How does different health related factors effects on the chances of the heart attack? And analysis of the heart attack symptoms and prediction.

April 2024

Aibassov Danial

Content

Introduction

Today, heart disease remains one of the leading causes of death worldwide, causing serious harm to human health and public health in general. In this regard, the study of stress factors, lifestyle changes and demographic dynamics, the study of factors influencing the risk of a heart attack, becomes extremely important.

A heart attack (cardiovascular diseases) occurs when the flow of blood to the heart muscle suddenly becomes blocked. From WHO statistics every year 17.9 million dying from heart attack. The medical study says that human life style is the main reason behind this heart problem. Apart from this there are many key factors which warns that the person may/may not getting chance of heart attack.

Our project is devoted to the analysis of the relationships between various risk factors and the likelihood of cardiac effects using econometric methods, which are based on the theory of regression analysis. It aims to identify the key principles that have the greatest impact on the likelihood of an incident occurring, and to assess the extent and direction of the event.

To achieve our goals, we will analyze the data available to us, including information about the consequences, lifestyle, genetic predispositions, and other factors that may be associated with the risk of a heart attack. Using modern econometric techniques, we estimate regression models to identify significant factors and provide practical recommendations for reducing the risk of heart attack.

Our work has the potential not only to advance theoretical understanding of the factors influencing heart disease, but also to help develop more effective methods for preventing and treating this serious medical condition. And also, in order to understand how good our model is, we use different methods of testing our regression for its accuracy.

Main Part

In this chapter, we will break down our regression model: how it works and what variables are included. In addition, we will tell you which programming tool and which libraries we used.

We have used open-sourced dataset available online. With 14 different attributes and 304 different observations. Data was taken from different hospital in Central Asia. Names, exact hospitals and countries were hidden.

In order to develop our software, we used Python as the main programming language. To build the model in Python, libraries such as *matplotlib, pandas, NumPy, seaborn and sklearn* were used. These libraries were needed for developing graphs, using statistical calculations, and working with a database.

As we see in the new line, we use the pandas function to show the view and data of the table that we took as the basis of our database. Our dataset contains information about patients with heart disease. It includes information such as age, sex, chest pain type, resting blood pressure, cholesterol levels, fasting blood sugar levels, resting electrocardiographic results, maximum heart rate achieved, exercise induced angina, oldpeak (ST depression induced by exercise relative to rest), slope of the peak exercise ST segment, number of major vessels colored by fluoroscopy, thalassemia (a type of blood disorder), and the presence or absence of heart disease. However, given variables are very specific and was taken from medicine, therefore confusing to understand. So, we have provided data dictionary:

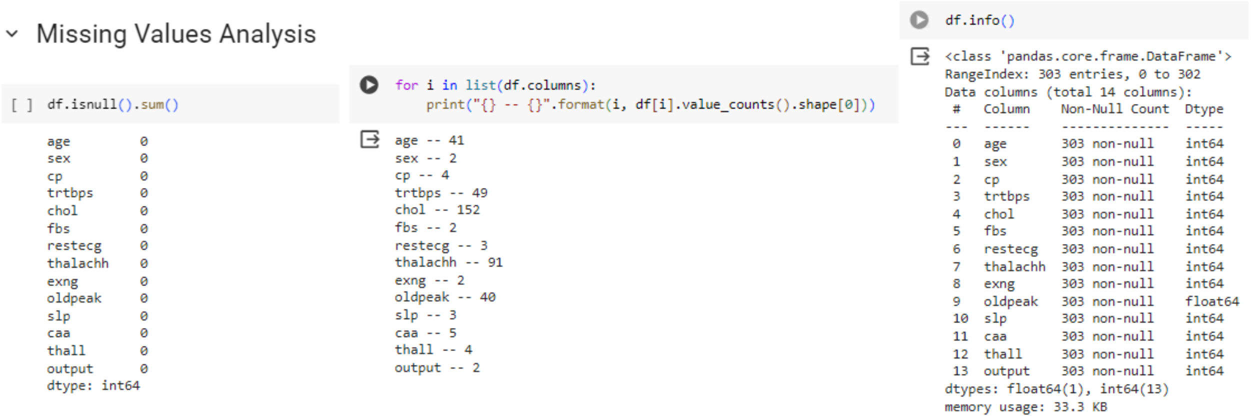
1. Age: Age of the patient
2. Sex: Sex of the patient
3. cp: Chest Pain type chest pain type ~
    a. Value 1: Typical Angina
    b. Value 2: Atypical Angina
    c. Value 3: Non-anginal Pain
    d. Value 4: Asymptomatic
4. trtbps: Resting blood pressure (in mm/Hg)
5. chol: Cholestoral in mg/dl fetched via BMI sensor
6. fbs: (fasting blood sugar > 120 mg/dl) (1 = true; 0 = false)
7. restecg: Resting electrocardiographic results
    a. Value 0: normal
    b. Value 1: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV)

    c. Value 2: showing probable or definite left ventricular hypertrophy by Estes' criteria

8. thalachh: Maximum heart rate achieved
9. oldpeak: Previous peak
10. slp: Slope
11. caa: Number of major vessels (0-3)
12. exng: Exercise induced angina (1 = yes; 0 = no)
13. thall: Thalassemia
14. output: Diagnosis of heart disease (0= false 1= true)

