

**Vidyavardhini's College of Engineering & Technology**

**Report On**

**HEART DISEASE PREDICTION**

**Name:** Abhishek Walinjkar

**Mail**: [abhiwa2000@gmail.com](mailto:abhiwa2000@gmail.com)

**Institute**: Vidyavardhini’s College of Engineering and Technology

**Trainer**

Sachin Lohar

Knowledge Solutions India

**INDEX**

|  |  |
| --- | --- |
| **Sr. No** | **Contents** |
| **1** | **Abstract** |
| **2** | **Software Libraries Used** |
| **3** | **Graphs/Figures** |
| **4** | **Code** |
| **5** | **Observations** |
| **6** | **Conclusion** |

**ABSTRACT**

Heart disease is one of the biggest causes of morbidity and mortality among the population of the world. Prediction of cardiovascular disease is regarded as one of the most important subjects in the section of clinical data science. The amount of data in the healthcare industry is huge.

Data Science may help doctors better predict the risk of patients developing heart diseases by ensuring accuracy, consistency and objectivity.The model will basically collect various information from a possible case of patient and help us determine if there is a possibility of heart disease or not.

Heart-disease prediction is carried out using decision tree classifier and KNN method of machine learning.

**INTRODUCTION**

It is a well-known fact that Heart Diseases are currently the leading cause of death across the world. The development of a computational system that can predict the presence of heart diseases will significantly reduce the mortality rates & substantially reduce the costs of health care. Such information predicted well in advance, can provide important insights to doctors.Early prediction of heart disease in a patient will help a lot as the treatment can be started as early as possible hence reducing the rate of fatal cases.

In this project, we will be applying Machine Learning techniques to classify whether a person is suffering from heart disease or not.We are using a dataset which consists of all the information about a patient that might help predict if there is a chance of heart disease or not.Attributes like age,sex,chest pain,resting blood pressure,fasting blood pressure, etc were taking into consideration.

**SOFTWARE LIBRARIES USED**

Main Programming Language: Python

Application used: Jupyter Notebook (Anaconda3)

Libraries used in Python:

* Sklearn
* Numpy
* Pandas
* Matplotlib
* Plotly
* Plotly.express
* Cufflinks
* Seaborn
* Os

Modules Used:

* Plotly.express
* Plotly.graph\_objs
* Matplotlib.pyplot
* Sklearn.metrics
* Plotly.offline

**GRAPHS/FIGURES**

|  |  |  |
| --- | --- | --- |
| Graph/Figure | Library Used for graph/figure | Visualization |
| histogram | matplotlib |  |
| bar | matplotlib |  |
| distplot | seaborn |  |
| pairplot | seaborn |  |
| countplot | seaborn |  |
| heatmap | seaborn |  |
| violinplot | seaborn | (Figure on the left) |
| swarmplot | seaborn | (Figure on the right) |
| pointplot | seaborn | (Left-most figure) |
| plot | matplotlib |  |

**CODE**

import sklearn

import numpy as np

import pandas as pd

import plotly as plot

import plotly.express as px

import plotly.graph\_objs as go

import cufflinks as cf

import matplotlib.pyplot as plt

import seaborn as sns

import os

from sklearn.metrics import accuracy\_score

import plotly.offline as pyo

from plotly.offline import init\_notebook\_mode,plot,iplot

pyo.init\_notebook\_mode(connected=True)

cf.go\_offline()

heart=pd.read\_csv(r‘heart.csv’)

heart

info = ["age","1: male, 0: female","chest pain type, 1: typical angina, 2: atypical angina, 3: non-anginal pain, 4: asymptomatic","resting blood pressure"," serum cholesterol in mg/dl","fasting blood sugar > 120 mg/dl","resting electrocardiographic results (values 0,1,2)"," maximum heart rate achieved","exercise induced angina","oldpeak = ST depression induced by exercise relative to rest","the slope of the peak exercise ST segment","number of major vessels (0-3) colored by fluoroscopy","thal: 3 = normal; 6 = fixed defect; 7 = reversible defect"]

