12/9/2020

Coronary Heart Disease (CHD) Study

Strategic Thinking Module

Egor Bogachev sba20172 - AI part time HDip Level 8 course
Alexandra Vaz Ferreira 2019160 – Data Analytics HDip Level 8 course
Míriam Fernández Romero sba20173 - AI part time HDip Level 8 course
,
Semester 1, AIP3 Group

CCT COLLEGUE DUBLIN

ABSTRACT

According to the World Health Organization (2020) 12 million deaths occur worldwide every year due to Heart diseases. Cardiovascular disease is the first death cause in the world. There are factors that can increase the risk of suffering Cardiovascular disease or reduce its complications. In this project we are going to analyse those factors and make a model which will predict if the user will have high or low possibilities of developing CHD according to their habits, demographic, behavioural and medical features.

Index

1. Introduction

As it was mentioned before, Cardiovascular disease is the first death cause in the world. There are factors that will increase the risk of developing the disease. In this project we are going to analyse those factors and make a model which will predict if the user will have high or low possibilities of developing CHD. In this project we are going to show the whole process of working with selected data on selected problem using CRISP-DM framework. Project will include next steps: planning and scheduling the project, business understanding of selected problem, data selection, data description, understanding and preparation of the data, building model, evaluating the model and finally deploying model. The study intent to be completed on risk factors contributing to development of Coronary Heard Disease CHD and identifying highest contributing risk factors for particular individuals.

2. Group Planning

In this section it is going to be reflected the planning done to reach the project goals, showing the project schedule. In this case we used MS Project application to build “Gantt chat” of group activities on this project in first Semester. This is a very important part of out project as we have to plan, organize and coordinate our work, creating in this case a road map, to guide us and work in a more efficient way the different parts of the project. (see Appendix 1)

3. Group Reflexion report

As throughout first semester we have continued to work on “Group Reflexion Report”. This section is have a weekly reflexions o group activities, where it is shown or goals and achievements on weekly basis as defined in project plan. As well as information regarding to challenges we find along while working on project. (See Appendix 2)

4. Problem Milestone Output Report

4.1. Businesses Understanding Report

As CHD is the first cause of world death, we decided to take it as our Project Topic, as there is a clear social need of guiding people in our society to reduce the risk of

developing the illness. There is different external factor, as well as habits than will increase or reduce this possibility.

The first part of the study is check what are the most high-risk CHD factors that will increase the possibility of suffering CHD. We aren't only interested of analysing the factors, also to develop a model that will predict which of those factors have impact on particular individuals.

4.2. Dataset Selection and Description

The text below will provide summary of data selected in first semester of part time AI course, Strategic Thinking module, of dataset chosen to work on selected problem. The problem selected for the project relate to humans coronary heart disease "CHD".

The database selected (Ongoing Heart Study) in first semester comes from kagle.com site and contains data from cardiovascular study on residents of the town of Framingham, Massachusetts.
In this semester we checked several datasets and decided to choose this one as it was big enough and contain relevant information to analyse.
The short reference to considered in first semester datasets shown below. The selected dataset is Dataset3 from below listed datasets.

First Dataset1

https://www.kaggle.com/mazharkarimi/heart-disease-and-stroke-prevention
This is one of the dataset provided by the National Cardiovascular Disease Surveillance System.

Second Dataset2

https://www.kaggle.com/andrewmvd/heart-failure-clinical-data
BMC Medical Informatics and Decision Making 20, 16 (2020)

Third Dataset3

https://www.kaggle.com/amanajmera1/framingham-heart-study-dataset/data[
The dataset is publically available on the Kaggle website, and it is from an ongoing cardiovascular study on residents of the town of Framingham, Massachusetts. The dataset provides the patient's information. It includes over 4,000 records and 15 attributes.

4.3. Data Understanding Report

Dataset consists of more than 4,000 records and of 15 features.

There are demographic, behavioural, and medical features in dataset. Each feature regarded as potential risk factor.

Target variable to predict: 10-year risk of coronary heart disease “CHD” (binary: “1”, means “Yes”, “0” means “No”)

Demographic dataset features

In this section are the demographic features where is going to be analysed the gender, age, and type of education, about groups of people.

Gender: Male or female.

(Dtype: int64, male: 1, female: 0)

Age: age of the patient.

(Dtype: int64, range: 32 to 70 years)

Education: Completed studies: 1 = Some High School; 2 = High School or GED; 3 = Some College or Vocational School; 4 = college.

Education: (Dtype: float64)

Behavioural dataset features

Behavioural features related to their habits.