Angina, plays cruel role in the identifying the probability of heart disease. However, in order to make precise conclusions, and in general understand the correlation between different variables, we have to more deeply understand the terms. So, in simple words, angina may feel like pressure in the chest, jaw or arm. It frequently may occur with exercise or stress. Some people with angina also report feeling lightheaded, overly tired, short of breath or nauseated. As the heart pumps harder to keep up with what you are doing, it needs more oxygen-rich blood. However, presence of Angina it is not the risk factor of the heart attack. The key difference between angina and a heart attack is that angina is the result of narrowed (rather than blocked) coronary arteries. This is why, unlike a heart attack, angina does not cause permanent heart damage. But symptoms are so similar, it may be hard to tell neither it is angina or heart attack. Angina has two types: stable and unstable. Stable angina is chest pain that occurs when your heart is working hard and needs more oxygen. The flow of blood can't keep up with the demand. This type of angina goes away when you rest or take medication. While, Unstable angina is chest pain that happens even when your heart is not working hard. This type of angina may be a warning sign of a heart attack. People with presence of Unstable angina are more likely to have a heart attack. However, dataset do not provide the information about what exact type of Angina is occurred.

Data Preprocessing



Missing Values Analysis

```
[ ] df.isnull().sum()

    age         0
    sex         0
    cp          0
    trtbps      0
    chol        0
    fbs         0
    restecg     0
    thalachh    0
    exng        0
    oldpeak     0
    slp         0
    caa         0
    thall       0
    output      0
    dtype: int64
```

```
for i in list(df.columns):
    print("{} -- {}".format(i, df[i].value_counts().shape[0]))

age -- 41
sex -- 2
cp -- 4
trtbps -- 49
chol -- 152
fbs -- 2
restecg -- 3
thalachh -- 91
exng -- 2
oldpeak -- 40
slp -- 3
caa -- 5
thall -- 4
output -- 2
```

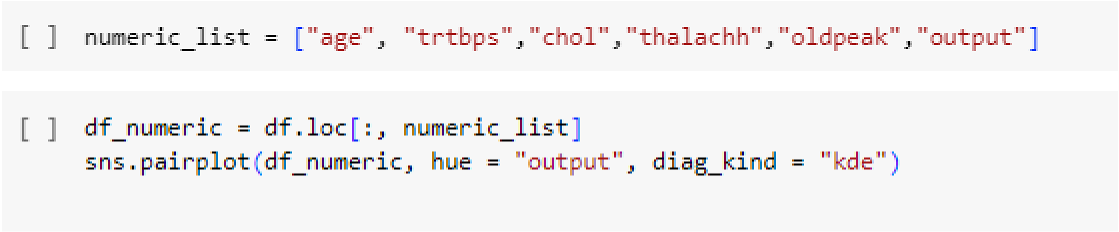
```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       303 non-null    int64
 1   sex       303 non-null    int64
 2   cp        303 non-null    int64
 3   trtbps    303 non-null    int64
 4   chol      303 non-null    int64
 5   fbs       303 non-null    int64
 6   restecg   303 non-null    int64
 7   thalachh  303 non-null    int64
 8   exng      303 non-null    int64
 9   oldpeak   303 non-null    float64
 10  slp       303 non-null    int64
 11  caa       303 non-null    int64
 12  thall     303 non-null    int64
 13  output    303 non-null    int64
dtypes: float64(1), int64(13)
memory usage: 33.3 KB
```

Picture 1

After we reviewed and verified that our data was correct, we decided to check the data for integrity and the presence of all data by row. After this, we count the number of true values, that is, missing values, in each column. As depicted, there is no any null values. Then create a loop that will loop through all the columns to count the unique values in each column.

Numeric Feature Analysis



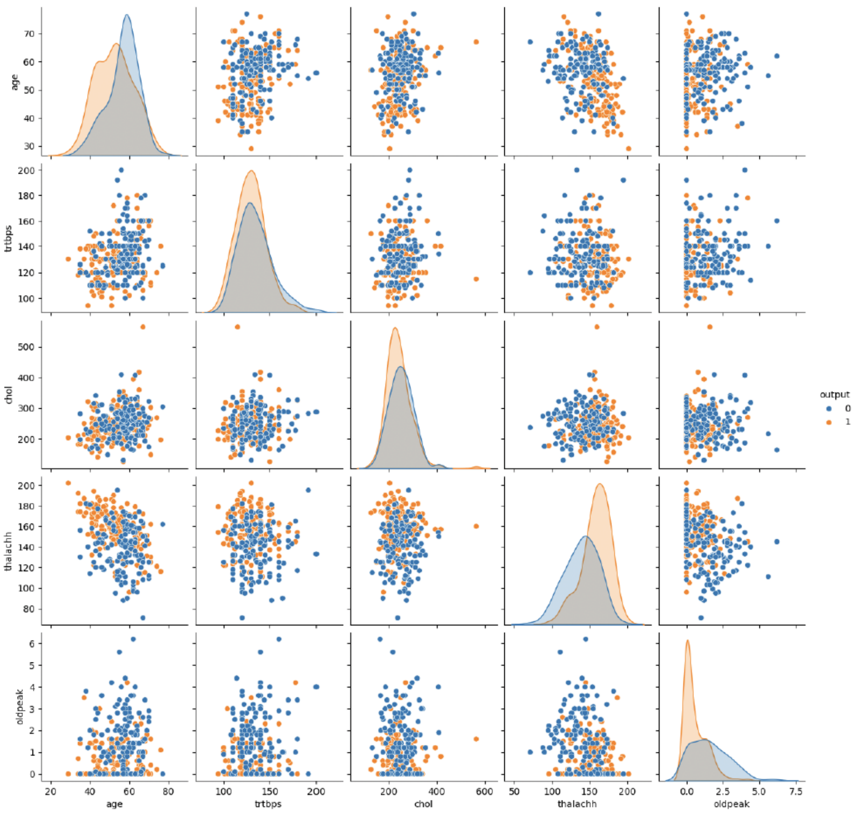
```
[ ] numeric_list = ["age", "trtbps","chol","thalachh","oldpeak","output"]
```

```
[ ] df_numeric = df.loc[:, numeric_list]
    sns.pairplot(df_numeric, hue = "output", diag_kind = "kde")
```

