

IBM-322

Analytics for Managerial Decision Making

Analysing Heart Attack Possibility Using
Physical And Cardiovascular Details Of
Patients

Group Number : 1
Authors

Vardaan Dua (21115155)

Akshat Chaudhary (21114009)

Mudit Gupta (21114061)

Pise Ashutosh Kalidas (21114073)

Rishi Kejriwal (21114081)

Problem Statement

Cardiovascular diseases (CVDs) represent a significant global health challenge, contributing to a substantial burden of morbidity and mortality. Early detection and accurate risk assessment of individuals prone to cardiovascular diseases are crucial for effective preventive interventions. In this project, we delve into the realm of data-driven healthcare by exploring a comprehensive dataset that encompasses diverse attributes relevant to cardiovascular health.

Our dataset incorporates key features such as **age, sex, gender, height, systolic and diastolic blood pressure readings, cholesterol levels, smoking and drinking habits, and levels of physical activity**. These parameters have been recognized as pivotal indicators in cardiovascular risk assessment and diagnosis.

The objective of our project is to leverage machine learning techniques to develop a predictive model capable of discerning patterns within the dataset and accurately identifying individuals at risk of cardiovascular diseases. By harnessing the power of data analytics, we aim to contribute to the ongoing efforts in the field of preventive healthcare, providing a valuable tool for clinicians and healthcare practitioners to proactively manage and mitigate cardiovascular risks.

This report outlines the analysis, and findings of our investigation into the detection of cardiovascular diseases using the mentioned dataset. We explore various **machine learning models, including logistic regression, multi-layer perceptron, support vector machines, and neural networks**, to evaluate their effectiveness in predicting cardiovascular risks based on the provided features.

Data Analysis

We have chosen the data set from Kaggle Cardio Vascular Disease Dataset.

The dataset contains 70,000 rows and 13 columns .

There are 3 types of input features:

1. Objective: factual information;
2. Examination: results of medical examination;
3. Subjective: information given by the patient.

The 13 columns include patient id , age of patient (Objective Feature) , gender (Objective Feature) , height (Objective Feature) , weight (Objective Feature) , Systolic Blood Pressue (Examination Feature) , Diastolic Blood Pressure (Examination Feature) ,cholesterol (Examination Feature) , glucose (Examination Feature) , smoking and alcohol drinking status (Subjective Feature) , and physical activity (Subjective Feature) . Cardio is the target variable.

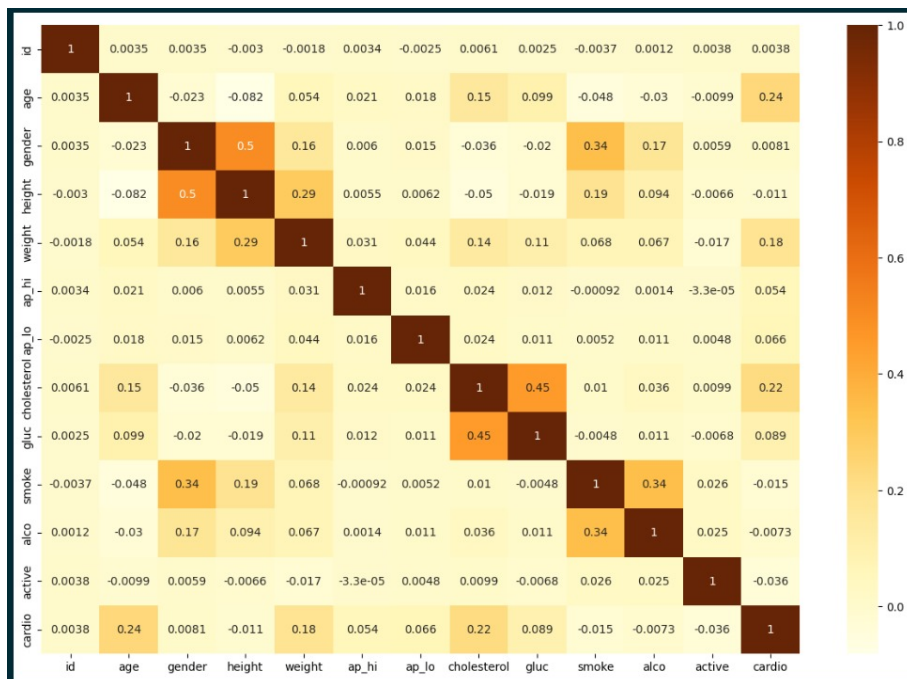
Data size (#Samples, #Features): (70000, 13)													
	id	age	gender	height	weight	ap_hi	ap_lo	cholesterol	gluc	smoke	alco	active	cardio
0	0	18393	2	168	62.0	110	80	1	1	0	0	1	0
1	1	20228	1	156	85.0	140	90	3	1	0	0	1	1
2	2	18857	1	165	64.0	130	70	3	1	0	0	0	1
3	3	17623	2	169	82.0	150	100	1	1	0	0	1	1
4	4	17474	1	156	56.0	100	60	1	1	0	0	0	0

