## In [1]: import pandas as pd import numpy as np from sklearn import tree In [ ]: df = pd.read\_csv("input.csv") df.head() Age Income Student Credit Rating Buy Car Out[]: **0** 1 Young High No Fair No **1** 2 Young High Good No **2** 3 Middle High No Fair Yes **3** 4 Old Medium No Fair Yes **4** 5 Old Low Yes Fair Yes Converting to numerical data In [ ]: age = df['Age'].factorize(); income = df['Income'].factorize(); student = df['Student'].factorize(); credit\_rating = df['Credit Rating'].factorize(); buy\_car = df['Buy Car'].factorize(); In [ ]: factorized\_data = {'Age':age[0], 'Income':income[0], 'Student':student[0], 'Credit Rating':credit\_rating[0], 'Buy Car':buy\_car[0]} df1 = pd.DataFrame(factorized\_data) df1.head() Age Income Student Credit Rating Buy Car Out[ ]: 0 0 0 0 0 0 1 0 0 2 0 0 0 1 1 1 0 2 0 1 2 1 0 1 Dividing the data into training set and testing set X = df1[['Age','Income','Student','Credit Rating']] y = df1['Buy Car'] In [ ]: from sklearn.model\_selection import train\_test\_split $X_{train}, X_{test}, y_{train}, y_{test} = train_{test_split}(X, y, test_size = 0.2, random_state=1)$ print(X\_train.shape) print(y\_train.shape) print(X\_test.shape) print(y\_test.shape) (11, 4)(11,)(3, 4)(3,) **Training Classifier** clf = tree.DecisionTreeClassifier(criterion='entropy', max\_depth=3, random\_state=1) In [ ]: clf.fit(X\_train,y\_train) prediction = clf.predict(X\_test) prediction Out[]: array([1, 0, 1]) Testing our classifier from sklearn import metrics print("Accuracy = {}%".format(metrics.accuracy\_score(y\_test,prediction) \* 100)) Accuracy = 100.0% Finding root node In [ ]: import matplotlib.pyplot as plt fn=['Age','Income','Student','Credit Rating'] cn=['No', 'Yes'] plt.figure(figsize = (10,10)) tree.plot\_tree(clf, $feature_names = fn,$ class\_names=cn, filled = True); Credit Rating <= 0.5 entropy = 0.946 samples = 11 value = [4, 7] class = Yes Age <= 0.5 entropy = 0.65 Age <= 1.5 entropy = 0.971 samples = 5 value = [1, 5] class = Yes value = [3, 2] class = No Income <= 1.0 entropy = 1.0 Income <= 0.5 entropy = 0.918 entropy = 0.0 samples = 2 value = [2, 0] class = No entropy = 0.0 samples = 4 samples = 2 samples = 3 /alue = [0, 4] | class = Yes value = [1, 2] class = Yes value = [1, 1] class = No entropy = 0.0entropy = 0.0samples = 1samples = 1samples = 1samples = 2value = [1, 0] class = No value = [1, 0] class = No Classification on Diabeties data In [5]: from sklearn import datasets from sklearn.model\_selection import train\_test\_split from sklearn import metrics In [13]: df = pd.read\_csv("diabetes.csv", header=None) df.head() 0 1 2 3 4 5 6 7 8 0 33.6 0.627 50 1 **0** 6 148 72 35 **1** 1 85 66 29 0 26.6 0.351 31 0 **2** 8 183 64 0 0 23.3 0.672 32 1 **3** 1 89 66 23 94 28.1 0.167 21 0 **4** 0 137 40 35 168 43.1 2.288 33 1 In [14]: X = df[[0,1,2,3,4,5,6,7]]y = df[[8]]In [15]: $X_{train}, X_{test}, y_{train}, y_{test} = train_{test_split}(X, y, test_size = 0.2, random_state=1)$ print(X\_train.shape) print(y\_train.shape) print(X\_test.shape) print(y\_test.shape) (614, 8)(614, 1)(154, 8)(154, 1)clf = tree.DecisionTreeClassifier(criterion='entropy', max\_depth=3, random\_state=1) In [30]: clf.fit(X\_train,y\_train) prediction = clf.predict(X\_test) prediction Out[30]: array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]) In [31]: from sklearn import metrics print("Accuracy = {}%".format(metrics.accuracy\_score(y\_test,prediction) \* 100)) Accuracy = 79.87012987012987%In [32]: import matplotlib.pyplot as plt plt.figure(figsize = (20,20)) tree.plot\_tree(clf, filled = True); X[1] <= 127.5 entropy = 0.931samples = 614 value = [401, 213] X[5] <= 26.45 X[1] <= 166.5 entropy = 0.707entropy = 0.963samples = 389 samples = 225 value = [314, 75] value = [87, 138] X[5] <= 9.1 entropy = 0.181 $X[7] \le 28.5$ X[5] <= 29.95 X[1] <= 172.5 entropy = 0.824 entropy = 0.999 entropy = 0.614samples = 110 samples = 279samples = 159 samples = 66 value = [107, 3] value = [10, 56] value = [207, 72] value = [77, 82] entropy = 0.863entropy = 0.079entropy = 0.568entropy = 0.963entropy = 0.867entropy = 0.954entropy = 0.0entropy = 0.699samples = 142samples = 107 samples = 13samples = 53samples = 7samples = 103samples = 137samples = 52valuė = [102, 1] value = [123, 19] value = [10, 43] value = [5, 2]value = [84, 53] value = [37, 15] value = [40, 67] value = [0, 13] In [32]:

Classification using Decision Tree

Importing necessary libraries