Picture 2

In 2nd picture we have created a numeric list which contains the names of the columns that will be used for plotting. The numeric list includes age, trtbps, chol, thalachh, oldpeak, output. After that, we extract the numeric columns to build pairwise graphs. We create graphs for numerical analysis.

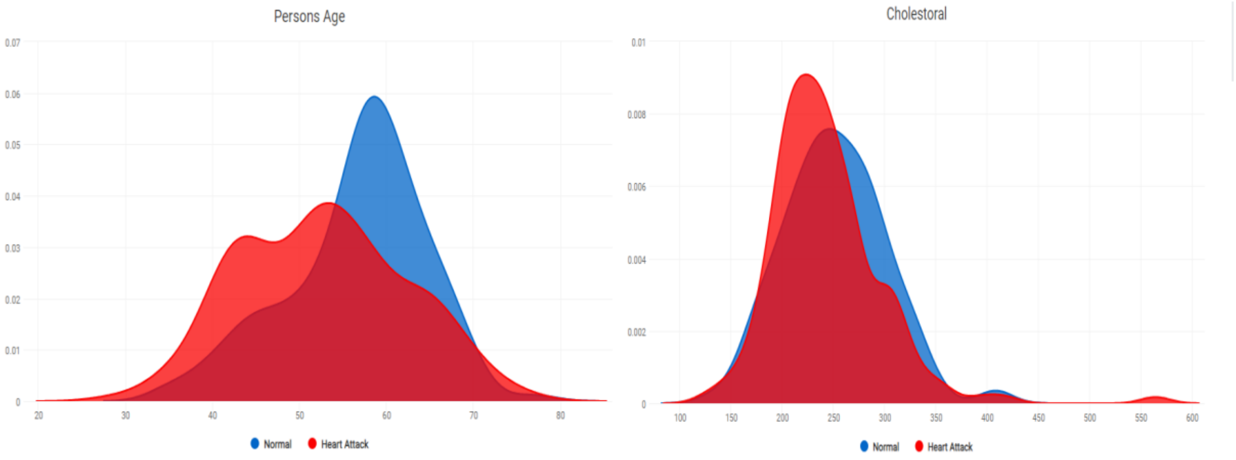
As we see in Picture 3, the result of our pairwise graph is shown where the color of the points on the graphs is determined by the value in the “output” column. Thanks to this, it is possible to visualize the relationship between different numerical characteristics that depend on “output”.



Picture 3

Bivariate Analysis

After analyzing the numeric features of different values, we have conducted the bivariate analysis, between chance of heart attack and other key different variables, looking for the how two variables are related to each other and distributed with each other. Such analysis, is much easier to understand the distribution between heart attack and other chosen variables. Hence, we conducted dense plots for quantitative data, between heart attack and other variables:



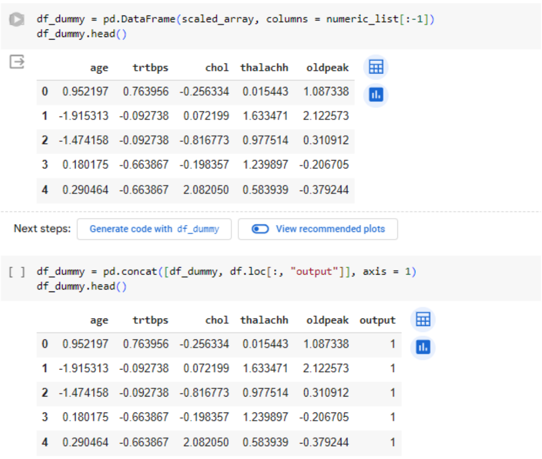
Picture 4

Blue color, is normal. Red color, is for heart attack occurring. As depicted, through dense plots, we can see that most cases of heart attack happened between approximately 45-55 years, and with cholestoral levels between approximately 190 – 270 mg/dl.

Standardization

In order to improve the convergence of models and improve the interpretation of results, we have standardized our data. We have standardized our data because, the input data from the dataset had large difference between range. It could cause some troubles for our ML model, the difference in range of initial data was large.