Current Smoker: Is patient a current smoker

(Dtype: int64, smoker: 1, non-smoker: 0)

Cigs Per Day: Number of cigarettes person smoked on average in one day

(Dtype: int64)

Medical dataset features

Medical features which analyse the possibility to develop CHD depending on healthy personal conditions.

Diabetes: It occurs when blood sugar in blood (glucose) is too high. Early detection gives the ability to protect against heart attack (HSE, 2020).

(Dtype: int64, binary: “1”, means “Yes”, “0” means “No”)

Glucose: People without diabetes typically have fasting blood sugar readings below 100 milligrams per decilitre.

(Dtype: float64)

Total Cholesterol: According to the U.S National library of Medicine (2020) the total cholesterol level is the overall amount of cholesterol in the human blood. It consists of low-density lipoproteins (LDL) and high-density lipoproteins (HDL). Total cholesterol normal range is 125 to 200mg/dl.

(Dtype: float64)

Other, of the important factors that seems to determinate the risk of developing CHD is the hypertension.

Prevalent Hypertension: *Is traditional defines as blood pressure and is* classified in different stages.

> The American Heart Association (2020) summarised the classification in stages and contains the information related to systolic and diastolic mmHg: stage 1 or normal blood pressure (< 120/<80); stage 2 or elevated (120-129/<80); stage 3 or high (130-139/80-89); stage 4 or higher level than stage 3 (140>/90>) and finally stage 5 or hypertensive crisis (>180/>120).

(Dtype: int64, “1”, means “Yes”, “0” means “No”)

Diastolic and systolic pressure generally rise together.

Systolic Blood Pressure (sysBP): it measures the force which heart must pump against to get blood to flow around the body. A normal systolic blood pressure is below 120 mm/Hg while a systolic blood pressure between 120 mm/Hg and 129 mm/Hg indicates an elevate hypertension.

(Dtype: float64)

Diastolic Blood Pressure (diaBP): This measures the resting pressure when the heart relaxes between heartbeats. A diastolic reading greater than 90 mmHg is considered high.

> An abnormally low diastolic blood pressure, less than 60 mm/Hg, might be too low to maintain adequate blood flow to your brain and other organs. Low diastolic blood pressure can make people feel lightheaded and dizzy; you could faint or collapse if your diastolic blood pressure drops too low (Perkins, S. 2018).

(Dtype: float64)

Blood Pressure Medications (BPMeds): Drugs used to treat High Blood Pressure.

(Dtype: float64, “1”, means “Yes”, “0” means “No”)

Prevalent Stroker: It is a type of "brain attack”. High blood pressure is a huge factor, doubling or even quadrupling your stroke risk if it is not controlled. During a stroke, the blood flow is being reduced to a human brain.

(Dtype: int64, “1”, means “Yes”, “0” means “No”)

Heart Rate:

> “*A healthy heart It speeds up and slows down to accommodate your changing need for oxygen as your activities vary throughout the day. What is a "normal" heart rate varying from person to person. However, an unusually high resting heart rate or low maximum heart rate may signify an increased risk of heart attack and death”.* (Harvard University, 2020)

(Dtype: float64)

Average heart rates by age

Age in years	Average maximum heart rate in beats per minute	Target heart rate range in beats per minute
40	180	90 to 153
45	175	88 to 149
50	170	85 to 145
55	165	83 to 140
60	160	80 to 136
65	155	78 to 132
70	150	75 to 128

Source: American Heart Association.

Body Mass Index (BMI): According to Centres for Disease, Control and Preventions(2020). It is a person’s weight in kilograms divided by the square of height in meters. It ranks weight per category: underweight (Below 18.5), healthy weight (18.5 – 24.9), overweight (25.0 – 29.9), and obesity (30.0 and Above).

(Dtype: float64)

4.4. Data preparation

Once we selected the dataset, we check if there are inconsistencies / missing data in the dataset we selected and prepare selected data for subsequent analysis / model build. To do the preparation we decided to use Jupyter Notebook.

In Annex 3 we provide Jupyter code we did to prepare the data with all the comments

5. Conclusion

In the first part of our work, we were dealing with the business understanding of the problem, data selection and data preparation. We found a wide variety of topics in on different dataset, but it was not easy to find a good data, with enough attributes and arrows for the project. Once our data was selected, we clean it, preparing it for modelling. The next step will be analysing data, create a model that will predict and categorize the riskiest factors to develop CHD.

6. References

American Heart Association. (2020). *Heart*. Available at: https://www.heart.org/en/health-topics/high-blood-pressure/understanding-blood-pressure-readings [Accessed 28th November 2020].

Centres for Disease, Control and Preventions. (2020). *CDC.* Available at: https://www.cdc.gov/healthyweight/assessing/bmi/adult_bmi/index.html [Accessed 29th November 2020].

