diagnostic_plots(data,"Amount_spent")
```



```
data['Amount_spent_num'] = data['Amount_spent']**(1/2)
diagnostic_plots(data, 'Amount_spent_num')
```



Задача 35

```
iris = load_iris()
dataX = iris.data
dataY = iris.target

d1 = pd.DataFrame(data=iris['data'], columns=iris['feature_names'])
d2 = pd.DataFrame(data=iris['target'], columns=['class']).apply(lambda
x: iris['target_names'][x])
df = pd.concat([d1,d2],axis=1)
```

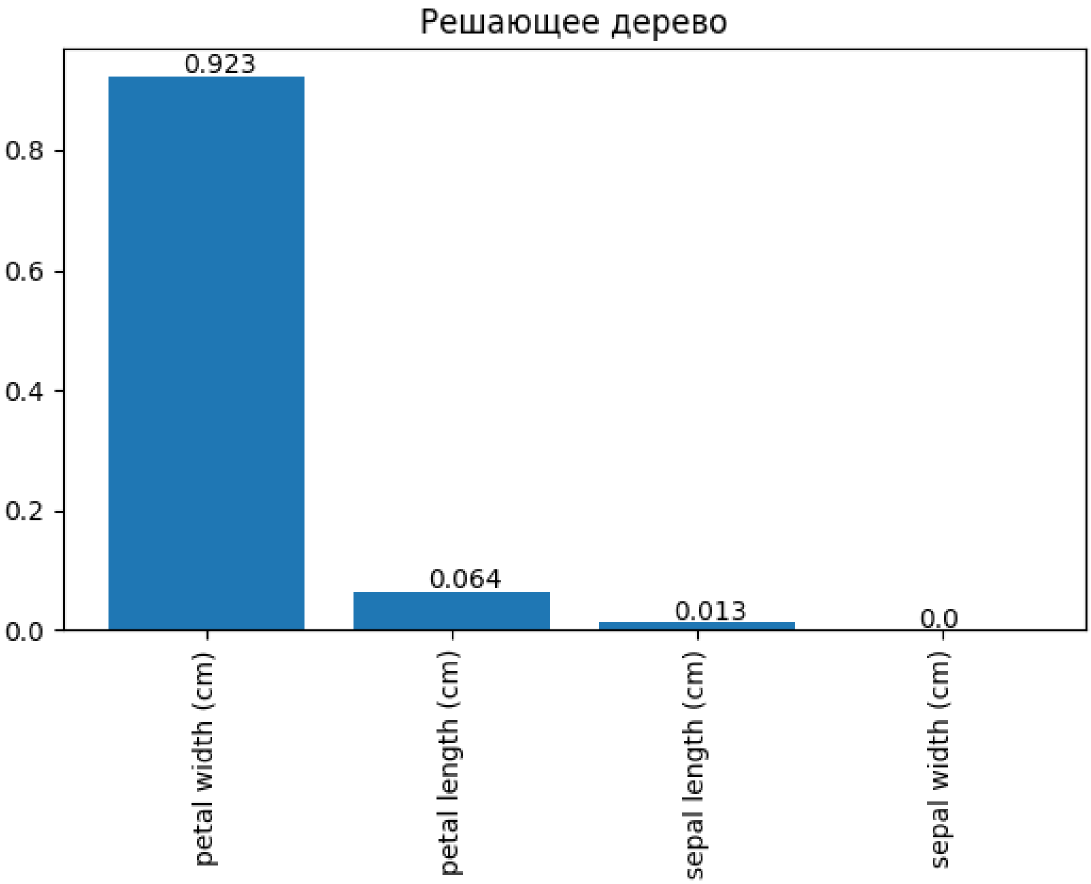
```
df.head()
```

```
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width
(cm)  \
0                5.1               3.5                1.4
0.2
1                4.9               3.0                1.4
0.2
2                4.7               3.2                1.3
0.2
3                4.6               3.1                1.5
0.2
4                5.0               3.6                1.4
0.2

    class
0  setosa
1  setosa
2  setosa
3  setosa
4  setosa
```

```
dtc1 = DecisionTreeClassifier()
dtc1.fit(dataX, dataY)