Box plots Analysis

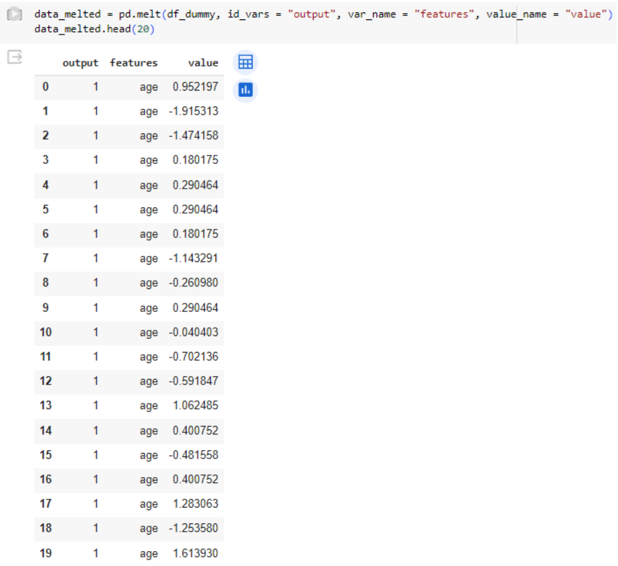


```
df_dummy = pd.DataFrame(scaled_array, columns = numeric_list[:-1])
df_dummy.head()
```

	age	trtbps	chol	thalachh	oldpeak
0	0.952197	0.763956	-0.256334	0.015443	1.087338
1	-1.915313	-0.092738	0.072199	1.633471	2.122573
2	-1.474158	-0.092738	-0.816773	0.977514	0.310912
3	0.180175	-0.663867	-0.198357	1.239897	-0.206705
4	0.290464	-0.663867	2.082050	0.583939	-0.379244

```
[ ] df_dummy = pd.concat([df_dummy, df.loc[:, "output"]], axis = 1)
    df_dummy.head()
```

	age	trtbps	chol	thalachh	oldpeak	output
0	0.952197	0.763956	-0.256334	0.015443	1.087338	1
1	-1.915313	-0.092738	0.072199	1.633471	2.122573	1
2	-1.474158	-0.092738	-0.816773	0.977514	0.310912	1
3	0.180175	-0.663867	-0.198357	1.239897	-0.206705	1
4	0.290464	-0.663867	2.082050	0.583939	-0.379244	1



```
data_melted = pd.melt(df_dummy, id_vars = "output", var_name = "features", value_name = "value")
data_melted.head(20)
```

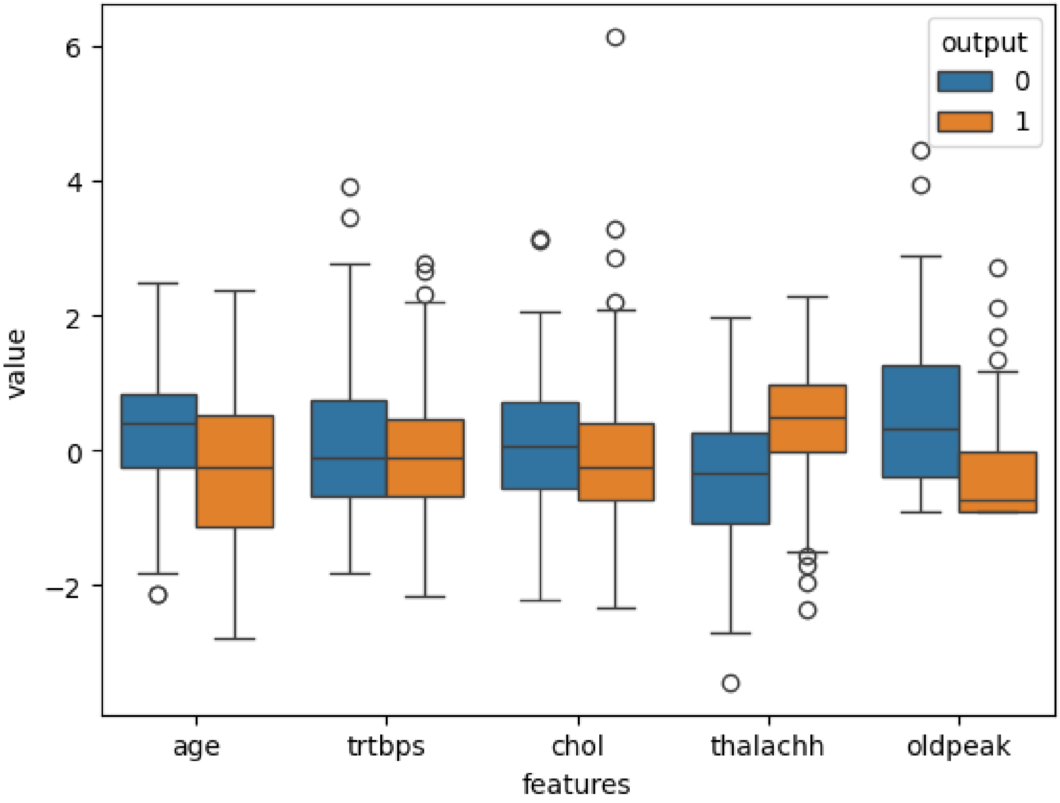
	output	features	value
0	1	age	0.952197
1	1	age	-1.915313
2	1	age	-1.474158
3	1	age	0.180175
4	1	age	0.290464
5	1	age	0.290464
6	1	age	0.180175
7	1	age	-1.143291
8	1	age	-0.260980
9	1	age	0.290464
10	1	age	-0.040403
11	1	age	-0.702136
12	1	age	-0.591847
13	1	age	1.062485
14	1	age	0.400752
15	1	age	-0.481558
16	1	age	0.400752
17	1	age	1.283063
18	1	age	-1.253580
19	1	age	1.613930

Picture 5

In Picture 5 we are creating a new dataframe df_dummy from the standardized scaled_array data. After that, we output the rows from the new dataframe to check them.