Harvard University. (2020). *Harvard Health Publishing.* Available at: https://www.health.harvard.edu/heart-health/what-your-heart-rate-is-telling-you [Accessed 24th November 2020].

HSE. (2020). *HSE*. Available at: https://www.hse.ie/eng/health/hl/living/diabetes/diabetesinformation.html [Accessed 29th November 2020].

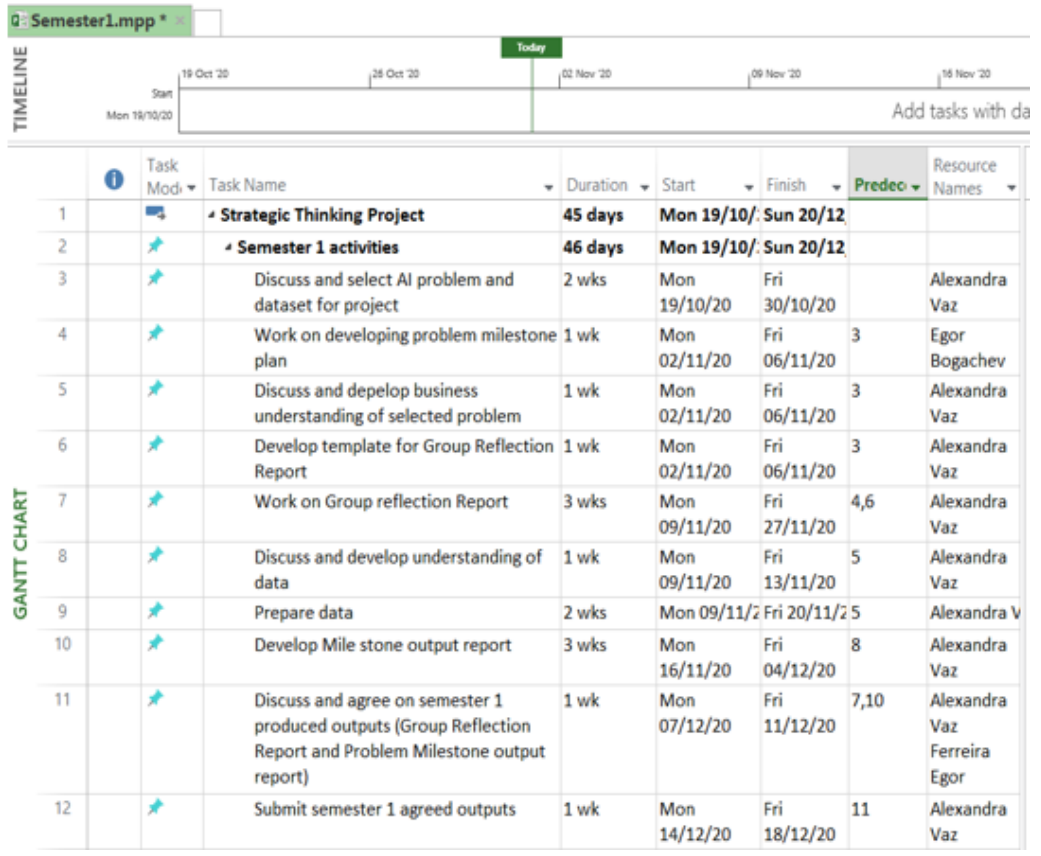
Ongoing Heart Study. *Kaggle.* Available at: https://www.kaggle.com/amanajmera1/framingham-heart-study-dataset/data[Accessed 3rd December 2020].

Perkins, S. (2018). *SFGATE*. Available at: https://healthyeating.sfgate.com/healthy-diastolic-ranges-4409.html [Accessed 23th November 2020].

