РК1

Мартынова Полина Владимировна ИУ5-21М

Вариант **10**

Номер задачи №1: **10**

Номер задачи №2: **30**

Дополнительные требования:

Для пары произвольных колонок данных построить график "Диаграмма рассеяния"

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats as stats
import datetime
import random
from scipy.stats import chi2_contingency
```

```
data = pd.read_csv('/content/heart2.csv')
```

```
data.drop('id', axis=1, inplace=True)
```

```
data.head()
```

	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type
0	Male	NaN	0	1	Yes	Private	Urbar
1	Female	61.0	0	0	Yes	Self-employed	Rura
2	Male	80.0	0	1	Yes	Private	Rura
3	Female	49.0	0	0	Yes	Private	Urbar

Next steps: View recommended plots

Задача 1

Для набора данных проведите устранение пропусков для одного (произвольного) категориального признака с использованием метода заполнения наиболее распространенным значением.

```
data.isnull().sum()
```

```
gender                0
age                  16
hypertension          0
heart_disease         0
ever_married          0
work_type             0
Residence_type        0
avg_glucose_level     0
bmi                 201
smoking_status        0
stroke                0
dtype: int64
```

Закодируем поле age, чтобы были пропуски в категориальном признаке

```
def encode_age_category(age):
    if pd.isnull(age):
        return np.nan
    if age < 12:
        return 'ребенок'
    elif age < 18:
        return 'подросток'
    else:
        return 'взрослый'
```

```
data['age_category'] = data['age'].apply(encode_age_category)
```

```
data.head()
```

	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type
0	Male	NaN	0	1	Yes	Private	Urbar
1	Female	61.0	0	0	Yes	Self-employed	Rura
2	Male	80.0	0	1	Yes	Private	Rura
3	Female	49.0	0	0	Yes	Private	Urbar

Next steps: View recommended plots

```
data.isnull().sum()
```

```
gender                0
age                  16
hypertension          0
heart_disease         0
ever_married          0
work_type             0
Residence_type        0
avg_glucose_level     0
bmi                 201
smoking_status        0
stroke                0
age_category         16
dtype: int64
```

Заполним пропуска в поле age с использованием метода заполнения наиболее распространенным значением

```
most_common_value = data['age_category'].mode()[0]
data['age_category'].fillna(most_common_value, inplace=True)
```

```
data.isnull().sum()
```

```
gender                0
age                  16
hypertension          0
heart_disease         0
ever_married          0
work_type             0
Residence_type        0
avg_glucose_level     0
bmi                 201
smoking_status        0
stroke                0
age_category          0
dtype: int64
```