After this new one, we convert df_dummy from a wide format to a long format for further use in boxplot analysis. var_name="features" is a new column that will contain the df_dummy column names except the "output" column. value_name="value" is also a new column in which similar actions occur. Using data_melted.head(20) we print the first 20 rows of the transformed data_melted in order to test them. In Picture 7 we can see the transformed view of our dataframe.

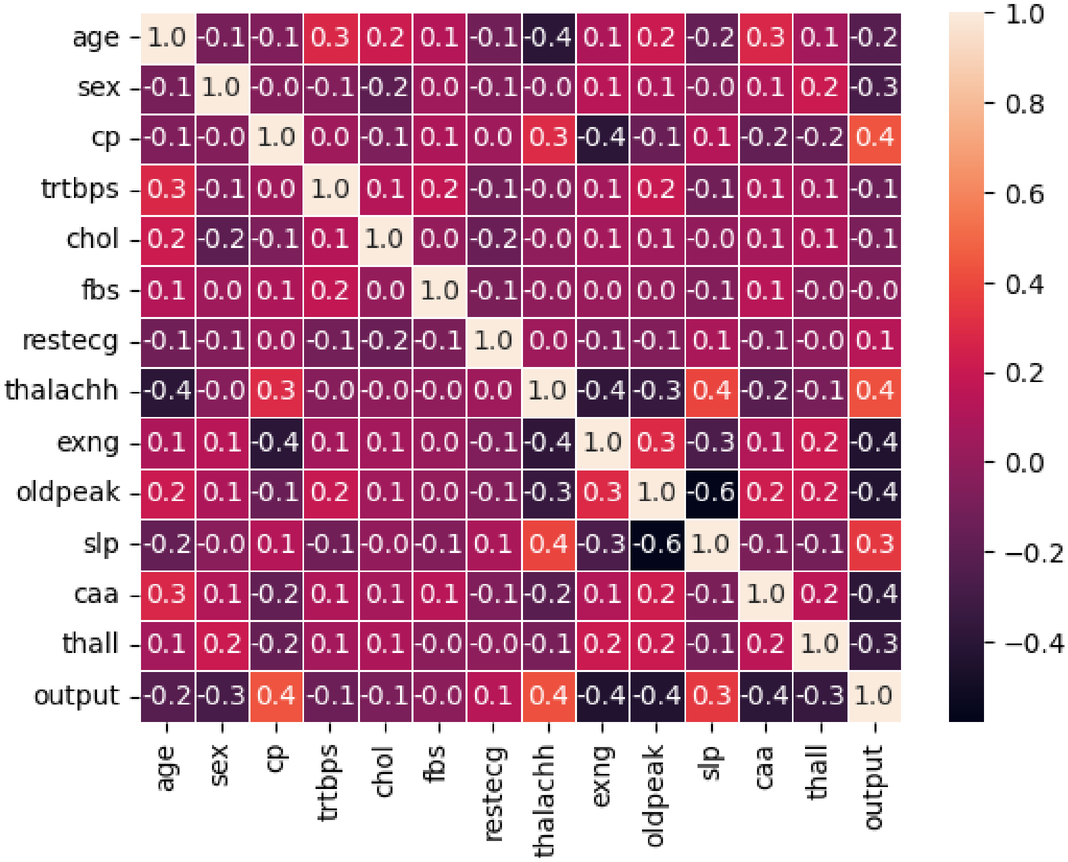
We used the code sns.boxplot(x="features", y="value", hue="output", data=data_melted) in Picture 6 to create a box plots using the seaborn library.



Picture 5

Picture % demonstrates the box plots. We have created box plots, because, box plots visually show distribution of data, also their helped us to side-by-side compare multiple distributions and we can visually identify the outliers, unusual data points. In the provided box plot, outliers represented by tiny empty circles. Also, we see difference in medians and spread of values between groups. As before, 1 – represents heart attack, while 0 – represents normal state of heart. From the box plots, we can conclude that, in general middle-aged people (40-60). *Trtbps* has week or slightly relation with heart attack. *Chol feature* (Cholestoral in mg/dl) also has week relation with heart attack. *Thalachh* (Maximum Heart Rate Achieved) has positive relation with heart attack. *Oldpeak* (Previous Peak) has negative relation with heart attack.

