SOURCE CODE:

import packages

import pandas as pd

import numpy as np

from matplotlib import pyplot as plt

%matplotlib inline

import seaborn as sns

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

from sklearn.naive\_bayes import MultinomialNB

from sklearn.linear\_model import LogisticRegression

from sklearn.ensemble import RandomForestClassifier

from sklearn import tree

from sklearn.svm import SVC

import pickle

Data Analysys

data1=pd.read\_csv("/content/archive (6) (1).zip")

data1.head()

data1.shape

data1.info()

data1.isnull().sum()

data1.describe()

data1['target'].value\_counts()

plt.title('HEART DISEASE DISTRIBUTION')

data1['target'].value\_counts().plot(kind="pie",autopct='%.1f%%', figsize=(8,8),shadow=True)

plt.title('Age Vs Heart Disease')

sns.boxplot(x='target', y='age', data=data1,palette='rainbow')

data1.groupby(['sex'])['target'].value\_counts()

sns.countplot(x=data1['sex'],hue=data1['target'])

plt.show()

sns.boxplot(x='target', y='thalach', data=data1,palette='rainbow')

sns.stripplot(x="target", y="thalach", data=data1)

sns.scatterplot(data=data1, x="age", y="thalach", hue="target")

data1.groupby(['exang'])['target'].value\_counts()

sns.countplot(x=data1['exang'],hue=data1['target'])

plt.show()

sns.boxplot(x='target', y='oldpeak', data=data1,palette='rainbow')

data1.groupby(['slope'])['target'].value\_counts()

sns.countplot(x=data1['slope'],hue=data1['target'])

plt.show()

data1.groupby(['ca'])['target'].value\_counts()

sns.countplot(x=data1['ca'],hue=data1['target'])

plt.show()

sns.countplot(x=data1['thal'],hue=data1['target'])

plt.show()

plt.figure(figsize = (15,15))

sns.heatmap(data1.corr(), vmin = -1, vmax = +1, annot = True, cmap = 'coolwarm')

X = data1.drop(['target'],axis='columns')

xix

X.head(10)

y = data1.target

y.head(3)

len(x)

len(y)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X,y,test\_size=0.2,random\_state=10)

model\_1 = MultinomialNB()

model\_1.fit(X\_train, y\_train)

nb=model\_1.score(X\_test, y\_test)

nb

Logistic Regression

model\_2 = LogisticRegression()

model\_2.fit(X\_train, y\_train)

lr=model\_2.score(X\_test, y\_test)

lr

Random Forest

model\_3 = RandomForestClassifier(n\_estimators=30)

model\_3.fit(X\_train, y\_train)

rf=model\_3.score(X\_test, y\_test)

rf

Decision Tree

model\_4 = tree.DecisionTreeClassifier(criterion='entropy')

model\_4.fit(X\_train, y\_train)

dt=model\_4.score(X\_train, y\_train)

dt

Support Vector Machine(SVC)

model\_5 = SVC()

model\_5.fit(X\_train, y\_train)

sv=model\_5.score(X\_test, y\_test)

sv

accuracy = [nb,lr,rf,dt,sv]

all\_models =

['NaiveBayesClassifier','LogisticRegression','RandomForestClassifier','DecisonTreeClassifier'

,'SVC']

score\_df = pd.DataFrame({'Algorithms': all\_models, 'Accuracy\_Score': accuracy})

score\_df.style.background\_gradient(cmap="YlGnBu",high=1,axis=0)

mylist=[]

mylist2=[]

mylist.append(nb)

mylist2.append("Naive Bayes")

mylist.append(lr)

mylist2.append("Logistic Regression")

mylist.append(rf)

mylist2.append("Random Forest")

mylist.append(dt)

mylist2.append("Decision Tree")

mylist.append(sv)

mylist2.append("SVM")

plt.rcParams['figure.figsize']=8,6

sns.set\_style("darkgrid")

pal\_style=['#F38BB2','#4C0028','#8A0030','#100C07','#FF0000']

ax = sns.barplot(x=mylist2, y=mylist, palette = pal\_style, saturation =1.5)

plt.xlabel("Classification Models", fontsize = 20 )

plt.ylabel("Accuracy", fontsize = 20)

plt.title("Accuracy of different Classification Models", fontsize = 20)

plt.xticks(fontsize = 11, horizontalalignment = 'center', rotation = 8)

plt.yticks(fontsize = 13)

for p in ax.patches:

width, height = p.get\_width(), p.get\_height()

x, y = p.get\_xy()

ax.annotate(f'{height:.2%}', (x + width/2, y + height\*1.02), ha='center', fontsize = 'x-large')

plt.show()

model\_4.predict([[56,1,1,120,236,0,1,178,0,0.8,2,0,2]])

model\_4.predict([[57,1,0,130,131,0,1,115,1,1.2,1,1,3]])