CSE 422 Lab 07

Assignment 05

Logistic Regression and Decision Tree

Name: Subhi Bhuiyan

ID: 17201116 Section: 03

import pandas as pd

import numpy as np

heart\_dataset = pd.read\_csv('/content/heart failur classification dataset.csv')

heart\_dataset.head(3)

heart\_dataset.shape

heart\_dataset.isnull().sum()

heart\_dataset = heart\_dataset.drop(['time'], axis = 1)

heart\_dataset = heart\_dataset.drop(['serum\_sodium'], axis = 1)

heart\_dataset.shape

heart\_dataset.isnull().sum()

heart\_dataset['sex']=heart\_dataset['sex'].map({'Male':0,'Female':1})

heart\_dataset['smoking']=heart\_dataset['smoking'].map({'No':0,'Yes':1})

Logistic Regression

# Import the dependencies for logistic regression

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import classification\_report

from sklearn.metrics import accuracy\_score

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

X = heart\_dataset.iloc[:, :-1]

y = heart\_dataset.iloc[:, -1]

x\_train, x\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=0)

#Train the model

model = LogisticRegression()

model.fit(x\_train, y\_train) #Training the model

predictions = model.predict(x\_test)

print(predictions)

score1 = accuracy\_score(y\_test, predictions)

print("Accuracy using Logistic Regression")

print(score1)

Decision Tree

from sklearn.tree import DecisionTreeClassifier

clf = DecisionTreeClassifier(criterion='entropy',random\_state=1)

clf.fit(x\_train,y\_train)

y\_pred = clf.predict(x\_test)

score2 = accuracy\_score(y\_pred,y\_test)

print("Accuracy using Decision Tree")

print(score2)

from sklearn import tree

import matplotlib.pyplot as plt

fig, axes = plt.subplots(nrows = 1,ncols = 1,figsize = (4,4), dpi=300)

tree.plot\_tree(clf,

feature\_names = X.columns,

class\_names=['1','2','3','4','5','6','7'],

filled = True);

Comparing Accuracy

#barchart

plt.bar(['Logistic Regression', 'Decision Tree'],[score1, score2], align='center' ,alpha=0.5)

plt.title('Comparing Accuracy')

plt.ylabel("Accuracy")

plt.show()