Correlation Analysis



	age	sex	cp	trtbps	chol	fbs	restecg	thalachh	exng	oldpeak	slp	caa	thall	output
age	1.0	-0.1	-0.1	0.3	0.2	0.1	-0.1	-0.4	0.1	0.2	-0.2	0.3	0.1	-0.2
sex	-0.1	1.0	-0.0	-0.1	-0.2	0.0	-0.1	-0.0	0.1	0.1	-0.0	0.1	0.2	-0.3
cp	-0.1	-0.0	1.0	0.0	-0.1	0.1	0.0	0.3	-0.4	-0.1	0.1	-0.2	-0.2	0.4
trtbps	0.3	-0.1	0.0	1.0	0.1	0.2	-0.1	-0.0	0.1	0.2	-0.1	0.1	0.1	-0.1
chol	0.2	-0.2	-0.1	0.1	1.0	0.0	-0.2	-0.0	0.1	0.1	-0.0	0.1	0.1	-0.1
fbs	0.1	0.0	0.1	0.2	0.0	1.0	-0.1	-0.0	0.0	0.0	-0.1	0.1	-0.0	-0.0
restecg	-0.1	-0.1	0.0	-0.1	-0.2	-0.1	1.0	0.0	-0.1	-0.1	0.1	-0.1	-0.0	0.1
thalachh	-0.4	-0.0	0.3	-0.0	-0.0	-0.0	0.0	1.0	-0.4	-0.3	0.4	-0.2	-0.1	0.4
exng	0.1	0.1	-0.4	0.1	0.1	0.0	-0.1	-0.4	1.0	0.3	-0.3	0.1	0.2	-0.4
oldpeak	0.2	0.1	-0.1	0.2	0.1	0.0	-0.1	-0.3	0.3	1.0	-0.6	0.2	0.2	-0.4
slp	-0.2	-0.0	0.1	-0.1	-0.0	-0.1	0.1	0.4	-0.3	-0.6	1.0	-0.1	-0.1	0.3
caa	0.3	0.1	-0.2	0.1	0.1	0.1	-0.1	-0.2	0.1	0.2	-0.1	1.0	0.2	-0.4
thall	0.1	0.2	-0.2	0.1	0.1	-0.0	-0.0	-0.1	0.2	0.2	-0.1	0.2	1.0	-0.3
output	-0.2	-0.3	0.4	-0.1	-0.1	-0.0	0.1	0.4	-0.4	-0.4	0.3	-0.4	-0.3	1.0

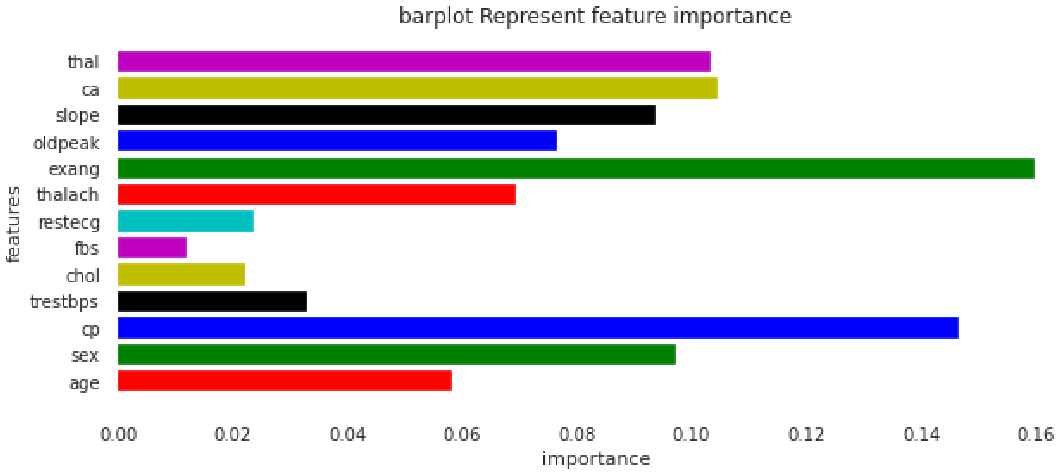
Picture 6

In Picture 6, a correlation analysis is carried out. Correlation is needed to predict and control for multicollinearity. In our snippet, we use a correlation heat map that changes color based on the level of column correlation.

In this heat map, if the values are closer to 1 then the colors are close to dark red, which means a strong positive correlation. Values close to -1 are colors close to dark blue, which means a strong negative correlation. Values close to 0 are colors close to white, which means there is no correlation between features.

Based on the results of this analysis, we can understand that the correlation between the characteristics is small in general. This means that there is no multicollinearity.