```
data.head()
```

	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	strol
0	Male	NaN	0	1	Yes	Private	Urban	228.69	36.6	formerly smoked	
1	Female	61.0	0	0	Yes	Self-employed	Rural	202.21	NaN	never smoked	
2	Male	80.0	0	1	Yes	Private	Rural	105.92	32.5	never smoked	
3	Female	49.0	0	0	Yes	Private	Urban	171.23	34.4	smokes	

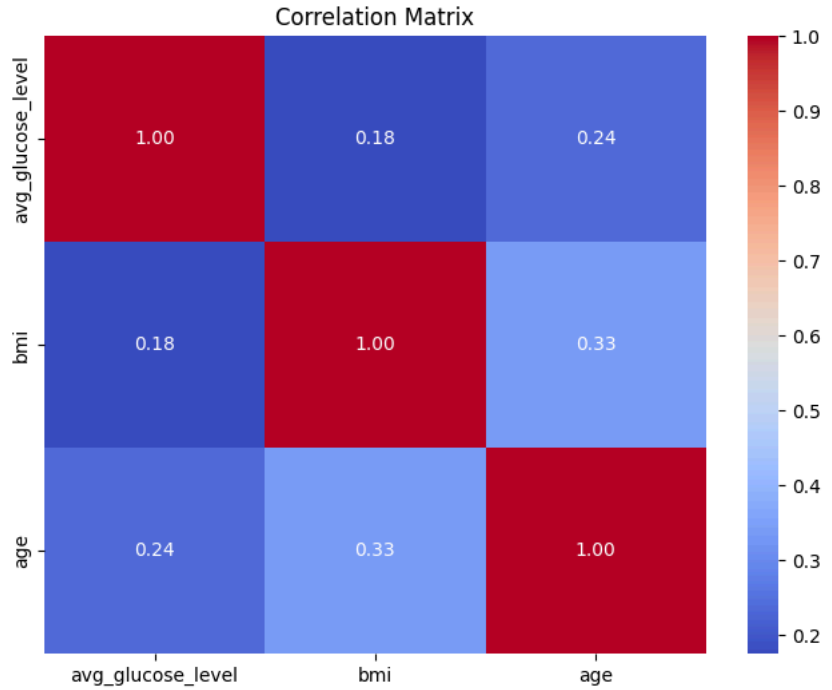
Next steps: View recommended plots

Задача 2

Для набора данных проведите удаление повторяющихся признаков.
Повторяющиеся признаки в машинном обучении означают, что в наборе данных присутствуют признаки, которые имеют одинаковые или очень похожие значения для всех или большинства наблюдений. Такие признаки несут мало информации и могут быть избыточными для модели машинного обучения.

```
correlation_matrix = data[['avg_glucose_level', 'bmi',  'age']].corr()
```

```
plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Matrix')
plt.show()
```



Проверим, есть ли корреляция между категориальными признаками с помощью коэффициента корреляции Крамера

```
def cramers_v(x, y):
    confusion_matrix = pd.crosstab(x, y)
    chi2, _, _, _ = chi2_contingency(confusion_matrix)
    n = confusion_matrix.sum().sum()
    phi2 = chi2 / n
    r, k = confusion_matrix.shape
    phi2corr = max(0, phi2 - ((k-1)*(r-1))/(n-1))
    rcorr = r - ((r-1)**2)/(n-1)
    kcorr = k - ((k-1)**2)/(n-1)
    return np.sqrt(phi2corr / min((kcorr-1), (rcorr-1)))

category_columns = ['gender', 'hypertension',   'heart_disease',    'ever_married', 'work_type',    'Residence_type', 'smoking_status',
for i in range(len(category_columns)-1):
  for j in range(i+1, len(category_columns)):
    print(f'Коэффициент корреляции между признаками {category_columns[i]} и {category_columns[j]}: ')
    print(cramers_v(data[category_columns[i]], data[category_columns[j]]))
```

