In [20]:
```
import pandas as pd
import numpy as np
from sklearn.preprocessing import LabelEncoder
```

Выберем датасет по инсультам с Kaggle

In [21]:
```
df = pd.read_csv('healthcare-dataset-stroke-data.csv')
df.head(5)
```

Out[21]:

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	a
0	9046	Male	67.0	0	1	Yes	Private	Urban	
1	51676	Female	61.0	0	0	Yes	Self-employed	Rural	
2	31112	Male	80.0	0	1	Yes	Private	Rural	
3	60182	Female	49.0	0	0	Yes	Private	Urban	
4	1665	Female	79.0	1	0	Yes	Self-employed	Rural	

In [22]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5110 entries, 0 to 5109
Data columns (total 12 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   id                 5110 non-null   int64  
 1   gender             5110 non-null   object 
 2   age                5110 non-null   float64
 3   hypertension       5110 non-null   int64  
 4   heart_disease      5110 non-null   int64  
 5   ever_married       5110 non-null   object 
 6   work_type          5110 non-null   object 
 7   Residence_type     5110 non-null   object 
 8   avg_glucose_level  5110 non-null   float64
 9   bmi                4909 non-null   float64
 10  smoking_status     5110 non-null   object 
 11  stroke             5110 non-null   int64  
dtypes: float64(3), int64(4), object(5)
memory usage: 479.2+ KB
```

Проведем обнаружение и удаление выбросов на основе 5% и 95% квантилей для признака avg_glucose_level

In [31]:
```
min = np.percentile(df.avg_glucose_level, 5)
max = np.percentile(df.avg_glucose_level, 95)
```

In [32]:
```
df[(df.avg_glucose_level > min) & (df.avg_glucose_level < max)]
```

Out[32]:

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type
1	51676	Female	61.0	0	0	Yes	Self-employed	Rura
2	31112	Male	80.0	0	1	Yes	Private	Rura
3	60182	Female	49.0	0	0	Yes	Private	Urban
4	1665	Female	79.0	1	0	Yes	Self-employed	Rura
5	56669	Male	81.0	0	0	Yes	Private	Urban
...	...	...	...	...	...	...	...	...
5105	18234	Female	80.0	1	0	Yes	Private	Urban
5106	44873	Female	81.0	0	0	Yes	Self-employed	Urban
5107	19723	Female	35.0	0	0	Yes	Self-employed	Rura
5108	37544	Male	51.0	0	0	Yes	Private	Rura
5109	44679	Female	44.0	0	0	Yes	Govt_job	Urban

4598 rows × 12 columns

Сделаем LabelEncoding для признака gender

In [35]:
```
le = LabelEncoder()
df['gender'] = le.fit_transform(df.gender)
df.gender
```

Out[35]:
```
0       1
1       0
2       1
3       0
4       0
       ..
5105    0
5106    0
5107    0
5108    1
5109    0
Name: gender, Length: 5110, dtype: int64
```