for i in range(len(info)):

print(heart.columns[i]+":\t\t\t"+info[i])

heart['target']

heart.groupby('target').size()

heart.groupby('target').sum()

heart.shape

heart.size

heart.describe()

heart.info()

heart['target'].unique()

heart.hist(figsize=(14,14))

plt.show()

plt.bar(x=heart['sex'],height=heart['age'])

plt.show()

sns.barplot(x="fbs", y="target", data=heart)

plt.show()

sns.barplot(heart["cp"],heart['target'])

sns.barplot(heart["sex"],heart['target'])

px.bar(heart,heart['sex'],heart['target'])

sns.distplot(heart["thal"])

sns.distplot(heart["chol"])

sns.pairplot(heart,hue='target')

numeric\_columns=['trestbps','chol','thalach','age','oldpeak']

heart['target']

y = heart["target"]

sns.countplot(y)

target\_temp = heart.target.value\_counts()

print(target\_temp)

# create a correlation heatmap

sns.heatmap(heart[numeric\_columns].corr(),annot=True, cmap='terrain', linewidths=0.1)

fig=plt.gcf()

fig.set\_size\_inches(8,6)

plt.show()

# create four distplots

plt.figure(figsize=(12,10))

plt.subplot(221)

sns.distplot(heart[heart['target']==0].age)

plt.title('Age of patients without heart disease')

plt.subplot(222)

sns.distplot(heart[heart['target']==1].age)

plt.title('Age of patients with heart disease')

plt.subplot(223)

sns.distplot(heart[heart['target']==0].thalach )

plt.title('Max heart rate of patients without heart disease')

plt.subplot(224)

sns.distplot(heart[heart['target']==1].thalach )

plt.title('Max heart rate of patients with heart disease')

plt.show()

plt.figure(figsize=(13,6))

plt.subplot(121)

sns.violinplot(x="target", y="thalach", data=heart, inner=None)

sns.swarmplot(x="target", y="thalach", data=heart, color='w', alpha=0.5)

plt.subplot(122)

sns.swarmplot(x="target", y="thalach", data=heart)

plt.show()

# create pairplot and two barplots

plt.figure(figsize=(16,6))

plt.subplot(131)

sns.pointplot(x="sex", y="target", hue='cp', data=heart)

plt.legend(['male = 1', 'female = 0'])

plt.subplot(132)

sns.barplot(x="exang", y="target", data=heart)

plt.legend(['yes = 1', 'no = 0'])

plt.subplot(133)

sns.countplot(x="slope", hue='target', data=heart)

plt.show()

heart['target'].value\_counts()

heart['target'].isnull()

heart['target'].sum()

heart['target'].unique()

heart.isnull().sum()

X,y=heart.loc[:,:'thal'],heart.loc[:,'target']

X

X.shape

y.shape

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

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

X

X\_train,X\_test,y\_train,y\_test=train\_test\_split(X,y,random\_state=10,test\_size=0.3,shuffle=True)

X\_test

y\_test

print ("train\_set\_x shape: " + str(X\_train.shape))

print ("train\_set\_y shape: " + str(y\_train.shape))

print ("test\_set\_x shape: " + str(X\_test.shape))

print ("test\_set\_y shape: " + str(y\_test.shape))

Category=['No....but i pray you don't get Heart Disease or at least CoronaVirus Soon...','Yes you have Heart Disease....RIP in Advance']

from sklearn.tree import DecisionTreeClassifier

dt=DecisionTreeClassifier()

dt.fit(X\_train,y\_train)

prediction=dt.predict(X\_test)

accuracy\_dt=accuracy\_score(y\_test,prediction)\*100

accuracy\_dt

print("Accuracy on training set: {:.3f}".format(dt.score(X\_train, y\_train)))

print("Accuracy on test set: {:.3f}".format(dt.score(X\_test, y\_test)))

