МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
им. Н.Э. Баумана

Кафедра «Систем обработки информации и управления»

Лабораторная работа 1

по дисциплине

«Методы машинного обучения»

по теме

«Разведочный анализ данных. Исследование и визуализация данных»

ИСПОЛНИТЕЛЬ: Чечнев А.А.
ФИО

группа ИУ5-23М

_____
подпись

"__"_____2020 г.

ПРЕПОДАВАТЕЛЬ: Гапанюк Ю. Е.
ФИО

_____
подпись

"__"_____2020 г.

Москва – 2020

Разведочный анализ данных. Исследование и визуализация данных.

Цель лабораторной работы: изучение различных методов визуализация данных.

Краткое описание.

Построение основных графиков, входящих в этап разведочного анализа данных. Корреляционный анализ данных. Формирование выводов о возможности построения моделей машинного обучения и о возможном вкладе признаков в модель.

Рекомендуемые инструментальные средства можно посмотреть здесь.

Требования к отчету:

Отчет по лабораторной работе должен содержать:

- титульный лист;
- описание задания;
- текст программы;
- экранные формы с примерами выполнения программы.

В случае использования ноутбуков фрагменты 3 и 4 соответствуют ячейкам ноутбуков. Отчеты размещаются в репозитории курса, который каждый студент создает в своем профиле на github.

Задание:

1. Выбрать набор данных (датасет). Вы можете найти список свободно распространяемых датасетов здесь. Для лабораторных работ не рекомендуется выбирать датасеты большого размера.
2. Создать ноутбук, который содержит следующие разделы:
3. Текстовое описание выбранного Вами набора данных.
4. Основные характеристики датасета.
5. Визуальное исследование датасета. Необходимо использовать не менее 2 различных библиотек и не менее 5 графиков.
6. Информация о корреляции признаков.
7. Сформировать отчет и разместить его в своем репозитории на github.

Средства и способы визуализации данных можно посмотреть здесь.

В качестве опорного примера для выполнения лабораторной работы можно использовать пример.

Дополнительно примеры решения задач, содержащие визуализацию, можно посмотреть в репозитории курса mlcourse.ai - https://github.com/Yorko/mlcourse.ai/wiki/Individual-projects-and-tutorials-(in-Russian) (https://github.com/Yorko/mlcourse.ai/wiki/Individual-projects-and-tutorials-(in-Russian))

Analyzing cardiovascular disease data

dataset:

Feature	Variable Type	Variable	Value Type
Age	Objective Feature	age	int (days)
Height	Objective Feature	height	int (cm)
Weight	Objective Feature	weight	float (kg)
Gender	Objective Feature	gender	categorical code
Systolic blood pressure	Examination Feature	ap_hi	int
Diastolic blood pressure	Examination Feature	ap_lo	int
Cholesterol	Examination Feature	cholesterol	1: normal, 2: above normal, 3: well above normal
Glucose	Examination Feature	gluc	1: normal, 2: above normal, 3: well above normal
Smoking	Subjective Feature	smoke	binary
Alcohol intake	Subjective Feature	alco	binary
Physical activity	Subjective Feature	active	binary
Presence or absence of cardiovascular disease	Target Variable	cardio	binary

In [2]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
sns.set(rc={'figure.figsize':(16,8)})
```

In [6]:

```
df = pd.read_csv('data/mlbootcamp5_train.csv')
df.head(7)
```

Out[6]:

	id	age	gender	height	weight	ap_hi	ap_lo	cholesterol	gluc	smoke	alco	active	ca
0	0	18393	2	168	62.0	110	80	1	1	0	0	1	
1	1	20228	1	156	85.0	140	90	3	1	0	0	1	
2	2	18857	1	165	64.0	130	70	3	1	0	0	0	
3	3	17623	2	169	82.0	150	100	1	1	0	0	1	
4	4	17474	1	156	56.0	100	60	1	1	0	0	0	
5	8	21914	1	151	67.0	120	80	2	2	0	0	0	
6	9	22113	1	157	93.0	130	80	3	1	0	0	1	

In [11]:

```
df.describe()
```

Out[11]:

	id	age	gender	height	weight	ap_hi	
count	70000.000000	70000.000000	70000.000000	70000.000000	70000.000000	70000.000000	70
mean	49972.419900	19468.865814	1.349571	164.359229	74.205690	128.817286	
std	28851.302323	2467.251667	0.476838	8.210126	14.395757	154.011419	
min	0.000000	10798.000000	1.000000	55.000000	10.000000	-150.000000	
25%	25006.750000	17664.000000	1.000000	159.000000	65.000000	120.000000	
50%	50001.500000	19703.000000	1.000000	165.000000	72.000000	120.000000	
75%	74889.250000	21327.000000	2.000000	170.000000	82.000000	140.000000	
max	99999.000000	23713.000000	2.000000	250.000000	200.000000	16020.000000	11

In [20]:

```
print('Количество пустых строк:')
df.isna().sum()
```

```
Количкство пустых строк:
```

Out[20]:

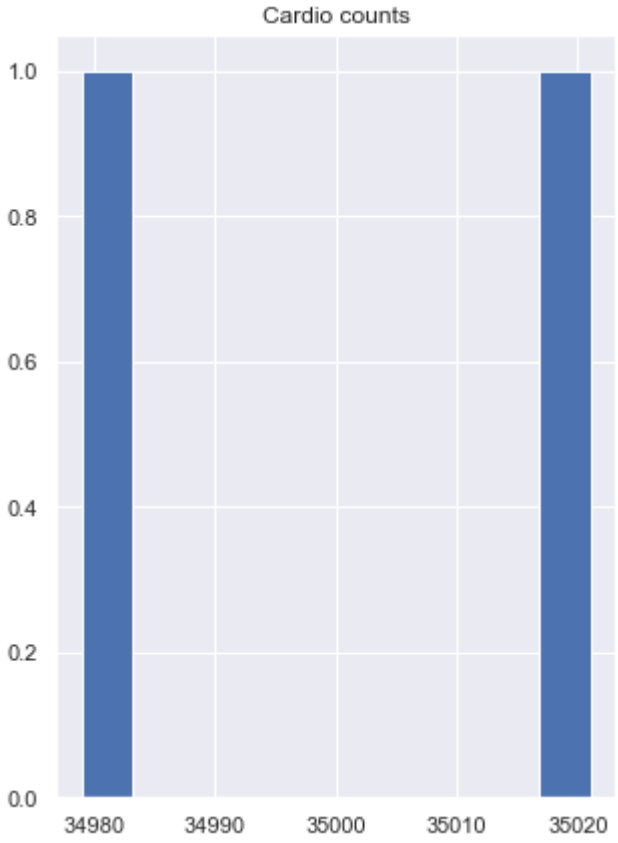
```
id             0
age            0
gender         0
height         0
weight         0
ap_hi          0
ap_lo          0
cholesterol    0
gluc           0
smoke          0
alco           0
active         0
cardio         0
dtype: int64
```

In [237]:

```
plt.title('Cardio counts')
df.cardio.value_counts().hist(figsize = (5,7))
df.cardio.value_counts()
```

Out[237]:

