Data Mining lab

Assignment 5

Name: Ayush Kumar

Regno: 20214284

Group: CSE-6B1

1.

import pandas as pd import numpy as np

# Step 1: Load the dataset data = pd.read\_csv('page\_block.csv')

# Step 2: Compute the mean of attribute values against both class labels mean\_negative\_class = data[data['class'] == 'negative'].mean() mean\_positive\_class = data[data['class'] == 'positive'].mean()

# Step 3: Compute the standard deviation of attribute values against both class labels std\_negative\_class = data[data['class'] == 'negative'].std() std\_positive\_class = data[data['class'] == 'positive'].std()

# Step 4: Score (f) for each attribute f\_score = (mean\_positive\_class - mean\_negative\_class) / (std\_positive\_class + std\_negative\_class)

# Step 5: Assign a rank for each attribute ranked\_features

= f\_score.rank(ascending=False)

# Select the top k ranked features k = 5 # Replace with the desired value of k selected\_features = ranked\_features.nlargest(k)

print(selected\_features)

2.

import pandas as pd import numpy as np

# Step 1: Load the dataset

data = pd.read\_csv('buys\_computer.csv')

# Step 2: Compute the entropy (info(dataset)) of the entire dataset def entropy(labels):

n\_labels = len(labels) if n\_labels <= 1:

return 0 value, counts = np.unique(labels, return\_counts=True)

probs = counts / n\_labels n\_classes = np.count\_nonzero(probs) if n\_classes <= 1:

return 0 ent = 0. for i in probs:

ent -= i \* np.log2(i) return ent

dataset\_entropy = entropy(data['class'])

# Step 3: Compute InfoA(Dataset) for each attribute A in the dataset def info\_A(dataset, attribute):

info = 0 for value in dataset[attribute].unique():

subset = dataset[dataset[attribute] == value] info += (len(subset) / len(dataset)) \* entropy(subset['class']) return info

info\_A\_values = {} for attribute in data.columns[:-1]:

info\_A\_values[attribute] = info\_A(data, attribute)

# Step 4: Compute Information Gain (A) for each attribute A in the dataset def information\_gain(dataset\_entropy, info\_A\_values):

gain\_values = {} for attribute, info\_A in info\_A\_values.items():

gain\_values[attribute] = dataset\_entropy - info\_A return gain\_values

information\_gain\_values = information\_gain(dataset\_entropy, info\_A\_values)

# Step 5: Select k attributes with the highest information gain k = 3 # Replace with the desired value of k selected\_attributes

= sorted(information\_gain\_values, key=information\_gain\_values.get, reverse=True)[:k]

print(selected\_attributes)