# Важность признаков
dtc1.feature_importances_, sum(dtc1.feature_importances_)
```

```
(array([0.01333333, 0.        , 0.06405596, 0.92261071]), 1.0)
```

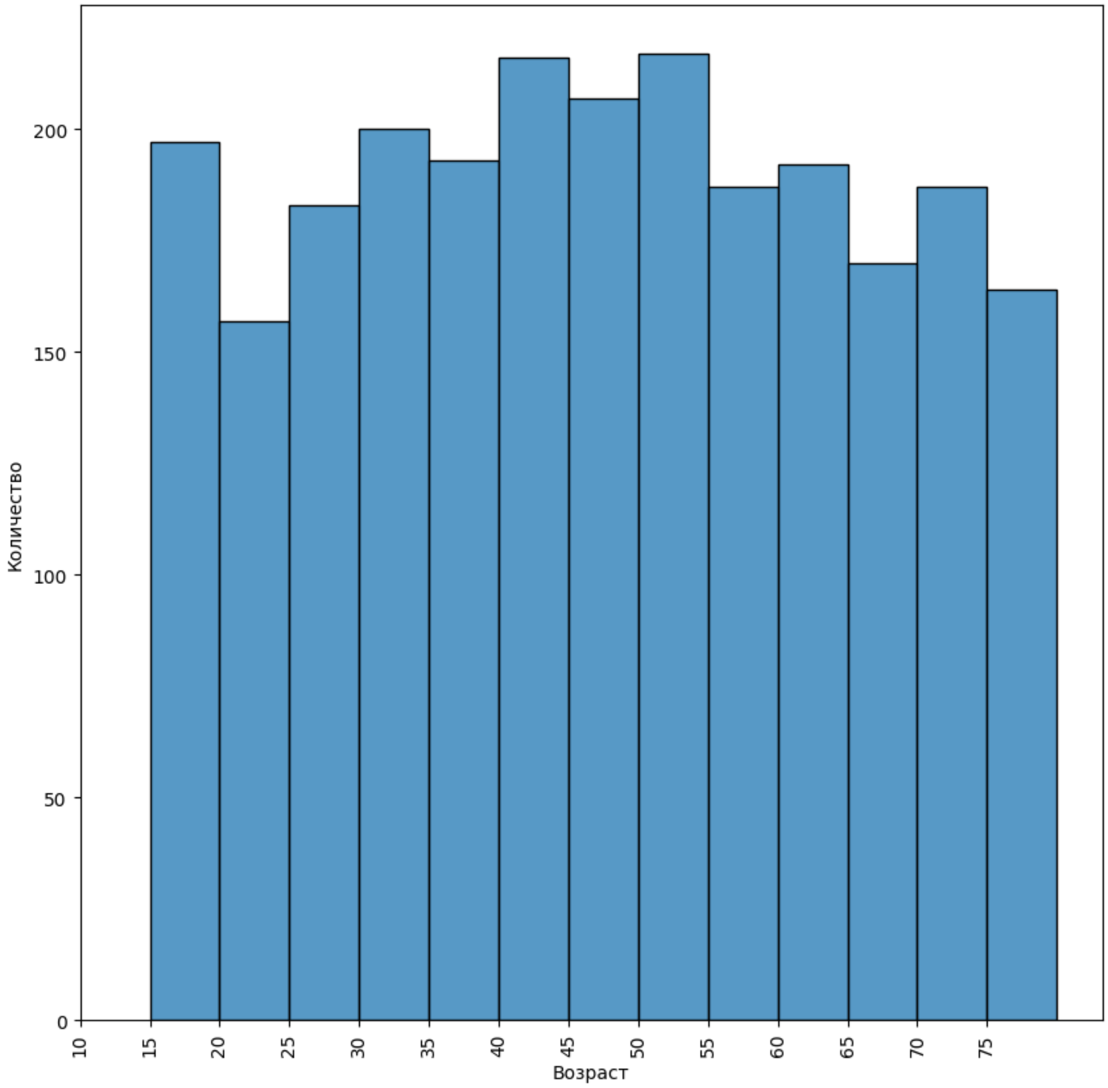
```
from operator import itemgetter

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

_,_=draw_feature_importances(dtc1, d1, 'Решающее дерево')
```



Дополнительное задание

```
data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2512 entries, 0 to 2511
Data columns (total 11 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Transaction_date   2512 non-null   object
 1   Transaction_ID     2512 non-null   int64
 2   Gender             2484 non-null   object
 3   Age                2470 non-null   float64
 4   Marital_status    2512 non-null   object 
 5   State_names       2512 non-null   object 
 6   Segment           2512 non-null   object 
 7   Employees_status  2486 non-null   object 
 8   Payment_method    2512 non-null   object 
 9   Referal           2357 non-null   float64
 10  Amount_spent      2270 non-null   float64
dtypes: float64(3), int64(1), object(7)
memory usage: 216.0+ KB
```

```
fig, ax = plt.subplots(figsize=(10, 10))
sns.histplot(data, x='Age', binwidth=5, ax=ax)
plt.xticks(range(10, 80, 5), rotation=90)
plt.xlabel('Возраст')
plt.ylabel('Количество')
plt.show()
```


Количество
Возраст