X\_DT=np.array([[63 ,1, 3,145,233,1,0,150,0,2.3,0,0,1]])

X\_DT\_prediction=dt.predict(X\_DT)

X\_DT\_prediction[0]

print(Category[int(X\_DT\_prediction[0])])

print("Feature importances:\n{}".format(dt.feature\_importances\_))

def plot\_feature\_importances\_diabetes(model):

plt.figure(figsize=(8,6))

n\_features = 13

plt.barh(range(n\_features), model.feature\_importances\_, align='center')

plt.yticks(np.arange(n\_features), X)

plt.xlabel("Feature importance")

plt.ylabel("Feature")

plt.ylim(-1, n\_features)

plot\_feature\_importances\_diabetes(dt)

plt.savefig('feature\_importance')

sc=StandardScaler().fit(X\_train)

X\_train\_std=sc.transform(X\_train)

X\_test\_std=sc.transform(X\_test)

X\_test\_std

from sklearn.neighbors import KNeighborsClassifier

knn=KNeighborsClassifier(n\_neighbors=4)

knn.fit(X\_train\_std,y\_train)

prediction\_knn=knn.predict(X\_test\_std)

accuracy\_knn=accuracy\_score(y\_test,prediction\_knn)\*100

print("Accuracy on training set: {:.3f}".format(knn.score(X\_train, y\_train)))

print("Accuracy on test set: {:.3f}".format(knn.score(X\_test, y\_test)))

k\_range=range(1,26)

scores={}

scores\_list=[]

for k in k\_range:

knn=KNeighborsClassifier(n\_neighbors=k)

knn.fit(X\_train\_std,y\_train)

prediction\_knn=knn.predict(X\_test\_std)

scores[k]=accuracy\_score(y\_test,prediction\_knn)

scores\_list.append(accuracy\_score(y\_test,prediction\_knn))

scores

plt.plot(k\_range,scores\_list)

px.line(x=k\_range,y=scores\_list)

X\_knn=np.array([[63 ,1, 3,145,233,1,0,150,0,2.3,0,0,1]])

X\_knn\_std=sc.transform(X\_knn)

X\_knn\_prediction=dt.predict(X\_knn)

X\_knn\_std

(X\_knn\_prediction[0])

print(Category[int(X\_knn\_prediction[0])])

algorithms=['Decision Tree','KNN']

scores=[accuracy\_dt,accuracy\_knn]

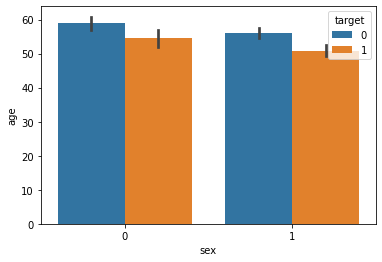
sns.set(rc={'figure.figsize':(15,7)})

plt.xlabel("Algorithms")

plt.ylabel("Accuracy score")

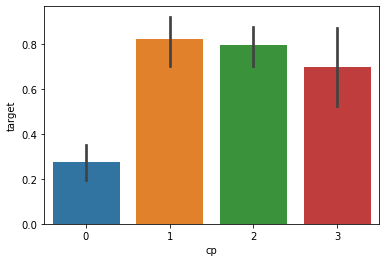
sns.barplot(algorithms,scores)