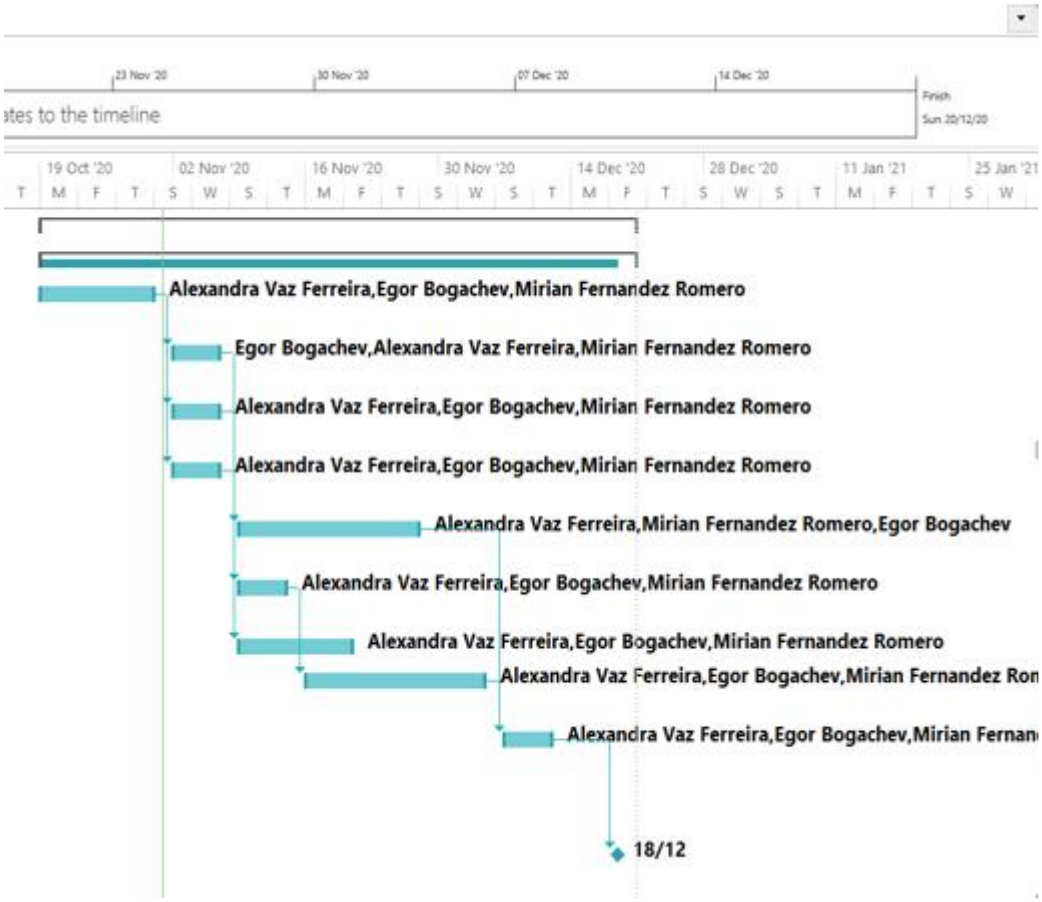
U.S National Library of Medicine (2020). *MedlinePlus*. Available at: https://medlineplus.gov/cholesterollevelswhatyouneedtoknow.html [Accessed 27th November 2020].

World Health Organization (2020). *Who*. Available at: https://www.who.int/health-topics/cardiovascular-diseases/#tab=tab_1 [Accessed 3rd December 2020].

Appendix 1. Gantt Chart / Semester 1 Plan



	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names
1		**Strategic Thinking Project**	**45 days**	**Mon 19/10/**	**Sun 20/12**		
2		**Semester 1 activities**	**46 days**	**Mon 19/10/**	**Sun 20/12**		
3		Discuss and select AI problem and dataset for project	2 wks	Mon 19/10/20	Fri 30/10/20		Alexandra Vaz
4		Work on developing problem milestone plan	1 wk	Mon 02/11/20	Fri 06/11/20	3	Egor Bogachev
5		Discuss and depelop business understanding of selected problem	1 wk	Mon 02/11/20	Fri 06/11/20	3	Alexandra Vaz
6		Develop template for Group Reflection Report	1 wk	Mon 02/11/20	Fri 06/11/20	3	Alexandra Vaz
7		Work on Group reflection Report	3 wks	Mon 09/11/20	Fri 27/11/20	4,6	Alexandra Vaz
8		Discuss and develop understanding of data	1 wk	Mon 09/11/20	Fri 13/11/20	5	Alexandra Vaz
9		Prepare data	2 wks	Mon 09/11/2	Fri 20/11/2	5	Alexandra V
10		Develop Mile stone output report	3 wks	Mon 16/11/20	Fri 04/12/20	8	Alexandra Vaz
11		Discuss and agree on semester 1 produced outputs (Group Reflection Report and Problem Milestone output report)	1 wk	Mon 07/12/20	Fri 11/12/20	7,10	Alexandra Vaz Ferreira Egor
12		Submit semester 1 agreed outputs	1 wk	Mon 14/12/20	Fri 18/12/20	11	Alexandra Vaz



Appendix 2. Group Reflexion Report

Week: 26/10/2020 to 30/10/2020

- Topics to be discussed (goals):
  - Discuss and select AI problem and dataset for project.
- Participants:
  - Egor Bogachev
  - Alexandra Vaz Ferreira
  - Míriam Fernández Romero
- Goals:
  - ✓ Achieved:
    - Yes.
  - ✓ Not achieved:
- Team work: All group members are proactive during the discussions. We have a good and fluid communication between us. We are going through all project parts together.
- Challenges: If yes, How do we overcome this challenges?: N/A

