Heart disease prediction using machine learning

Rahma ELHusseiny
Arab Academy for Science and Technology, Egypt, r.e.abdelsamia@student.aast.edu

Supervisor: Faculty Supervisor Muhammed Salah , Teacher Assistant
Arab Academy for Science and Technology, Egypt, mohamedsalah95@aast.edu

Abstract *– Heart diseases have increased enormously in the modern world. People now are facing inducements (e.g. Fast Food) that can have a direct effect on The Heart. All doctors deal with those cases as they do their best to save their lives. To do this, they must have the right results and to help them in diagnosis. Using a model that can predict the vulnerable situation of The heart. Given the basic symptoms such as age, gender, fasting blood sugar, resting blood pressure, person's cholesterol, chest pain experience, person's maximum heart rate, exercise-induced angina, ST depression, The slope of the peak exercise ST segment, The number of major vessels and Thalassemia.*

This can help doctors to recheck their results. The dataset that's been used for this analysis is Framingham" obtained from Kaggle and Heart disease dataset with 14 features is obtained from UCI. This paper presents the analysis implemented on different models like Decision trees, Random Forest, and K- Nearest Neighbors. After comparing the analysis of these models with each other it has proven that the Random Forest model has the highest accuracy. With an accuracy of 90.16% means it's proven that it's the most accurate and trustworthy.

I. Introduction

For the past decade, heart disease or coronary heart disease has remained the leading cause of death worldwide. The World Health Organization estimates that more than 17.9 million people worldwide die each year from heart disease, and 80% of these deaths are from coronary heart disease and stroke (1]. High mortality rates are common in low- and middle-income countries [2]. „ as well as existing cardiovascular conditions, are factors that cause heart disease. Effective and accurate diagnosis and early treatment of heart disease play a very important role in taking preventative measures.

Data mining refers to the extraction of the required information from large data sets in various fields such as the medical sector, business sector, and the field of education. Machine learning is one of the fastest-growing areas of artificial intelligence. These algorithms can analyze large amounts of data from a variety of fields, one of the most important fields being the medical field. It replaces the method of modeling a common guess using a computer to gain an understanding of complex and offline communication between various factors by minimizing errors in predicted and accurate results [3]. Data mining explores large data sets to extract critical decision-making data from past collections for future analysis. The medical field contains extensive patient information. This data needs to be mined by various machine learning algorithms. Healthcare professionals analyze this data to achieve an effective diagnostic decision by health professionals. Mining medical data using classification algorithms provides clinical assistance with analysis. It examines the classification algorithms to predict heart disease in patient

Methodology

The methodology for predicting cardiovascular disease was done using those algorithms.

1. *Decision Tree*
2. *Random Forest*
3. *K Nearest Neighbour*

A. Decision Tree

Introduction

A decision tree is a flowchart-like structure in which each internal node represents a "test" on an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes).

Dataset

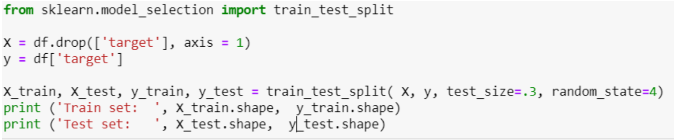
The dataset which was used for analysis are "Framingham"
obtained from Kaggle. Heart disease
dataset with 14 features is obtained from UCI Machine Learning Repository
It contains numeric and categorical values.

Implementation

A. *We load the dataset we have*
B. *Split it into Train and Test datasets*
C. *K fold cross validation is done on the Train dataset (k = 5)*
D. *Make predictions after training the model*
E. *Calculate the accuracy*

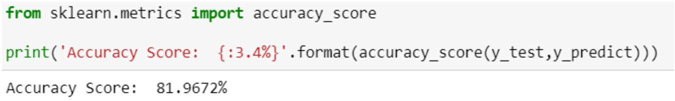
*Accuracy = ((TP + TN) / (TP + TN + FN + FB)) * 100*