**OBSERVATIONS**

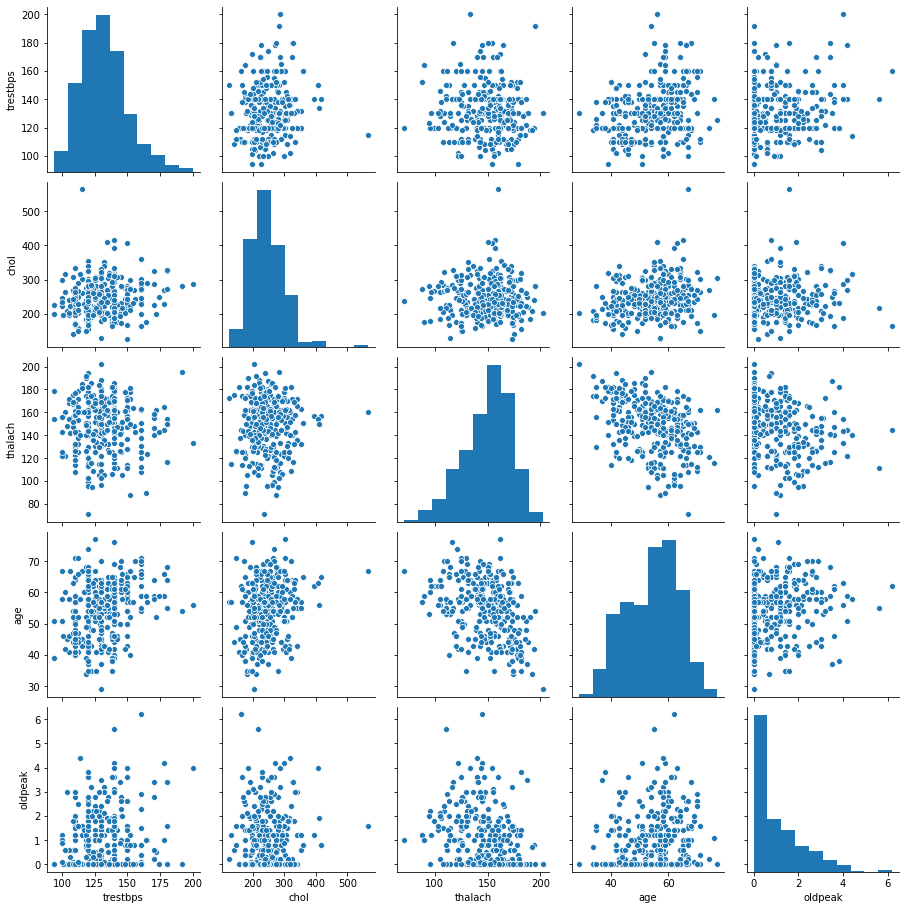
****

In the above barplot, we have the attributes ‘sex’ and ‘age’ on the x-axis and y-axis respectively. The value 1 is for males and the value 0 is for females. In the ‘target’ attribute, value 1 means there is heart disease and if 0, it means no heart disease.

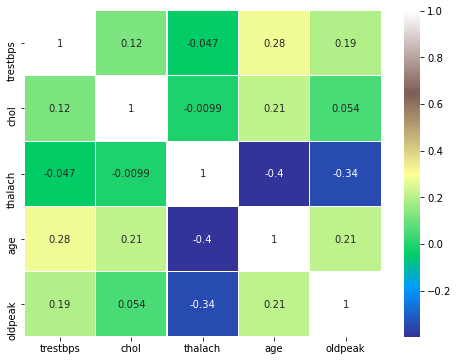
We find out that females who are suffering from the disease are older than males.