```
Коэффициент корреляции между признаками gender и hypertension:
0.00894920534403781
Коэффициент корреляции между признаками gender и heart_disease:
0.08344356644424192
Коэффициент корреляции между признаками gender и ever_married:
0.029869761513336545
Коэффициент корреляции между признаками gender и work_type:
0.058593410035366174
Коэффициент корреляции между признаками gender и Residence_type:
0.0
Коэффициент корреляции между признаками gender и smoking_status:
0.07088846101750869
Коэффициент корреляции между признаками gender и stroke:
0.0
Коэффициент корреляции между признаками hypertension и heart_disease:
0.10593696442705003
Коэффициент корреляции между признаками hypertension и ever_married:
0.16296495871402336
Коэффициент корреляции между признаками hypertension и work_type:
0.1602496670438487
Коэффициент корреляции между признаками hypertension и Residence_type:
0.0
Коэффициент корреляции между признаками hypertension и smoking_status:
0.14051440405503587
Коэффициент корреляции между признаками hypertension и stroke:
0.1256069045133085
Коэффициент корреляции между признаками heart_disease и ever_married:
0.11287999419087245
Коэффициент корреляции между признаками heart_disease и work_type:
```

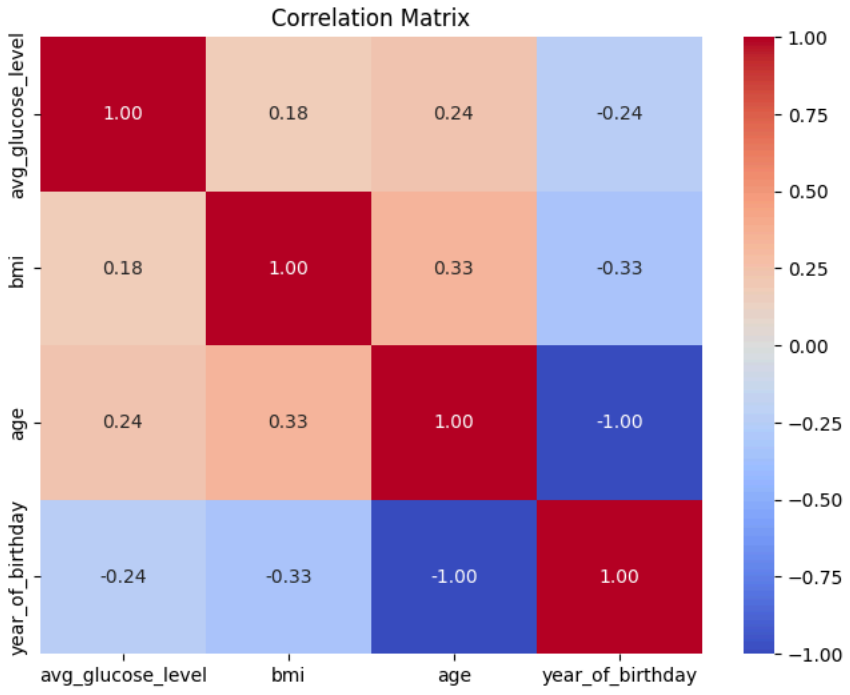
```
0.11426167744807013
Коэффициент корреляции между признаками heart_disease и Residence_type:
0.0
Коэффициент корреляции между признаками heart_disease и smoking_status:
0.09029643617077479
Коэффициент корреляции между признаками heart_disease и stroke:
0.1321779325804356
Коэффициент корреляции между признаками ever_married и work_type:
0.5665892792932351
Коэффициент корреляции между признаками ever_married и Residence_type:
0.0
Коэффициент корреляции между признаками ever_married и smoking_status:
0.34156385364332964
Коэффициент корреляции между признаками ever_married и stroke:
0.10647807334277677
Коэффициент корреляции между признаками work_type и Residence_type:
0.011301366820332882
Коэффициент корреляции между признаками work_type и smoking_status:
0.29980569140643637
Коэффициент корреляции между признаками work_type и stroke:
0.09402041104598291
Коэффициент корреляции между признаками Residence_type и smoking_status:
0.021675770567566924
Коэффициент корреляции между признаками Residence_type и stroke:
0.003992580152454577
Коэффициент корреляции между признаками smoking_status и stroke:
0.07153858559970024
```

Никакие признаки не коррелируют между собой, добавим признак "Год рождения"

```
data['year_of_birthday'] = datetime.datetime.now().year - data['age']  + random.randint(-1, 1)
```

```
correlation_matrix = data[['avg_glucose_level', 'bmi',  'age', 'year_of_birthday']].corr()

plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Matrix')
plt.show()
```



Когда коэффициент корреляции близок к -1, это означает, что между признаками существует полная отрицательная линейная зависимость. Уберем признак "Год рождения".

```
data.drop('year_of_birthday', axis=1, inplace=True)
```

Дополнительное задание

```
import matplotlib.pyplot as plt

column1 = 'avg_glucose_level'
column2 = 'bmi'

plt.figure(figsize=(8, 6))
plt.scatter(data[column1], data[column2], color='b', alpha=0.5)
plt.title('Диаграмма рассеяния между {} и {}'.format(column1, column2))
plt.xlabel(column1)
plt.ylabel(column2)
plt.grid(True)
plt.show()
```