TP- True Positive (prediction is yes, and they do have the disease.
TN-True Negative (prediction is no, and they don't have the disease.)
FP-False Positive (We predicted yes, but they don't actually have the disease. (Also known as a
"Type I error.")
FN-False Negative (We predicted no, but they actually do have the disease. (Also known as a
"Type II error.")
The Accuracy = 81.96%



```
from sklearn.model_selection import train_test_split

X = df.drop(['target'], axis = 1)
y = df['target']

X_train, X_test, y_train, y_test = train_test_split( X, y, test_size=.3, random_state=4)
print ('Train set:  ', X_train.shape,  y_train.shape)
print ('Test set:   ', X_test.shape,  y_test.shape)
```

Fig1 Sample code of splitting the dataset



```
from sklearn.metrics import accuracy_score

print('Accuracy Score:  {:3.4%}'.format(accuracy_score(y_test,y_predict)))
```

```
Accuracy Score:  81.9672%
```

Fig 2 accuracy of DT algorithm.

A. Random Forest

Introduction

Random forests *or* ***random decision forests*** *are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of* ***decision*** *trees at training time.*

Dataset

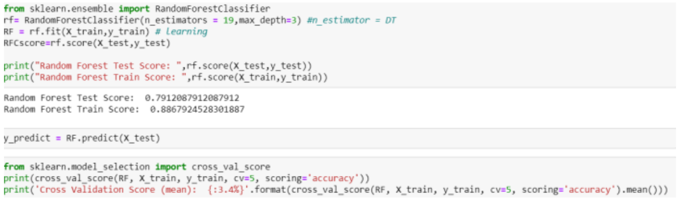
The dataset which was used for analysis are "Framingham"
obtained from Kaggle. Heart disease
dataset with 14 features is obtained from UCI Machine Learning Repository
It contains numeric and categorical values.

Implementation

A.We load the dataset we have
B.Split it into Train and Test datasets
C.K fold cross validation is done on the Train dataset (k = 5)
D. Make predictions after training the model
A. Calculate the accuracy
*Accuracy = ((TP + TN) / (TP + TN + FN + FB)) * 100*

TP- True Positive (prediction is yes, and they do have the disease.
TN-True Negative (prediction is no, and they don't have the disease.)
FP-False Positive (We predicted yes, but they don't actually have the disease. (Also known as a
"Type I error.")
FN-False Negative (We predicted no, but they actually do have the disease. (Also known as a
"Type II error.")

The Accuracy = 90.61 %



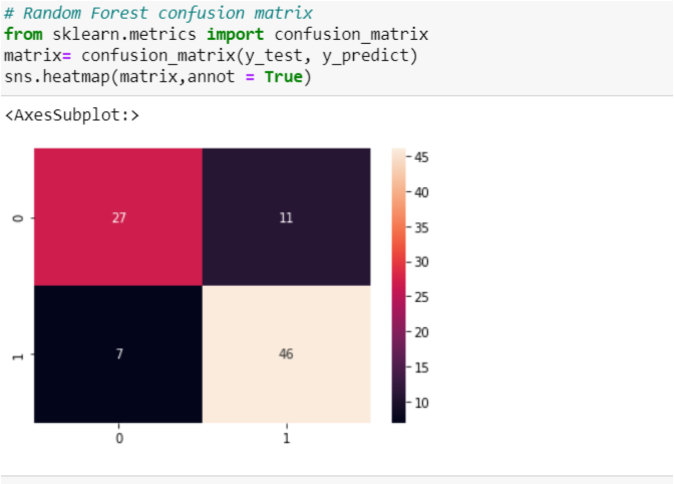
```
from sklearn.ensemble import RandomForestClassifier
rf= RandomForestClassifier(n_estimators = 19,max_depth=3) #n_estimator = DT
RF = rf.fit(X_train,y_train) # Learning
RFCscore=rf.score(X_test,y_test)

print("Random Forest Test Score: ",rf.score(X_test,y_test))
print("Random Forest Train Score: ",rf.score(X_train,y_train))
```

```
Random Forest Test Score:  0.7912087912087912
Random Forest Train Score:  0.8867924528301887
```

```
y_predict = RF.predict(X_test)
```

```
from sklearn.model_selection import cross_val_score
print(cross_val_score(RF, X_train, y_train, cv=5, scoring='accuracy'))
print('Cross Validation Score (mean):  {:3.4%}'.format(cross_val_score(RF, X_train, y_train, cv=5, scoring='accuracy').mean()))
```