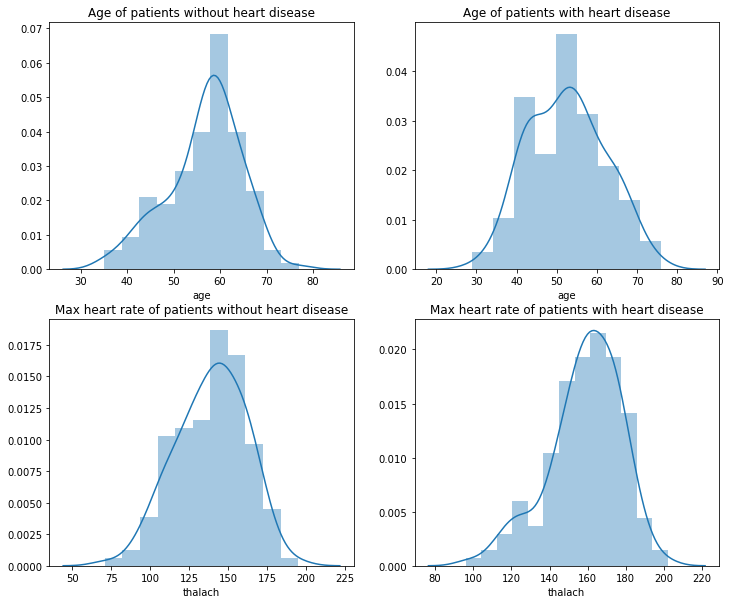
The barplot having ‘cp’(chest pain) on the x-axis and ‘target’ on the y-axis shows that among different types of chest pains, where 0: typical angina, 1: atypical angina, 2: non-anginal pain, 3: asymptomatic respectively. The chest pain type 1 which is atypical results in a maximum rate of having a heart disease.



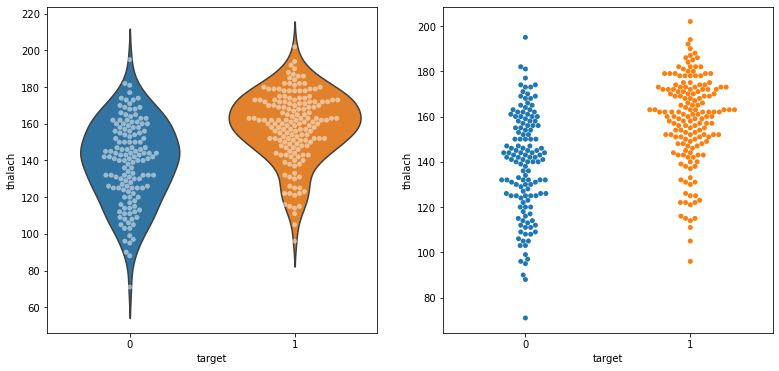
In the above pairplot, we cannot find any clear positive or negative correlation between the attributes trestbps, chol, thalach, age and oldpeak.



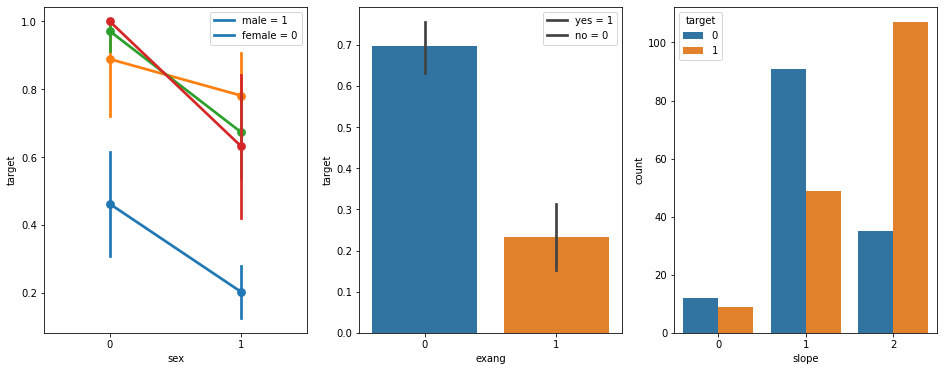
In the above heatmap, we find out that there is no strong correlation between any of the attributes which are taken into consideration.



From the above distplots, we find out that the age of patients suffering from heart diseases is lesser than that of patients without heart disease.Also, the max heart rate of patients with heart disease is greater.



The above violin plot shows us that patients with heart disease have greater thalach values i.e maximum heart rate. The swarm plot gives us the same observation as with attributes thalach and target.



In the pointplot, we can observe that the females with chest pains are more likely to get heart diseases than males. If the exang value i.e exercise induced angina is greater, then the chances of getting heart disease is less.