```
0    35021
1    34979
Name: cardio, dtype: int64
```



Классы целевой переменной сбалансированны

Построим гистограмму категориальных признаков

In [240]:

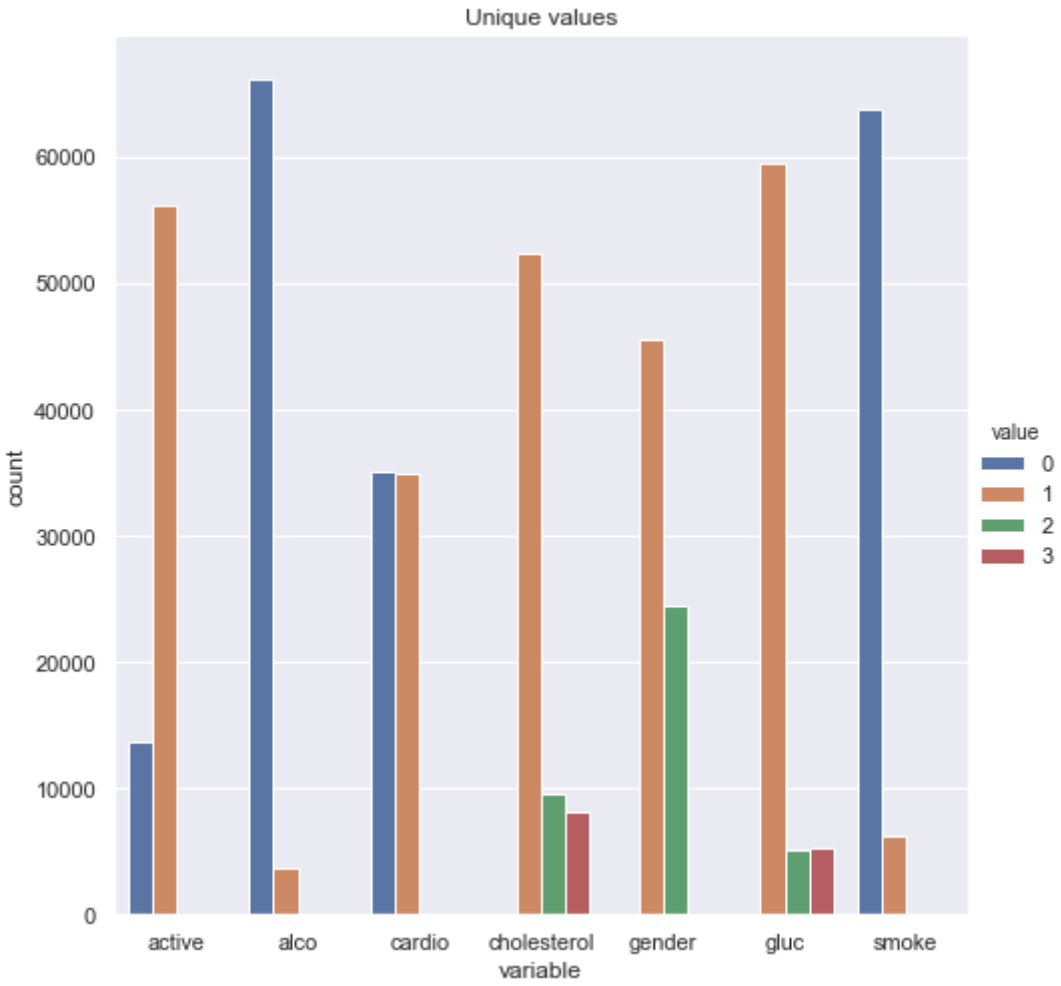
```
categ_columns = ['gender','cholesterol',
                 'gluc', 'smoke', 'alco',
                 'active', 'cardio']

df_uniques = df.melt(value_vars=categ_columns)


df_uniques = df_uniques.groupby(by=['variable', 'value'])\
                       .agg({'value':'count'})\
                       .rename(columns={'value':'count'})\
                       .sort_index(level=[0, 1])\
                       .reset_index()
```

In [241]:

```
sns.catplot(x='variable', y='count', hue='value',
               data=df_uniques, kind='bar', height=7);
plt.title('Unique values');
```



In [106]:

```
df_uniques.head()
```

Out[106]:

	variable	value	count
0	active	0	13739
1	active	1	56261
2	alco	0	66236
3	alco	1	3764
4	cardio	0	35021

In [111]:

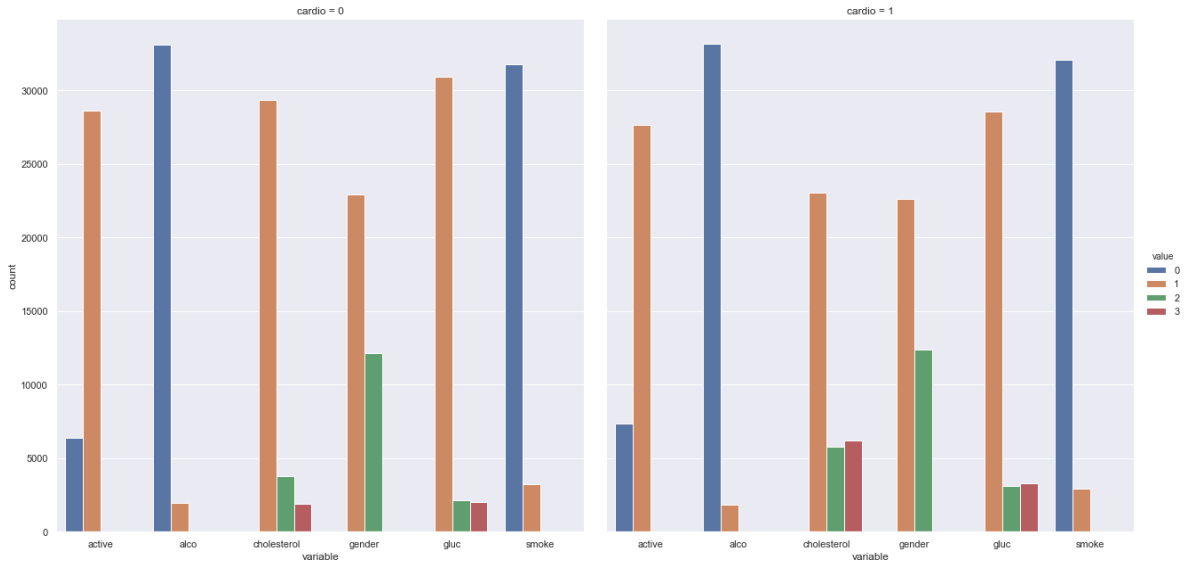
```
categ_columns.remove('cardio')
```

In [118]:

```
df_unique_cardio = df.melt(value_vars=categ_columns, id_vars='cardio')\
            .groupby(by=['variable', 'value', 'cardio'])\
            .agg({'value':'count'})\
            .rename(columns={'value':'count'})\
            .reset_index()
```

In [121]:

```
sns.catplot(x='variable', y='count', hue='value',
              col='cardio', data=df_unique_cardio, kind='bar', height=9);
```



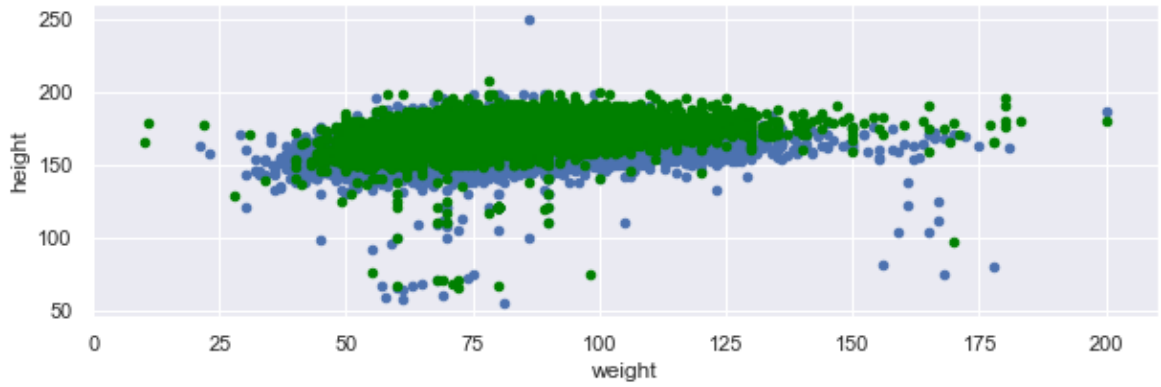
Unusefull graph (remove this later)

In [187]:

```
#ax = plt.plot()

#for val in df.gender.unique():
ax = df[df.gender == 1].loc[:, ['height', 'weight']].plot.scatter(x='weight', y='height')
df[df.gender == 2].loc[:, ['height', 'weight']].plot.scatter(x='weight', y='height', ax=ax,
```

```
'c' argument looks like a single numeric RGB or RGBA sequence, which should
be avoided as value-mapping will have precedence in case its length matches
with 'x' & 'y'.  Please use a 2-D array with a single row if you really want
to specify the same RGB or RGBA value for all points.
```



In [233]:

```
#df.plot.scatter(x='weight', y='height')
```

Расчитаем возраст пациентов и рассмотрим влияние возраста на появление заболевания

In [151]:

```
df['age_y'] = round(df.age/265.25)
df.head()
```

Out[151]:

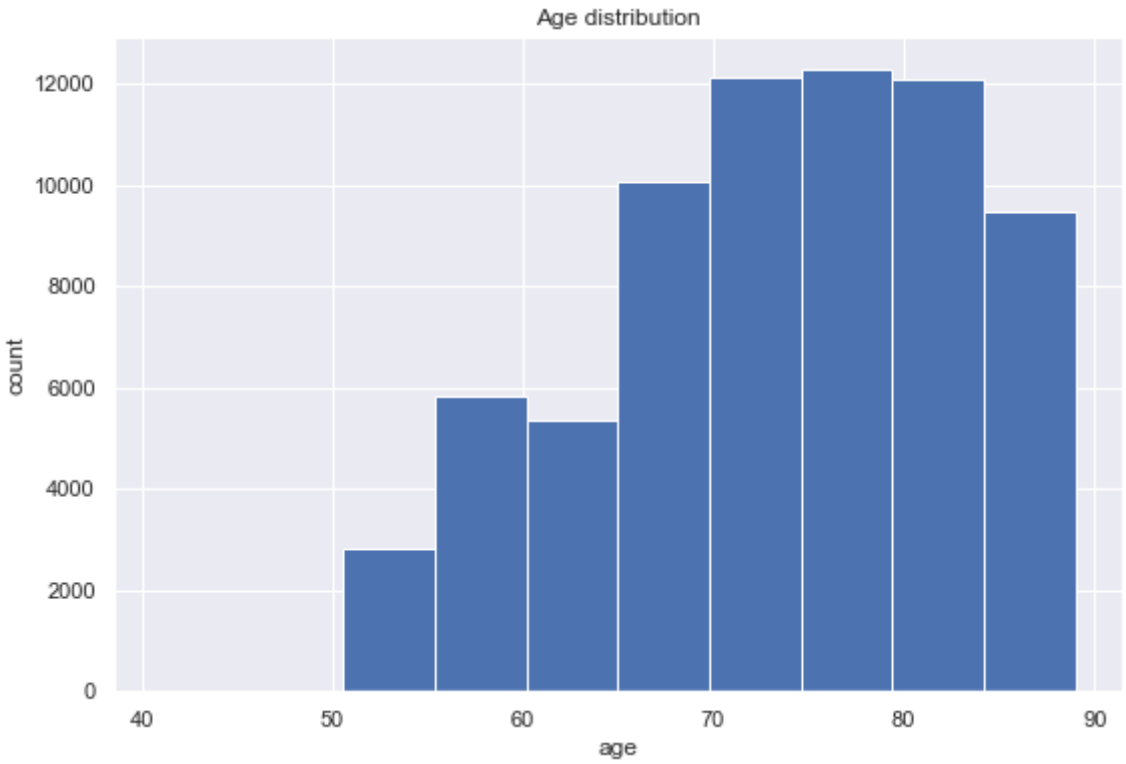
	id	age	gender	height	weight	ap_hi	ap_lo	cholesterol	gluc	smoke	alco	active	ca
0	0	18393	2	168	62.0	110	80	1	1	0	0	1	
1	1	20228	1	156	85.0	140	90	3	1	0	0	1	
2	2	18857	1	165	64.0	130	70	3	1	0	0	0	
3	3	17623	2	169	82.0	150	100	1	1	0	0	1	
4	4	17474	1	156	56.0	100	60	1	1	0	0	0	

In [209]:

```
df.age_y.hist(figsize = (9,6))
plt.xlabel('age')
plt.ylabel('count')
plt.title('Age distribution')
```

Out[209]:

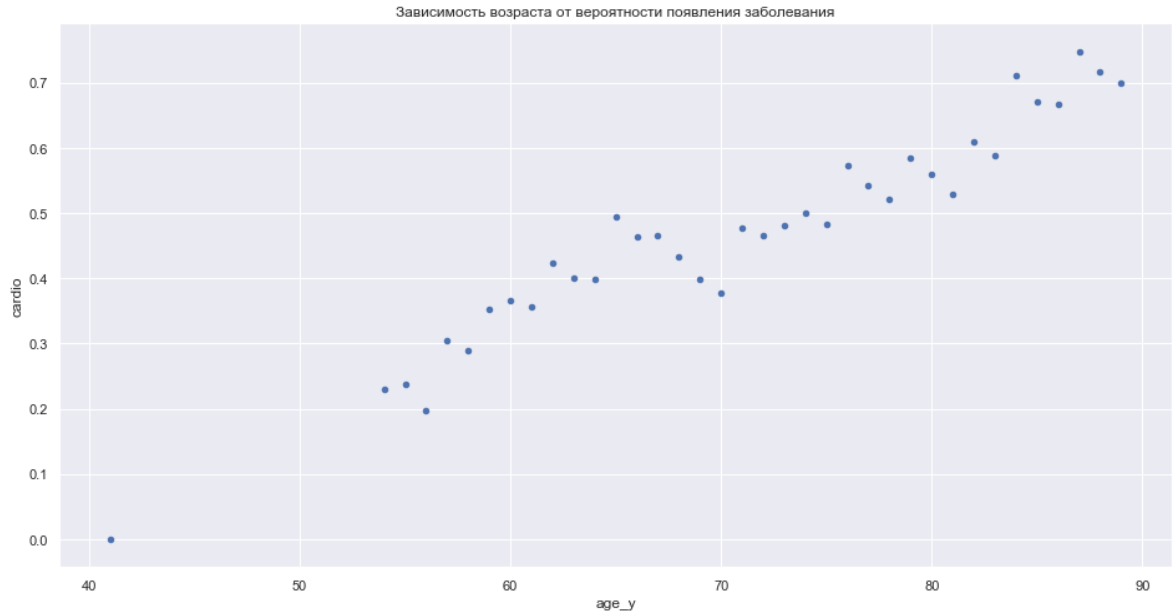
```
Text(0.5, 1.0, 'Age distribution')
```



In [245]:

```
#df.hist(figsize = (9,6), by='gender')
df.groupby(by='age_y', as_index=False).agg({'cardio':'mean'})\
        .plot.scatter(x='age_y', y='cardio');
plt.title('Зависимость возраста от вероятности появления заболевания');
```

```
'c' argument looks like a single numeric RGB or RGBA sequence, which should
be avoided as value-mapping will have precedence in case its length matches
with 'x' & 'y'.  Please use a 2-D array with a single row if you really want
to specify the same RGB or RGBA value for all points.
```



Как видно с возрастом повышается вероятность кардиоваскулярных заболеваний

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

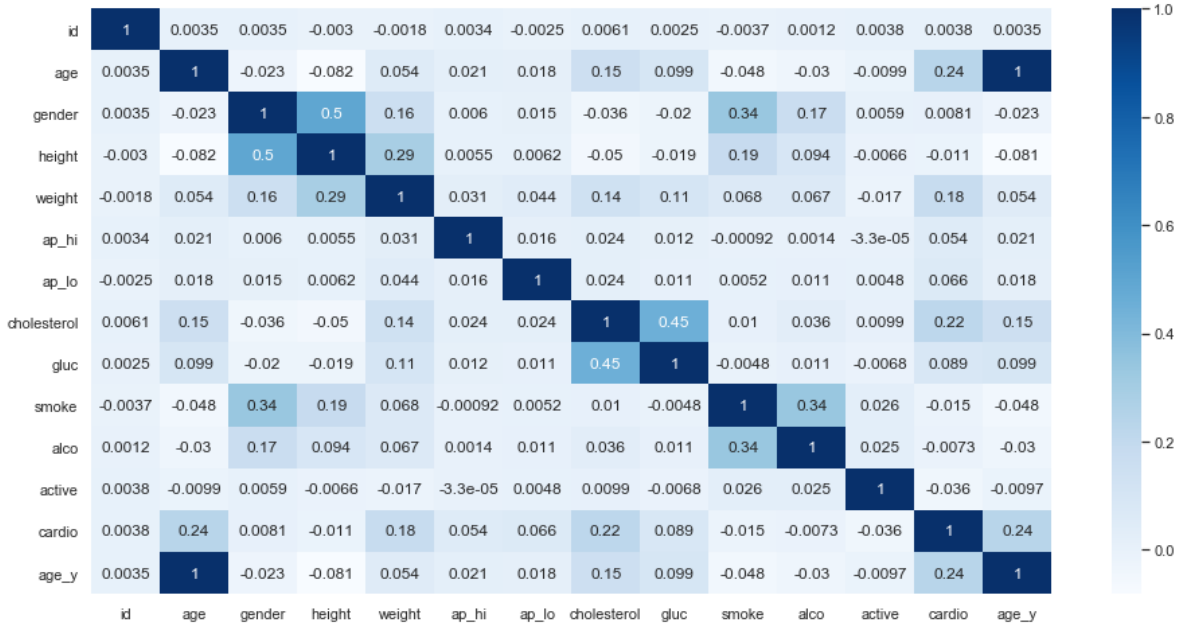
```

```

Рассмотрим корреляцию признаков при помощи температурной карты

In [193]:

```
sns.set(rc={'figure.figsize':(16,8)})
ax = sns.heatmap(df.corr(), annot=True, cmap="Blues")
```



Целевая переменная кардтозаболевания в наибольшей степени (однако довольно слабо) коррелирует с возрастом, весом, и холестирином и практически не связана с полом и алкоголем