Features Analysis

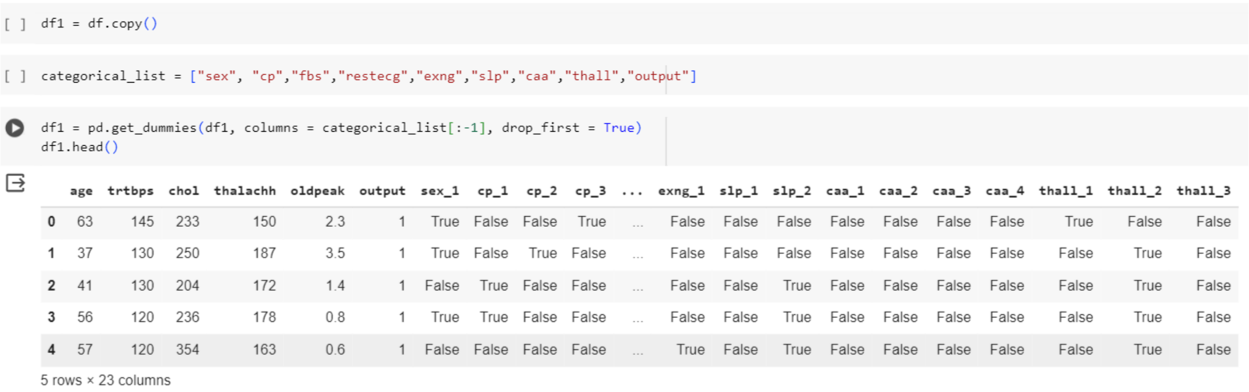


Picture 7

In Picture 7, the effect of each individual attribute is shown. In the given bar plot, the importance of each attribute is depicted. It is clearly shown that the attributes: "Exercise-induced angina (exang)" and "Chest pain (cp)" are the main symptoms of a heart attack. The definition of Angina is given on page 4. However, it is important not to ignore other symptoms and make conclusions based solely on these factors. Human health is very complicated, and regular screenings and health checks are essential.

Modeling

We have used Logistic Regression statistical technique in our heart attack prediction work. Because, our work is binary classification problem, where we our outcome variable (dependent variable) has only two possible classes. Below, we have described our stages of creating and analyzing the model.



```
[ ] df1 = df.copy()
```

```
[ ] categorical_list = ["sex", "cp","fbs","restecg","exng","slp","caa","thall","output"]
```

```
df1 = pd.get_dummies(df1, columns = categorical_list[:-1], drop_first = True)
df1.head()
```

	age	trtbps	chol	thalachh	oldpeak	output	sex_1	cp_1	cp_2	cp_3	...	exng_1	slp_1	slp_2	caa_1	caa_2	caa_3	caa_4	thall_1	thall_2	thall_3
0	63	145	233	150	2.3	1	True	False	False	True	...	False	False	False	False	False	False	False	True	False	False
1	37	130	250	187	3.5	1	True	False	True	False	...	False	False	False	False	False	False	False	False	True	False
2	41	130	204	172	1.4	1	False	True	False	False	...	False	False	True	False	False	False	False	False	True	False
3	56	120	236	178	0.8	1	True	True	False	False	...	False	False	True	False	False	False	False	False	True	False
4	57	120	354	163	0.6	1	False	False	False	False	...	True	False	True	False	False	False	False	False	True	False

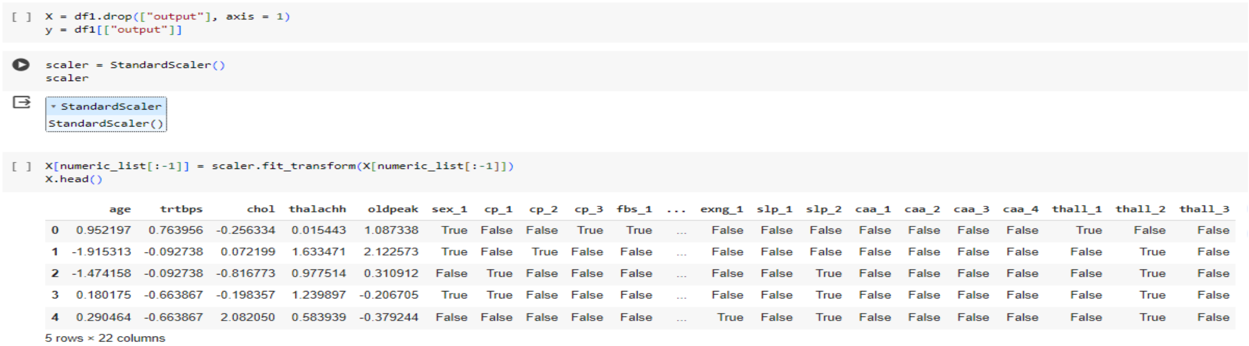
5 rows × 23 columns

Picture 8

In the modeling fragment we copy the data from df to df1 in order to avoid accidental data loss. After that, we check the list of categorical variables and use pd.get_dummies() to convert this data into dummy variables. To avoid duplicate data, we remove the first column of each categorical variable after transforming it. We end up testing our new dataframe by printing data using df1.head().

After this we create a matrix of X and Y features, which will contain all the columns from the df1 dataframe, with the exception of the "output" column. This matrix was created so that, thanks to sklearn, it would be possible to create machine learning models, where X will be a matrix of features, and Y will be a vector of the target variable.

Scaler



```
[ ] X = df1.drop(["output"], axis = 1)
    y = df1[["output"]]
```

```
scaler = StandardScaler()
scaler
```

```
▾ StandardScaler
StandardScaler()
```

```
[ ] X[numeric_list[:-1]] = scaler.fit_transform(X[numeric_list[:-1]])
    X.head()
```

