Cardiovascular Disease Detection Model

Omar Kassar, Yosr Mdemagh and Ala Kerkeni

Dataset description:

Project goals:
To build an application to classify the patients to be healthy or suffering from cardiovascular disease based on the given attributes.

Dataset: collected during medical examination

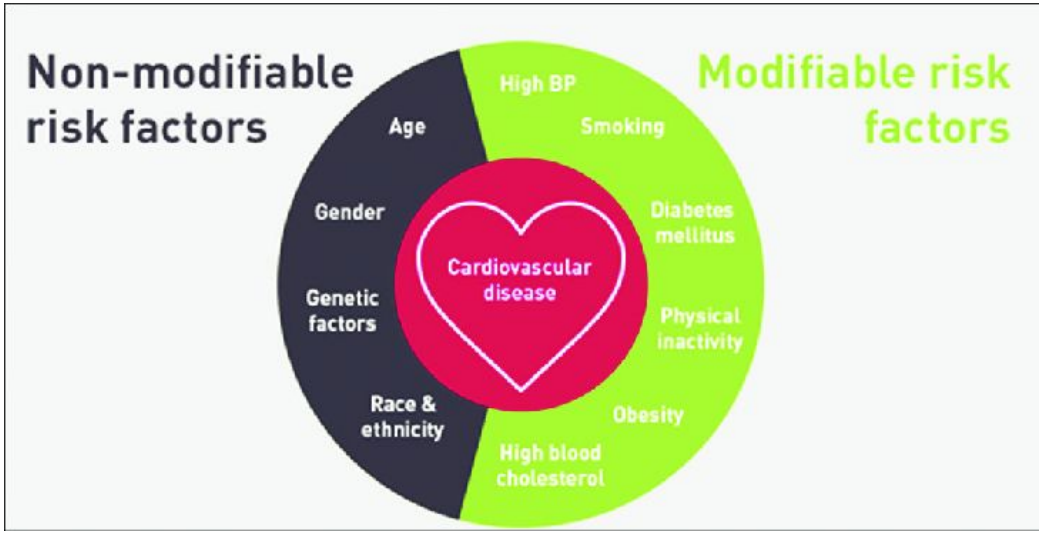
size: 69301 rows and 13 columns

Dataset Owner: Svetlana Ulianova

Data created: 20/11/2019

Missing data: there isn't

memory usage: 6.9 MB



Attribute information:

Features:

1. **age :** int (days)
2. **height** : int (cm)
3. **weight :** float (kg)
4. **gender :** categorical code
5. **Systolic blood pressure (ap_hi) :** int
6. **Diastolic blood pressure (ap_lo)** :int
7. **Cholesterol :** 1: normal, 2: above normal, 3: well above normal
8. **Glucose (gluc)** 1: normal, 2: above normal, 3: well above normal
9. **Smoking (smoke)** :binary
10. **Alcohol intake(alco)** :binary
11. **Physical activity (active)** :binary

target:

Presence or absence of cardiovascular disease (cardio) : binary

Class distribution :

	cardio
0	50.033922
1	49.966078

- We have 50% of tuples with class 0 and 50% with class 1.
- It's notable that there is equity in the class distribution

Steps:

1. Data preparation
2. Data Preprocessing and analysis
3. Model Training and Evaluation

1. Data Preparation

1. Importing the data and the necessary libraries for machine learning

```
import sklearn
from sklearn import datasets
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
[ ] ## Load data from CSV file into a DataFrame
    df=pd.read_csv("/content/cardio_train.csv",sep=";")
```