Data Visualization

Correlation Matrix

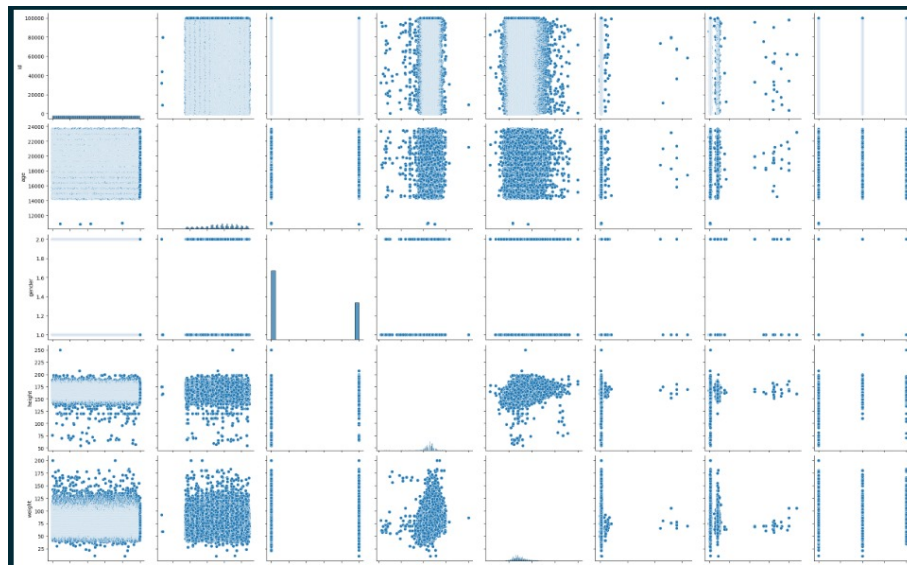
The Correlation Matrix assumes a pivotal role in unraveling intricate relationships among diverse features within the dataset. This analytical tool allows us to quantify and visualize the degree and direction of associations between variables, offering a holistic understanding of their interdependencies. By examining the correlation matrix, we gain insights into which factors exhibit significant correlations and identify potential multicollinearity concerns.

Here a value in i th row and j th column denotes the correlation coefficient value between the i th and j th feature.



Bi - Featurely Plot

The Bi-Featurely Plot emerges as a powerful visual aid, providing a nuanced perspective on the relationships between pairs of features within the dataset. This plotting technique allows us to explore bivariate interactions, enabling the identification of potential patterns, trends, and outliers.



Logistic Regression

We have used Logistic Regression in majorly two ways that are using the threshold of 0.5 for class prediction but since heart diseases are a crucial issue we have also reduced the threshold to **0.3 for reducing cases where a person had a disease but couldn't be identified.**

Results for Logistic Regression with threshold 0.5

Here the model classifies a data point to be of type 1 that is having a disease if the probability of having a disease is greater than 0.5 . We get 0.7235 accuracy. Explanation of Confusion Matrix is as follows :

The number of people who didn't have the disease and were classified correctly are : 5361

The number of people who didn't have the disease and couldn't be classified correctly are : 1627

The number of people who had the disease but couldn't be classified correctly are : 2244

The number of people who had the disease and are classified correctly : 4768

```
Accuracy: 72.35%

Confusion Matrix:
[[5361 1627]
 [2244 4768]]

Classification Report:
              precision    recall  f1-score   support

     0       0.70      0.77      0.73       6988
     1       0.75      0.68      0.71       7012

   accuracy          0.72      0.72      0.72      14000
  macro avg          0.73      0.72      0.72      14000
 weighted avg          0.73      0.72      0.72      14000
```

Results for Logistic Regression with threshold 0.3

Here the model classifies a data point to be of type 1 that is having a disease if the probability of having a disease is greater than 0.3 . We get 0.6260 accuracy.