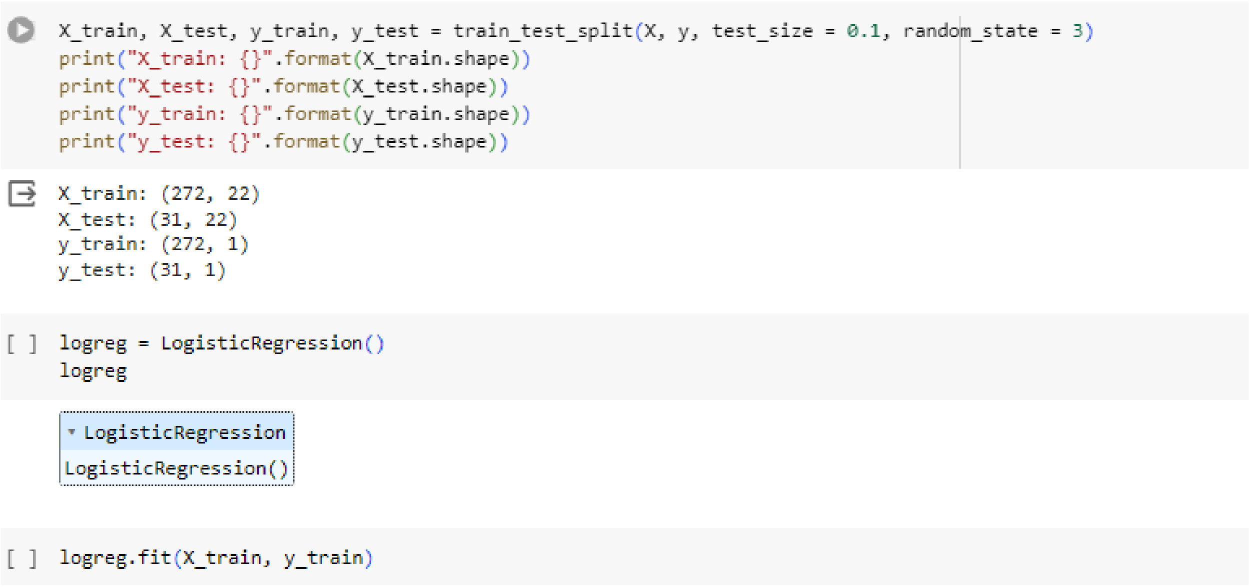
	age	trtbps	chol	thalachh	oldpeak	sex_1	cp_1	cp_2	cp_3	fbs_1	...	exng_1	slp_1	slp_2	caa_1	caa_2	caa_3	caa_4	thall_1	thall_2	thall_3
0	0.952197	0.763956	-0.256334	0.015443	1.087338	True	False	False	True	True	...	False	False	False	False	False	False	False	True	False	False
1	-1.915313	-0.092738	0.072199	1.633471	2.122573	True	False	True	False	False	...	False	False	False	False	False	False	False	False	True	False
2	-1.474158	-0.092738	-0.816773	0.977514	0.310912	False	True	False	False	False	...	False	False	True	False	False	False	False	False	True	False
3	0.180175	-0.663867	-0.198357	1.239897	-0.206705	True	True	False	False	False	...	False	False	True	False	False	False	False	False	True	False
4	0.290464	-0.663867	2.082050	0.583939	-0.379244	False	False	False	False	False	...	True	False	True	False	False	False	False	False	True	False

5 rows × 22 columns

Picture 9

In our modeling we have used scaling, in order to achieve most accurate and reliable results. It also ensures that the threshold is applied consistently across features, leading to more reliable predictions.

Train/Test Data Split



```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.1, random_state = 3)
print("X_train: {}".format(X_train.shape))
print("X_test: {}".format(X_test.shape))
print("y_train: {}".format(y_train.shape))
print("y_test: {}".format(y_test.shape))
```

```
X_train: (272, 22)
X_test: (31, 22)
y_train: (272, 1)
y_test: (31, 1)
```

```
[ ] logreg = LogisticRegression()
    logreg
```

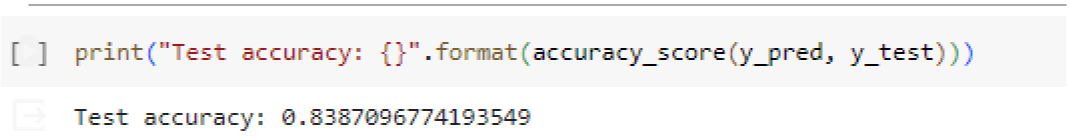
```
▾ LogisticRegression
LogisticRegression()
```

```
[ ] logreg.fit(X_train, y_train)
```

Picture 10

In given picture below, the process of splitting the data into train and test have depicted. We choose, test_size = 0.1, meaning that 0.1 proportion of the dataset is included into the test split.

Result



```
[ ] print("Test accuracy: {}".format(accuracy_score(y_pred, y_test)))
```

```
Test accuracy: 0.8387096774193549
```

Picture 10

The accuracy of our model is ~0.84. Which means we can predict the probability of the heart attack according to the given necessary variables with 84% precision.

Conclusion

In conclusion, during our project work we have conducted several analyzing techniques in order identify the factors and their effects on the chance of having heart attack. Our heart attack analysis revealed several important insights. Firstly, most heart attacks occurred in individuals aged 45-55, emphasizing the vulnerability of this age group. Secondly, cholesterol levels between 190 and 270 mg/dl were associated with higher risk. Thirdly, middle-aged people (around 40-60) were generally more affected. Regarding specific features:

- Resting Blood Pressure (Trtbps): While not a strong predictor, elevated blood pressure remains relevant.
- Cholesterol (Chol): Maintaining healthy cholesterol levels is crucial.
- Maximum Heart Rate Achieved (Thalachh): A higher maximum heart rate correlated positively with heart attacks.
- Previous Peak (Oldpeak): Lower Oldpeak values indicated higher risk.

This feature reflects the heart's response to stress. These findings can guide preventive measures and promote heart health.