For development of models we divide our data into x and y parts using sklearn library.The shapes of train data for x and y and test data data for x and y is:

train\_set\_x shape: (212, 13)

train\_set\_y shape: (212,)

test\_set\_x shape: (91, 13)

test\_set\_y shape: (91,)

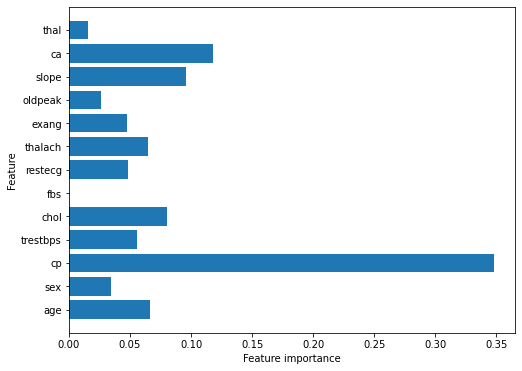
Now, using the decision tree classifier algorithm we find out the accuracy on training dataset and test dataset.The accuracy for decision tree after calculating is 78.02197802197803.We find out the accuracy for the test and training datasets next.

We get,

Accuracy on training set: 1.000

Accuracy on test set: 0.780

By taking a single row of data into consideration and using the predict method, we get the output if the patient has heart disease or not.

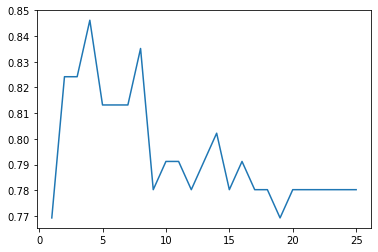


From the above horizontal bar plot, we can clearly see what features have the most impact on having heart disease. Chest pains can be seen as the major factor for a patient for having a heart disease from the above plot.

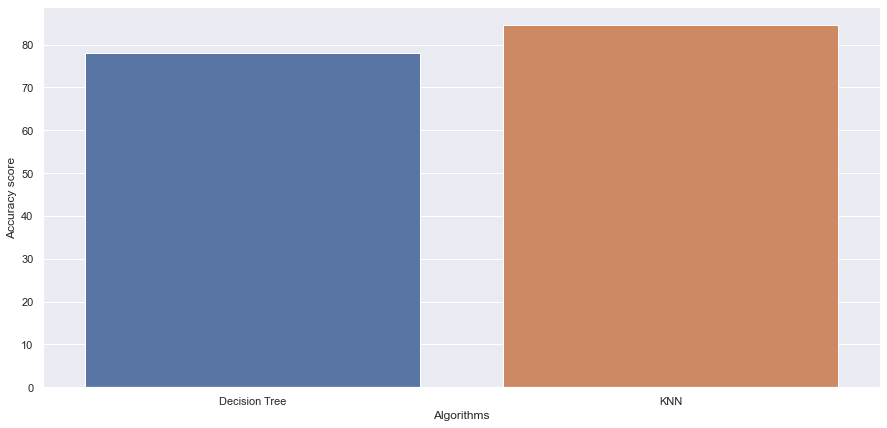
For the K-Nearest Neighbors algorithm, we set the neighbors value to 4. The accuracy for KNN after calculating is 84. 61538461538461.We find out the accuracy for the test and training datasets next. We get,

Accuracy on training set: 0.373

Accuracy on test set: 0.516



From the above plot we can see the distribution of nearest neighbors and the accuracy scores from the KNN algorithm.



On comparing both the algorithms used i.e. KNN and Decision Tree we get the result that KNN is more accurate than decision tree algorithm. The KNN provides an accuracy of 84.61538461538461 while the accuracy of decision tree was 78.02197802197803 respectively.

**CONCLUSION**

The early prediction of cardiovascular diseases can aid in making decisions on lifestyle changes in high-risk patients and in turn reduce the complications, which can be a great milestone in the field of medicine. Further for its enhancement, we can train on models and predict the types of cardiovascular diseases providing recommendations to the users, and also use more enhanced models.