Explanation of Confusion Matrix

The number of people who didn't have the disease and were classified correctly are : 2176

The number of people who didn't have the disease and couldn't be classified correctly are : 4812

The number of people who had the disease but couldn't be classified correctly are : 424

The number of people who had the disease and are classified correctly : 6588

```
Accuracy: 62.60%

Confusion Matrix:
[[2176 4812]
 [ 424 6588]]

Classification Report:
```

	precision	recall	f1-score	support
0	0.84	0.31	0.45	6988
1	0.58	0.94	0.72	7012
accuracy			0.63	14000
macro avg	0.71	0.63	0.58	14000
weighted avg	0.71	0.63	0.58	14000

Multilayer Perceptron

The relation between the given features and Cardiac Disease possibility may or may not be linear. Hence we have tried Multi Layer Perceptron which excels in capturing complex, non-linear relationships within the dataset, allowing for a more nuanced understanding of the interplay between diverse cardiovascular risk factors. Its ability to learn intricate patterns makes it a valuable asset in our predictive modeling, striving to uncover hidden insights and enhance the accuracy of our cardiovascular risk predictions. We have used multiplayer perceptron model with 100 neurons in each hidden layer along with 50 hidden layers . We have got an accuracy of 0.7221.

Explanation of Confusion Matrix

The number of people who didn't have the disease and were classified correctly are : 5345

The number of people who didn't have the disease and couldn't be classified correctly are : 1643

The number of people who had the disease but couldn't be classified correctly are : 2248

The number of people who had the disease and are classified correctly : 4764

Accuracy: 72.21%

Confusion Matrix:

```
[[5345 1643]
 [2248 4764]]
```

Classification Report:

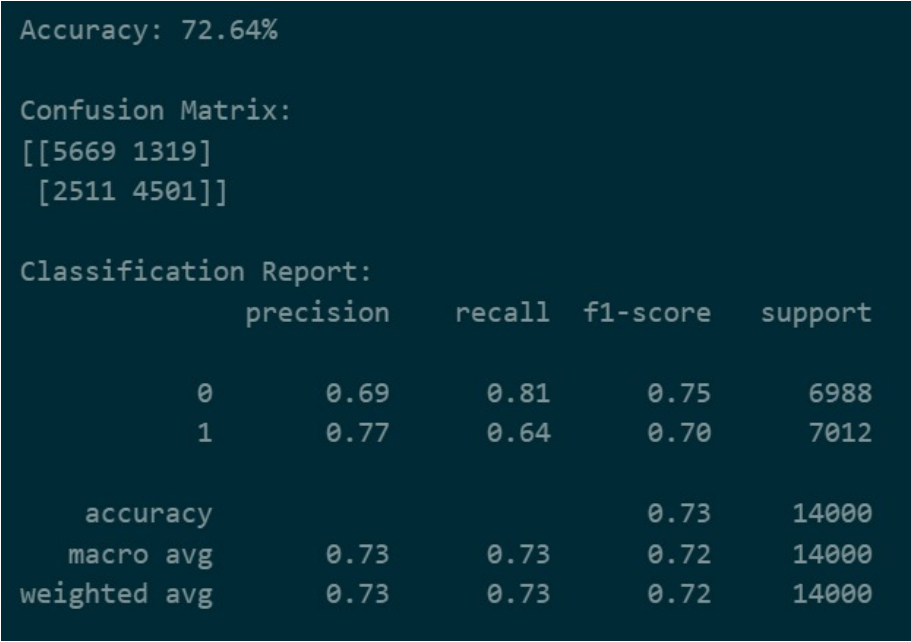
	precision	recall	f1-score	support
0	0.70	0.76	0.73	6988
1	0.74	0.68	0.71	7012
accuracy			0.72	14000
macro avg	0.72	0.72	0.72	14000
weighted avg	0.72	0.72	0.72	14000

Support Vector Machine

SVM is a supervised Machine Learning algorithm that can be used for classification or regression tasks . The basic idea behind SVM is to find a hyperplane that best separates data points of different classes in a high dimensional space. In SVM's kernel plays a crucial role in allowing the algorithm to handle non linear relationships between features. We have used two kernels namely linear and rbf (with degree 3).

