* **Analyse the heartDisease.csv dataset to explore the machine learning algorithms and build decision tree models to predict the**

**disease.**

* import numpy as np
* import pandas as pd
* from sklearn.model\_selection import train\_test\_split
* from sklearn.linear\_model import LogisticRegression
* from sklearn.tree import DecisionTreeClassifier
* from sklearn.metrics import accuracy\_score, classification\_report
* import warnings
* warnings.filterwarnings("ignore")
* df = pd.read\_csv("HeartDisease.csv")
* df
* df.info()
* df["target"].value\_counts()
* x = df.iloc[:,:-1]
* x
* y = df.iloc[:,-1]
* y
* xtrain, xtest, ytrain, ytest = train\_test\_split(x,y, test\_size = 0.3, random\_state = 42)
* def mymodel(model):
* model.fit(xtrain, ytrain)
* ypred = model.predict(xtest)
* acc = accuracy\_score(ytest, ypred)
* cr = classification\_report(ytest, ypred)
* print(f"Accuracy is {acc} \n\n Classfication Report \n {cr}")
* lr = LogisticRegression()
* mymodel(lr)
* dt = DecisionTreeClassifier()
* mymodel(dt)
* dt.score(xtrain, ytrain)
* dt.score(xtest, ytest)
* acc\_list = []
* depth\_list = []
* for i in range(1,20):
* dt1 = DecisionTreeClassifier(max\_depth = i)
* dt1.fit(xtrain, ytrain)
* ypred = dt1.predict(xtest)
* acc = accuracy\_score(ytest, ypred)
* acc\_list.append(acc)
* depth\_list.append(i)
* print(f"The accuracy for max\_depth = {i} is {acc}")
* max\_acc = acc\_list[0]
* for i in range(1,len(acc\_list)):
* if acc\_list[i] > max\_acc:
* max\_acc = acc\_list[i]
* max\_index = acc\_list.index(max\_acc)
* print(f"We observed that, Highest accuracy for our model is we get when max\_depth is {max\_index + 1}")
* dt2 = DecisionTreeClassifier(max\_depth = 3)
* mymodel(dt2)
* dt2.score(xtrain, ytrain)
* acc\_list = []
* split\_list = []
* for i in range(2,100):
* dt3 = DecisionTreeClassifier(min\_samples\_split = i)
* dt3.fit(xtrain, ytrain)
* ypred = dt3.predict(xtest)
* acc = accuracy\_score(ytest, ypred)
* acc\_list.append(acc)
* split\_list.append(i)
* print(f"The accuracy for min\_samples\_split = {i} is {acc}")
* max\_acc = acc\_list[0]
* for i in range(1,len(acc\_list)):
* if acc\_list[i] > max\_acc:
* max\_acc = acc\_list[i]
* max\_index = acc\_list.index(max\_acc)
* print(f"We observed that, Highest accuracy for our model is we get when min\_samples\_split is {max\_index + 2}")
* dt4 = DecisionTreeClassifier(min\_samples\_split = 20)
* mymodel(dt4)
* dt4.score(xtrain, ytrain)
* acc\_list = []
* leaf\_list = []
* for i in range(1,50):
* dt5 = DecisionTreeClassifier(min\_samples\_leaf = i)
* dt5.fit(xtrain, ytrain)
* ypred = dt5.predict(xtest)
* acc = accuracy\_score(ytest, ypred)
* acc\_list.append(acc)
* leaf\_list.append(i)
* print(f"The accuracy for min\_samples\_leaf = {i} is {acc}")
* max\_acc = acc\_list[0]
* for i in range(1,len(acc\_list)):
* if acc\_list[i] > max\_acc:
* max\_acc = acc\_list[i]
* max\_index = acc\_list.index(max\_acc)
* print(f"We observed that, Highest accuracy for our model is we get when min\_samples\_leaf is {max\_index + 1}")
* dt6 = DecisionTreeClassifier(min\_samples\_leaf = 10)
* mymodel(dt6)
* dt6.score(xtrain, ytrain)
* dt7 = DecisionTreeClassifier(max\_depth = 3, min\_samples\_split = 20,min\_samples\_leaf = 10)
* mymodel(dt7)
* dt7.score(xtrain, ytrain)