CSE422 Lab Assignment 4

Md. Shamiul Islam

ID: 17301108

Theory Section: 8

Lab Section: 4

**import numpy as np**

**import pandas as pd**

**shamiul = pd.read\_csv('/content/mushroom edibility classification dataset.csv')**

**shamiul.head()**

**shamiul.shape**

**shamiul.isnull().sum()**

**print("rows with null values in cap-color column",shamiul["cap-color"].isnull().sum())**

**print("rows with null values in cap-shape column",shamiul["cap-shape"].isnull().sum())**

**shamiul= shamiul.dropna(axis=0, subset=["cap-shape","cap-color"])**

**print("Shape after removing null values: ", shamiul.shape)**

**shamiul.shape**

**shamiul["class"].unique()**

**shamiul["bruises"].unique()**

**#encoding part**

**from sklearn.preprocessing import LabelEncoder**

**enc = LabelEncoder()**

**shamiul['class'] = enc.fit\_transform(shamiul['class'])**

**shamiul['bruises'] = enc.fit\_transform(shamiul['bruises'])**

**shamiul.head()**

**type(shamiul)**

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

**#from sklearn.metrics import classification\_report**

**#from sklearn.metrics import accuracy\_score**

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

**terget = shamiul["class"].values**

**dataset = shamiul.drop(["class"], axis=1).values**

**x\_train, x\_test, y\_train, y\_test = train\_test\_split(dataset, terget, test\_size=0.2, random\_state=42)**

**print(x\_train.shape)**

**print(x\_test.shape)**

**print(y\_train.shape)**

**print(y\_test.shape)**

**print("per-feature minimum before scaling:\n {}".format(x\_train.min(axis=0)))**

**print("per-feature maximum before scaling:\n {}".format(x\_train.max(axis=0)))**

**#processing MinMaxScaler**

**from sklearn.preprocessing import MinMaxScaler**

**scaler = MinMaxScaler()**

**scaler.fit(x\_train)**

**X\_train\_scaled = scaler.transform(x\_train)**

**X\_test\_scaled = scaler.transform(x\_test)**

**print("per-feature minimum after scaling:\n {}".format(**

**X\_train\_scaled.min(axis=0)))**

**print("per-feature maximum after scaling:\n {}".format(**

**X\_train\_scaled.max(axis=0)))**

**x\_train**

**x\_test**