Results for Support Vector Machine with linear kernel

We have got an accuracy of 0.7264.

A screenshot of a terminal window with a dark blue background and white text. It displays the results of an SVM model with a linear kernel. The output includes the accuracy (72.64%), a confusion matrix, and a classification report. The confusion matrix shows 5669 true positives and 1319 false positives in the first row, and 2511 false negatives and 4501 true negatives in the second row. The classification report provides precision, recall, f1-score, and support for each class, as well as overall accuracy and average metrics.

```
Accuracy: 72.64%

Confusion Matrix:
[[5669 1319]
 [2511 4501]]

Classification Report:
              precision    recall  f1-score   support

     0       0.69         0.81         0.75         6988
     1       0.77         0.64         0.70         7012

 accuracy          0.73
 macro avg         0.73         0.73         0.72         14000
weighted avg         0.73         0.73         0.72         14000
```

The explanation for confusion matrix is as follows :

The number of people who didn't have the disease and were classified correctly are : 5669

The number of people who didn't have the disease and couldn't be classified correctly are : 1319

The number of people who had the disease but couldn't be classified correctly are : 2511

The number of people who had the disease and are classified correctly : 4501

Results for Support Vector Machine with rbf kernel

We have got an accuracy of 0.7319.

```
Accuracy: 73.19%

Confusion Matrix:
[[5321 1667]
 [2086 4926]]

Classification Report:

```

	precision	recall	f1-score	support
0	0.72	0.76	0.74	6988
1	0.75	0.70	0.72	7012
accuracy			0.73	14000
macro avg	0.73	0.73	0.73	14000
weighted avg	0.73	0.73	0.73	14000

The explanation for confusion matrix is as follows :

The number of people who didn't have the disease and were classified correctly are : 5321

The number of people who didn't have the disease and couldn't be classified correctly are : 1667

The number of people who had the disease but couldn't be classified correctly are : 2086

The number of people who had the disease and are classified correctly : 4926

Neural Network

In Multilayer perceptron we don't have the freedom to tweak the neural network inside but here we have full control on the activation function for each layer that is each layer can have a different activation function and hence we can customize our neural network to make the best predictions given the dataset. For such type of fine grained control over the layers we have used the **tensorflow** library. The model is compiled using **Adam Optimiser** which is a popular optimisation algorithm ,it optimises Stochastic Gradient Descent by dynamically adjusting learning rates based on individual weights.

Building the neural network

```
# Build the neural network model
nn = tf.keras.Sequential([
    tf.keras.layers.Dense(64, activation='relu', input_shape=(X_train.shape[1],)),
    tf.keras.layers.Dense(32, activation='relu'),
    tf.keras.layers.Dense(1, activation='sigmoid')
])

# Compile the model
nn.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
```

The first layer contains 64 nodes and uses ReLu activation function, second layer contains 32 nodes and uses ReLu activation function, the third layer contains 1 node and uses sigmoid activation function.

Prediction Results for Neural Network

We have got an accuracy 0.7371 . The explanation for confusion matrix is as follows :

The number of people who didn't have the disease and were classified correctly are : 5520

The number of people who didn't have the disease and couldn't be classified correctly are : 1468

The number of people who had the disease but couldn't be classified correctly are : 2213

The number of people who had the disease and are classified correctly : 4799

Accuracy: 73.71%

Confusion Matrix:

```
[[5520 1468]
 [2213 4799]]
```

Classification Report:

	precision	recall	f1-score	support
0	0.71	0.79	0.75	6988
1	0.77	0.68	0.72	7012
accuracy			0.74	14000
macro avg	0.74	0.74	0.74	14000
weighted avg	0.74	0.74	0.74	14000