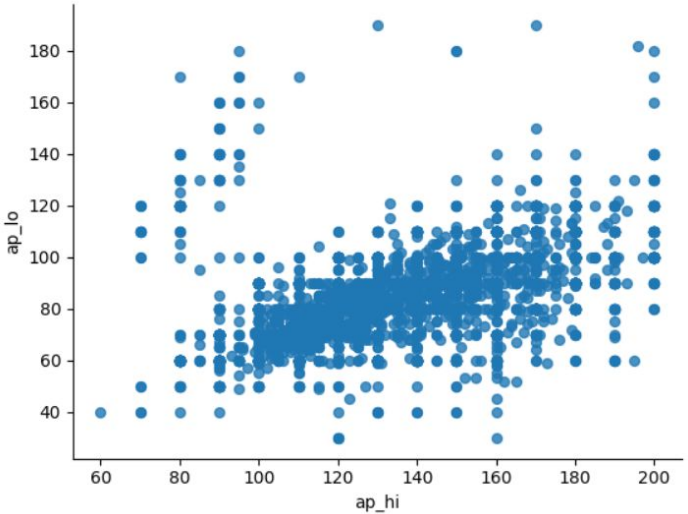
2. Overview of the dataset

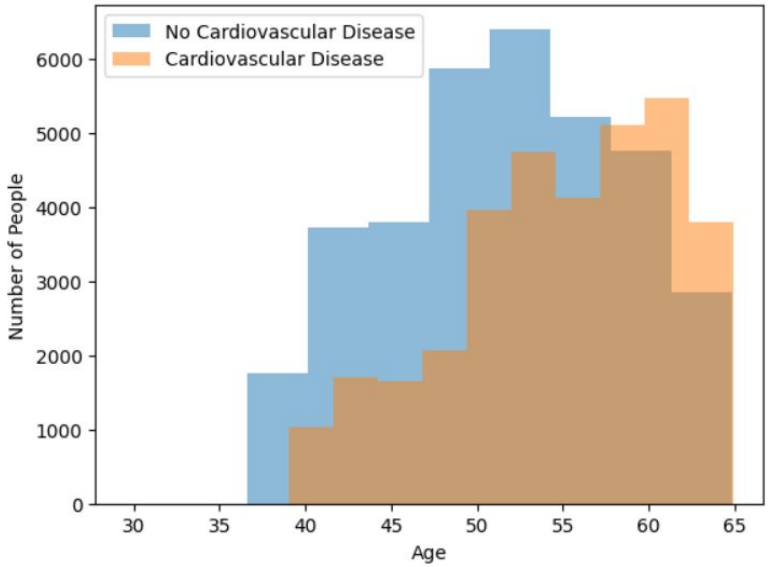
displaying the first 5 rows of the DataFrame

```
[ ]  df.head()
```

	id	age	gender	height	weight	ap_hi	ap_lo	cholesterol	gluc	smoke	alco	active	cardio
0	988	22469	1	155	69.0	130	80	2	2	0	0	1	0
1	989	14648	1	163	71.0	110	70	1	1	0	0	1	1
2	990	21901	1	165	70.0	120	80	1	1	0	0	1	0
3	991	14549	2	165	85.0	120	80	1	1	1	1	1	0
4	992	23393	1	155	62.0	120	80	1	1	0	0	1	0

Insights


ap_lo
ap_hi
180
160
140
120
100
80
60
40
60
80
100
120
140
160
180
200


No Cardiovascular Disease
Cardiovascular Disease
Number of People
Age
6000
5000
4000
3000
2000
1000
0
30
35
40
45
50
55
60
65

2. Data Preprocessing and Analysis

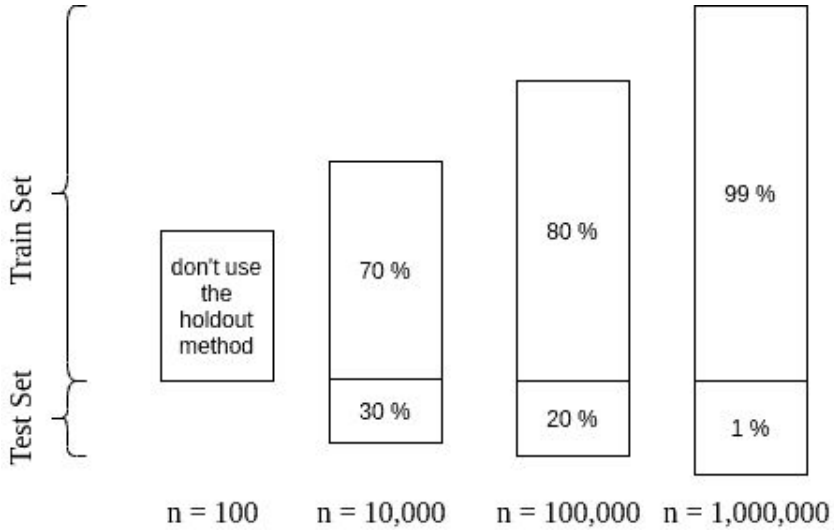
Preprocessing steps of the data:

- Removing the ID variable
- Converting the age from days to years for better readability
- Checking for highly correlated variables: none found
- Combining the height and weight into one single variable
- Deleting outliers

Model Training and Evaluation

Splitting the data into training and test sets.

according to this chart made by Dr. Michal Abin (Software engineering professor at British Columbia Institute of Technology);
The test set should take around 20% since the data is in the order of 100,000



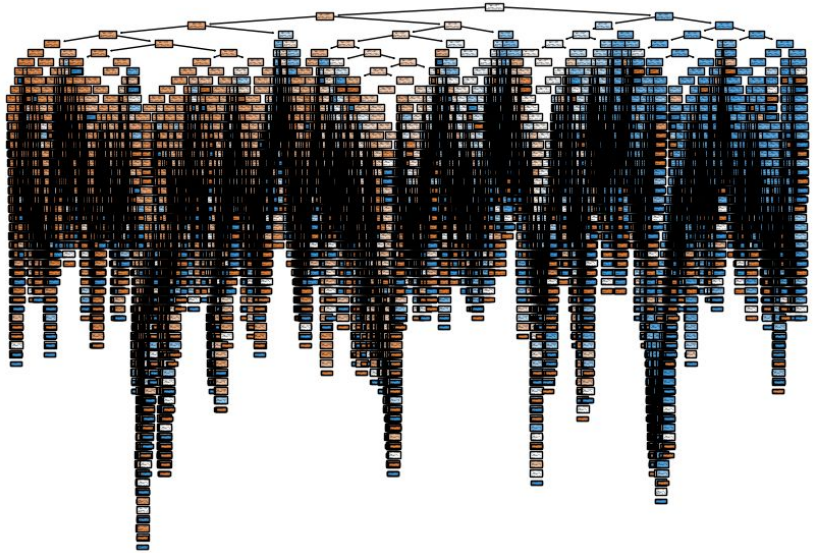
Unpruned tree:

Massive, hard to interpret, and overfit to the training data

Accuracy:

63.37% (on the test set)

99.99% (on the training set)



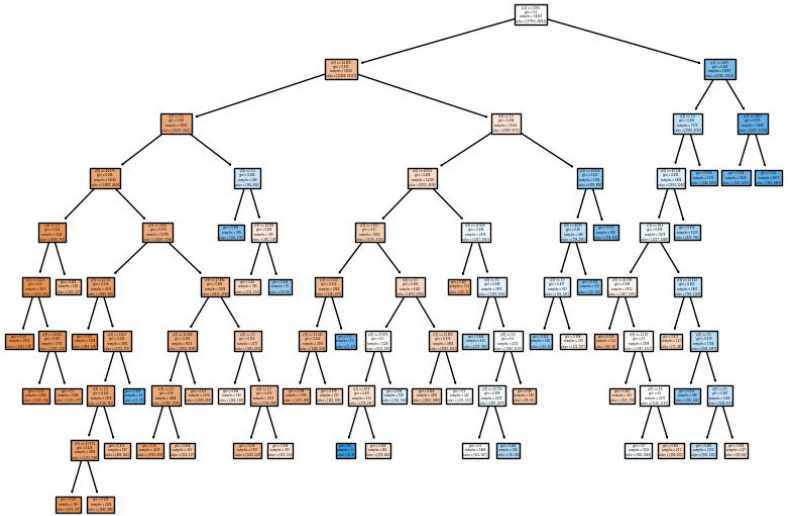
Prepruned tree:

Easier to interpret and more accurate for the test set

Accuracy:

73.57% (on the test set)
73.25% (on the training set)

Thresholds used for optimising the accuracy:
max_depth=10
min_impurity_decrease=0.001



Postpruned tree:

Accuracy:

72.91% (on the test set)

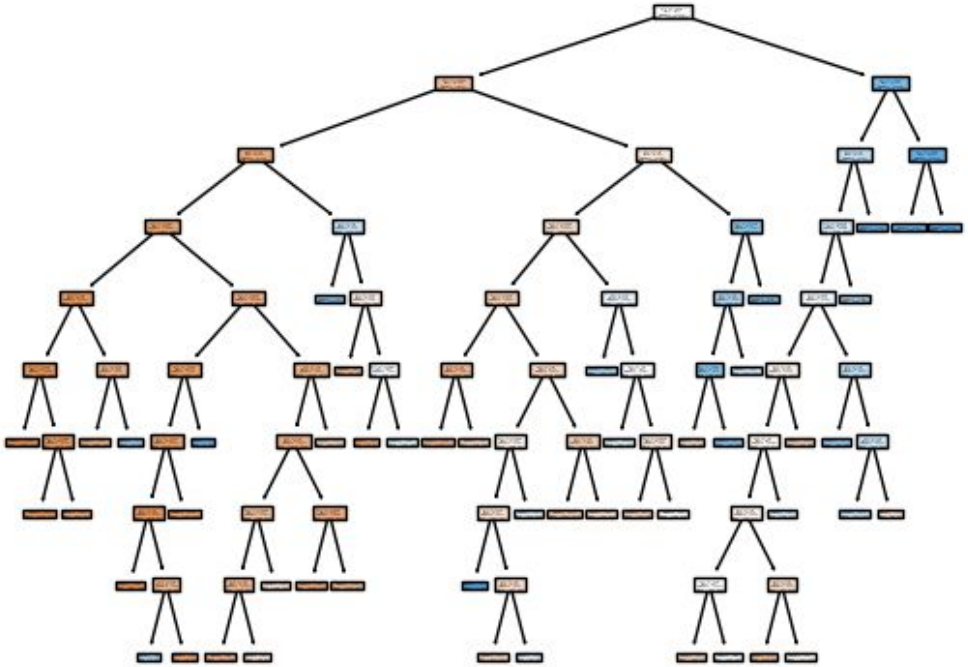
73.23% (on validation set)

73.72% (on the training set)

Number of nodes :

post pruned tree: 97

original tree: 32129



Comparison and Recommendations

the accuracy of the pre pruned tree 73.26% is slightly bigger than the accuracy of the post pruned tree 72.91%.
The original tree overfits to the training set, as the training accuracy(98.46%) is much higher than the test accuracy(63.25%).

We recommend using the post-pruned tree for classifying future data. Although it has a slightly lower accuracy on the training set compared to the pre-pruned tree, its pruning process likely leads to better generalization and less overfitting.

Combining pre-pruning and post-pruning can help create a more robust decision tree model by controlling its size during growth and further refining its structure to improve generalization