Fig 3 Random forest and cross validation algorithm



```
# Random Forest confusion matrix
from sklearn.metrics import confusion_matrix
matrix= confusion_matrix(y_test, y_predict)
sns.heatmap(matrix,annot = True)
```

```
<AxesSubplot:>
```

Fig 4 Random forest confusion matrix

A. K Nearest Neighbor

Introduction

- *K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.*
- *K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.*
- *K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.*
- *K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.*

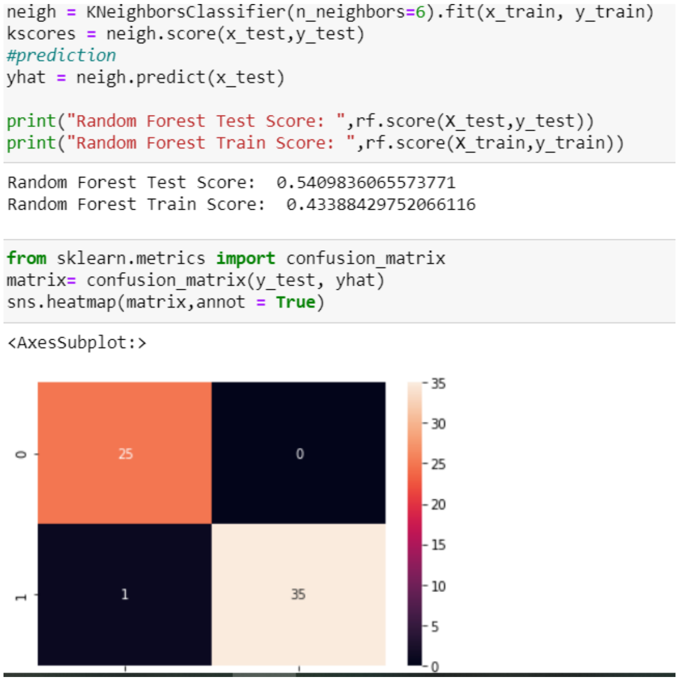
Dataset

The dataset which was used for analysis are "Framingham"
obtained from Kaggle. Heart disease
dataset with 14 features is obtained from UCI Machine Learning Repository
It contains numeric and categorical values.

Implementation

A. We load the dataset we have
B.Split it into Train and Test datasets
C.We Choose number of k neighbors to be used in classification
D. K fold cross validation is done on the Train dataset (k = 5)
E.Make predictions after training the model
F. Calculate the accuracy
*Accuracy = ((TP + TN) / (TP + TN + FN + FB)) * 100*

TP- True Positive (prediction is yes, and they do have the disease.
TN-True Negative (prediction is no, and they don't have the disease.)
FP-False Positive (We predicted yes, but they don't actually have the disease. (Also known as a
"Type I error.")
FN-False Negative (We predicted no, but they actually do have the disease. (Also known as a
"Type II error.")
The accuracy = 88.52 %



```
neigh = KNeighborsClassifier(n_neighbors=6).fit(x_train, y_train)
kscores = neigh.score(x_test,y_test)
#prediction
yhat = neigh.predict(x_test)

print("Random Forest Test Score: ",rf.score(X_test,y_test))
print("Random Forest Train Score: ",rf.score(X_train,y_train))
```

```
Random Forest Test Score:  0.5409836065573771
Random Forest Train Score:  0.43388429752066116
```

```
from sklearn.metrics import confusion_matrix
matrix= confusion_matrix(y_test, yhat)
sns.heatmap(matrix,annot = True)
```

```
<AxesSubplot:>
```

Fig 5 KNN algorithm and confusion matrix

Conclusion

Using ML in these kinds of diseases will help doctors discover the disease in early stages and it can make them able to save more lives. They can guide people to a healthy life and help them to avoid that kind of disease for a long time.

References

[1] *https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/#:~:text=Introduction%20The%20abbreviation%20KNN%20stands,to%20a%20new%20unknown%20variable…*

[2] *https://www.mentalhelp.net/heart-disease/*

[3] *https://link.springer.com/article/10.1007/s42979-020-00365-y*