Week: 02/11/2020 to 06/11/2020

- Topics to be discussed (goals):
  - Work on developing problems milestone plan (1).
  - Discuss and develop businesses understanding of selected problem (2).
  - Develop template for Group Reflexion Report (3).
- Participants:
  - Egor Bogachev
  - Alexandra Vaz Ferreira
  - Míriam Fernández Romero
- Goals:
  - ✓ Achieved:
    - Develop business understanding of the problem and compile business understanding report.
    - (3)Discussed and agreed on 04/11/20.
  - ✓ Not achieved:
- Team work: short meeting on 04/11/20 to clarify what will be done next. All group members are proactive during the discussions. We have a good and fluid communication between us. We are going through all project parts together.
- Challenges: If yes, How do we overcome this challenges?: Not sufficient knowledge in machine learning and AI to make decision on projects.

Week: 09/11/2020 to 13/11/2020

- Topics to be discussed (goals):
  - Look a dataset and develop data understanding.
  - Work on group reflection report.
- Participants:
  - Egor Bogachev
  - Alexandra Vaz Ferreira
  - Míriam Fernández Romero.
- Goals:
  - ✓ Achieved: we went through the different datasets chosen; analysing the features and finally deciding the one we will use for our project.
  - ✓ Not achieved:
- Team work: We are going through all project parts together.
- Challenges: If yes, How do we overcome this challenges?:

Week: 16/11/2020 to 20/11/2020

- Topics to be discussed (goals):
  - Develop Milestone output report.
  - Work on group reflection report.
  - Data preparation.
- Participants:
  - Egor Bogachev
  - Alexandra Vaz Ferreira
  - Míriam Fernández Romero
- Goals:
  - ✓ Achieved: We have started preparing data as there were some Null values. In order to do so, it was chosen to take the average in each one of the variables. Still working on it.
  - ✓ Not achieved:
- Team work:
  - Egor cleaned data.
  - Alexandra will continue with the Data Understanding Report.
  - Miriam will continue with the group reflexion updates.
- Challenges: If yes, How do we overcome this challenges?:

Week: 23/11/2020 to 27/11/2020

- Topics to be discussed (goals):
  - Develop Milestone output report.
  - Work on group reflection report.
  - Data preparation.
- Participants:
  - Egor Bogachev
  - Alexandra Vaz Ferreira
  - Míriam Fernández Romero
- Goals:
  - ✓ Achieved:
    - Develop Milestone output report with all the chapters needs to be. How we can combine it with what is already done.
    - Work on group reflection report.
    - Data preparation. We went all through the data, checking all features, if there are some missing values and adding a short description of each to the data understanding report.
  - ✓ Not achieved:
- Teamwork:
  - All group members are proactive during the discussions. We have a good and fluid communication between us. We are going through all project parts together which is good as we are involved in the whole process.
- Challenges: If yes, how do we overcome this challenges:

Week: 07/12/2020 to 11/12/2020

- Topics to be discussed (goals):
  - Problem Milestone output report.
  - Work on group reflection report.
- Participants:
  - Egor Bogachev
  - Alexandra Vaz Ferreira
  - Míriam Fernández Romero
- Goals:
  - ✓ Achieved:
    - We went all through all the parts of Problem Milestone output report, checking the content to agree if we need to add/modify or delete information.
    - Work on group reflexion report.
  - ✓ Not achieved:
- Teamwork:
  - We checked all the parts together.
- Challenges: If yes, how do we overcome this challenges:

Appendix 3: Jupyter code on dataset preparation

The dataset preparation for Gtrategic Thinking module project AIP3 group

In [11]:
```
import pandas as pd
import seaborn as sns
import numpy as np
```

In below two cells we read dataset downloaded from kagge.com and create the dataframe to work with. Dataset located in https://www.kaggle.com/dileep070/heart-disease-prediction-using-logistic-regression

In [7]:
```
database3_url = "framingham.csv"
```

In [8]:
```
database3_df=pd.read_csv(database3_url)
```

Below cell shows shows first 4 rows out above dataset

In [9]:
```
database3_df.head()
```

Out[9]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diab
0	1	39	4.0	0	0.0	0.0	0	0	
1	0	46	2.0	0	0.0	0.0	0	0	
2	1	48	1.0	1	20.0	0.0	0	0	
3	0	61	3.0	1	30.0	0.0	0	1	
4	0	46	3.0	1	23.0	0.0	0	0	

In below two cells we list basic information on all columns in dataset.

