import pandas as pd

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

from sklearn import tree

from sklearn.metrics import accuracy\_score, confusion\_matrix

data = pd.read\_csv('tips.csv')

pd.options.display.max\_columns = None

print(data)

print(data.info())

# Recode Tip Percent

def categorizeTip(tip):

if tip < .162:

return 0

else:

return 1

data['Tip Level'] = data['Tip percent'].apply(categorizeTip)

# Recode Day

day\_dict = {'Thur':0,'Fri':1,'Sat':2,'Sun':0}

data.day =[day\_dict[item] for item in data.day]

# Recode Time

time\_dict = {'Lunch':0,'Dinner':1}

data.time =[time\_dict[item] for item in data.time]

data.drop(columns = ['total\_bill'], inplace=True)

data.drop(columns = ['Tip percent'], inplace=True)

print(data.info())

# SEPARATE X and y

X = data.drop(columns=['Tip Level'])

y = data['Tip Level']

# split 60-40

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X,y,test\_size=.4)

# CREATE THE MODEL

#dectree = tree.DecisionTreeClassifier()

dectree = tree.DecisionTreeClassifier(criterion= 'entropy')

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

##

# HOW GOOD IS OUR MODEL ??

predictions = dectree.predict(X\_test)

print('\n\nAccuracy score:')

print(accuracy\_score(y\_test,predictions))

print('\n\nConfusion matrix:')

print(confusion\_matrix(y\_test,predictions))

import matplotlib.pyplot as plt

fig = plt.figure(figsize=(20,20))

tree.plot\_tree(dectree,feature\_names=X.columns.tolist())

plt.show()