In [6]:
```
database3_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4238 entries, 0 to 4237
Data columns (total 16 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   male             4238 non-null   int64  
 1   age              4238 non-null   int64  
 2   education        4133 non-null   float64
 3   currentSmoker    4238 non-null   int64  
 4   cigsPerDay       4209 non-null   float64
 5   BPMeds           4185 non-null   float64
 6   prevalentStroke  4238 non-null   int64  
 7   prevalentHyp     4238 non-null   int64  
 8   diabetes         4238 non-null   int64  
 9   totChol          4188 non-null   float64
 10  sysBP            4238 non-null   float64
 11  diaBP            4238 non-null   float64
 12  BMI              4219 non-null   float64
 13  heartRate        4237 non-null   float64
 14  glucose          3850 non-null   float64
 15  TenYearCHD       4238 non-null   int64  
dtypes: float64(9), int64(7)
memory usage: 529.9 KB
```

In [7]:
```
database3_df.columns
```

Out[7]:
```
Index(['male', 'age', 'education', 'currentSmoker', 'cigsPerDay', 'BPMeds',
       'prevalentStroke', 'prevalentHyp', 'diabetes', 'totChol', 'sysBP',
       'diaBP', 'BMI', 'heartRate', 'glucose', 'TenYearCHD'],
      dtype='object')
```

Below two cells list magnitude of missing data for each feature in the dataset

In [11]:
```
missing_data = database3_df.isnull().sum()
```

```
In [12]: missing_data
```

```
Out[12]: male                0
         age                 0
         education         105
         currentSmoker       0
         cigsPerDay         29
         BPMeds             53
         prevalentStroke     0
         prevalentHyp        0
         diabetes            0
         totChol            50
         sysBP               0
         diaBP               0
         BMI                19
         heartRate           1
         glucose           388
         TenYearCHD          0
         dtype: int64
```

Below two cels show the age range of patients in the dataset

```
In [9]: database3_df['age'].min()
```

```
Out[9]: 32
```

```
In [10]: database3_df['age'].max()
```

```
Out[10]: 70
```

In below two cells we evaluate total percantage of missing data

```
In [15]: total_missing = missing_data.sum()
         total_cells = np.product(database3_df.shape)
         percent_missing = (total_missing / total_cells) * 100
```

```
In [16]: print(percent_missing)
```

```
0.9512151958470976
```

In below two cells we look at mean valueof data for each featuree in the dataset and repalce the missing value in dataset with mean value of data in column.

In [17]:
```
database3_df.mean()
```

Out[17]:
```
male                0.429212
age                49.584946
education           1.978950
currentSmoker       0.494101
cigsPerDay          9.003089
BPMeds              0.029630
prevalentStroke     0.005899
prevalentHyp        0.310524
diabetes            0.025720
totChol           236.721585
sysBP             132.352407
diaBP              82.893464
BMI                25.802008
heartRate          75.878924
glucose            81.966753
TenYearCHD          0.151958
dtype: float64
```

In [24]:
```
database3_df = database3_df.fillna(database3_df.mean())
database3_df = np.round
```

Now we chack again for missing data in dataset, there are no mising data anymore.

In [25]:
```
database3_df.isnull().sum()
```

Out[25]:
```
male               0
age                0
education          0
currentSmoker      0
cigsPerDay         0
BPMeds             0
prevalentStroke    0
prevalentHyp       0
diabetes           0
totChol            0
sysBP              0
diaBP              0
BMI                0
heartRate          0
glucose            0
TenYearCHD         0
dtype: int64
```

In below x7 cells we chack for data inconcistencies after inserting mean columns values into the cells with missing data. We do it by chcking unique data values in each of x7 columns we modified. The data below shows some data inconsistencies in x6 ot of x7 colums that we modified.

In [26]:
```
database3_df.education.unique()
```

Out[26]:
```
array([4.        , 2.        , 1.        , 3.        , 1.97894992])
```

In [27]:
```
database3_df.cigsPerDay.unique()
```

Out[27]:
```
array([ 0.        , 20.        , 30.        , 23.        , 15.        ,
        9.        , 10.        ,  5.        , 35.        , 43.        ,
        1.        , 40.        ,  3.        ,  2.        ,  9.00308862,
       12.        ,  4.        , 18.        , 25.        , 60.        ,
       14.        , 45.        ,  8.        , 50.        , 13.        ,
       11.        ,  7.        ,  6.        , 38.        , 29.        ,
       17.        , 16.        , 19.        , 70.        ])
```

In [28]:
```
database3_df.BPMeds.unique()
```

Out[28]:
```
array([0.        , 1.        , 0.02962963])
```

In [29]:
```
database3_df.totChol.unique()
```

Out[29]:
```
array([195.        , 250.        , 245.        , 225.        ,
       285.        , 228.        , 205.        , 313.        ,
       260.        , 254.        , 247.        , 294.        ,
       332.        , 226.        , 221.        , 232.        ,
       291.        , 190.        , 185.        , 234.        ,
       215.        , 270.        , 272.        , 295.        ,
       209.        , 175.        , 214.        , 257.        ,
       178.        , 233.        , 180.        , 243.        ,
       237.        , 236.72158548, 311.        , 208.        ,
       252.        , 261.        , 179.        , 194.        ,
       267.        , 216.        , 240.        , 266.        ,
       255.        , 220.        , 235.        , 212.        ,
       223.        , 300.        , 302.        , 248.        ,
       200.        , 189.        , 258.        , 202.        ,
       213.        , 183.        , 274.        , 170.        ,
```

In [30]:
```
database3_df.BMI.unique()
```

Out[30]:
```
array([26.97, 28.73, 25.34, ..., 39.17, 26.7 , 43.67])
```

In [31]:
```
database3_df.heartRate.unique()
```

Out[31]:
```
array([ 80.        ,  95.        ,  75.        ,  65.        ,
        85.        ,  77.        ,  60.        ,  79.        ,
        76.        ,  93.        ,  72.        ,  98.        ,
        64.        ,  70.        ,  71.        ,  62.        ,
        73.        ,  90.        ,  96.        ,  68.        ,
        63.        ,  88.        ,  78.        ,  83.        ,
       100.        ,  67.        ,  84.        ,  57.        ,
        50.        ,  74.        ,  86.        ,  55.        ,
        92.        ,  66.        ,  87.        , 110.        ,
        81.        ,  56.        ,  89.        ,  82.        ,
        48.        , 105.        ,  61.        ,  54.        ,
        69.        ,  52.        ,  94.        , 140.        ,
       130.        ,  58.        , 108.        , 104.        ,
        91.        ,  53.        ,  75.87892377, 106.        ,
        59.        ,  51.        , 102.        , 107.        ,
       112.        , 125.        , 103.        ,  44.        ,
        47.        ,  45.        ,  97.        , 122.        ,
       120.        ,  99.        , 115.        , 143.        ,
       101.        ,  46.        ])
```

In [32]:
```
database3_df.glucose.unique()
```

Out[32]:
```
array([ 77.        ,  76.        ,  70.        , 103.        ,
        85.        ,  99.        ,  78.        ,  79.        ,
        88.        ,  61.        ,  64.        ,  84.        ,
        81.96675325,  72.        ,  89.        ,  65.        ,
       113.        ,  75.        ,  83.        ,  66.        ,
        74.        ,  63.        ,  87.        , 225.        ,
        90.        ,  80.        , 100.        , 215.        ,
        98.        ,  62.        ,  95.        ,  94.        ,
        55.        ,  82.        ,  93.        ,  73.        ,
        45.        , 202.        ,  68.        ,  97.        ,
       104.        ,  96.        , 126.        , 120.        ,
       105.        ,  71.        ,  56.        ,  60.        ,
       117.        , 102.        ,  58.        ,  92.        ,
       109.        ,  86.        , 107.        ,  54.        ,
        67.        ,  69.        ,  57.        ,  91.        ,
       132.        , 150.        ,  59.        ,  81.        ,
       115.        , 140.        , 112.        , 118.        ,
       143.        , 114.        , 160.        , 110.        ,
       123.        , 108.        , 145.        , 122.        ,
       137.        , 106.        , 127.        , 205.        ,
       130.        , 101.        ,  47.        ,  53.        ,
       216.        , 163.        , 144.        , 116.        ,
       121.        , 172.        , 124.        , 111.        ,
        40.        , 186.        , 223.        , 325.        ,
        44.        , 156.        , 268.        ,  50.        ,
       274.        , 292.        , 255.        , 136.        ,
       206.        , 131.        , 148.        , 297.        ,
        43.        , 173.        ,  48.        , 386.        ,
       155.        , 147.        , 170.        ,  52.        ,
       320.        , 254.        , 394.        , 270.        ,
       244.        , 183.        , 142.        , 119.        ,
       135.        , 167.        , 207.        , 129.        ,
```

Because we found some data inconcistencies in the above x6 modified columns(the decimals which mean values have when we replaced the missing values), we now will round the values in the x6 out of x7 modified columns in the below cell.

In [36]:
```
roundCols = ['education','cigsPerDay','BPMeds','totChol','heartRate','glucose']
```

In [37]:
```
database3_df[roundCols] = database3_df[roundCols].round(0)
```

```
Now we check again the unique values in our modified columns (those x7 columns
in which we replaced the missing values with column mean values).
```

In [38]:
```
database3_df.education.unique()
```

Out[38]:
```
array([4., 2., 1., 3.])
```

In [39]:
```
database3_df.cigsPerDay.unique()
```

Out[39]:
```
array([ 0., 20., 30., 23., 15.,  9., 10.,  5., 35., 43.,  1., 40.,  3.,
        2., 12.,  4., 18., 25., 60., 14., 45.,  8., 50., 13., 11.,  7.,
        6., 38., 29., 17., 16., 19., 70.])
```

In [40]:
```
database3_df.BPMeds.unique()
```

Out[40]:
```
array([0., 1.])
```

```
In [41]: database3_df.totChol.unique()
```

```
Out[41]: array([195., 250., 245., 225., 285., 228., 205., 313., 260., 254., 247.,
                294., 332., 226., 221., 232., 291., 190., 185., 234., 215., 270.,
                272., 295., 209., 175., 214., 257., 178., 233., 180., 243., 237.,
                311., 208., 252., 261., 179., 194., 267., 216., 240., 266., 255.,
                220., 235., 212., 223., 300., 302., 248., 200., 189., 258., 202.,
                213., 183., 274., 170., 210., 197., 326., 188., 256., 244., 193.,
                239., 296., 269., 275., 268., 265., 173., 273., 290., 278., 264.,
                282., 241., 288., 222., 303., 246., 150., 187., 286., 154., 279.,
                293., 259., 219., 230., 320., 312., 165., 159., 174., 242., 301.,
                167., 308., 325., 229., 236., 224., 253., 464., 171., 186., 227.,
                249., 176., 163., 191., 263., 196., 310., 164., 135., 238., 207.,
                342., 287., 182., 352., 284., 217., 203., 262., 129., 155., 323.,
                206., 283., 319., 304., 340., 328., 280., 368., 218., 276., 339.,
                231., 198., 177., 201., 277., 184., 199., 168., 292., 305., 306.,
                152., 161., 181., 251., 271., 370., 439., 145., 330., 157., 398.,
                162., 314., 166., 160., 281., 289., 355., 307., 156., 329., 143.,
                211., 298., 334., 192., 204., 318., 309., 353., 360., 335., 158.,
                372., 346., 169., 140., 324., 600., 315., 392., 322., 149., 137.,
                172., 317., 358., 153., 345., 391., 410., 297., 356., 338., 107.,
                148., 366., 333., 327., 344., 126., 365., 362., 316., 144., 351.
```

```
In [42]: database3_df.BMI.unique()
```

```
Out[42]: array([26.97, 28.73, 25.34, ..., 39.17, 26.7 , 43.67])
```

```
In [43]: database3_df.heartRate.unique()
```

```
Out[43]: array([ 80.,  95.,  75.,  65.,  85.,  77.,  60.,  79.,  76.,  93.,  72.,
                 98.,  64.,  70.,  71.,  62.,  73.,  90.,  96.,  68.,  63.,  88.,
                 78.,  83., 100.,  67.,  84.,  57.,  50.,  74.,  86.,  55.,  92.,
                 66.,  87., 110.,  81.,  56.,  89.,  82.,  48., 105.,  61.,  54.,
                 69.,  52.,  94., 140., 130.,  58., 108., 104.,  91.,  53., 106.,
                 59.,  51., 102., 107., 112., 125., 103.,  44.,  47.,  45.,  97.,
                122., 120.,  99., 115., 143., 101.,  46.])
```

```
In [44]: database3_df.glucose.unique()
```

```
Out[44]: array([ 77.,  76.,  70., 103.,  85.,  99.,  78.,  79.,  88.,  61.,  64.,
                 84.,  82.,  72.,  89.,  65., 113.,  75.,  83.,  66.,  74.,  63.,
                 87., 225.,  90.,  80., 100., 215.,  98.,  62.,  95.,  94.,  55.,
                 93.,  73.,  45., 202.,  68.,  97., 104.,  96., 126., 120., 105.,
                 71.,  56.,  60., 117., 102.,  58.,  92., 109.,  86., 107.,  54.,
                 67.,  69.,  57.,  91., 132., 150.,  59.,  81., 115., 140., 112.,
                118., 143., 114., 160., 110., 123., 108., 145., 122., 137., 106.,
                127., 205., 130., 101.,  47.,  53., 216., 163., 144., 116., 121.,
                172., 124., 111.,  40., 186., 223., 325.,  44., 156., 268.,  50.,
                274., 292., 255., 136., 206., 131., 148., 297.,  43., 173.,  48.,
                386., 155., 147., 170.,  52., 320., 254., 394., 270., 244., 183.,
                142., 119., 135., 167., 207., 129., 177., 250., 294., 166., 125.,
                332., 368., 348., 248., 370., 193., 191., 256., 235., 210., 260.])
```

The data now looks concistent and ready for further analysis / model build.