*# -\*- coding: utf-8 -\*-*

*"""Lab5\_DataPreprocessing\_18201160.ipynb*

*Automatically generated by Colaboratory.*

*Original file is located at*

*https://colab.research.google.com/drive/1fE07vOFXUiXUV5nw7awXFbcMP7rerSRE*

*1. Found the dataset "mushroom edibility classification".*

*"""*

import pandas as pd

import numpy as np

*"""3. Loading the dataset as dataframe using pandas"""*

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

print(mushroom.shape)

mushroom.head()

#Dropping the 'Unnamed:0' column

mushroom = mushroom.drop('Unnamed: 0', axis=1)

*"""4. Handling the missing values after checking if it's needed or not-"""*

mushroom.isnull()

mushroom.isnull().sum()

*"""Since only a few rows of only 'cap-shape' and 'cap-color' are null we will not drop any columns or rows in whole.*

*Have to handle the Null values in 'cap-shape' and 'cap-color' by Imputing them.*

*"""*

# Imputting

from sklearn.impute import SimpleImputer

impute = SimpleImputer(missing\_values=np.nan, strategy='mean')

impute.fit(mushroom[['cap-shape','cap-color']])

mushroom[['cap-shape','cap-color']] = impute.transform(mushroom[['cap-shape','cap-color']])

# These two columns after imputting.

mushroom[['cap-shape','cap-color']].isnull().sum()

*"""5. Encoding the categorical features after checking if it's needed or not-"""*

mushroom.info()

mushroom['class'].unique()

mushroom['bruises'].unique()

*"""Since both 'class' and 'bruises' have two unique types of values each, I am going to use Binary Categorical Values Encoding."""*

from sklearn.preprocessing import LabelEncoder

# Set up the LabelEncoder object

enc = LabelEncoder()

# Apply the encoding to the "class" and "bruises" column

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

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

print(mushroom[['class']].head())

print(mushroom[['bruises']].head())

*"""6. Scaling all the values between 0-1 with proper scaling technique"""*

from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()

scaler.fit(mushroom)

mushroom\_scaled = scaler.transform(mushroom)

pd.DataFrame(mushroom\_scaled, columns = list(mushroom))

mushroom\_scaled

mushroom.info()

*"""7. Split the dataset into features and labels. Use your intuition to determine which column indicates the labels."""*

*'''*

*'class' is the label because it tells us if the mushrooms are edible or not.*

*'''*

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

from sklearn.neighbors import KNeighborsClassifier

label=pd.DataFrame(mushroom\_scaled[:,0])

X\_train, X\_test, y\_train, y\_test = train\_test\_split(mushroom\_scaled[:,1:], mushroom\_scaled[:,0],test\_size=0.25, stratify=label, random\_state=0)

knn=KNeighborsClassifier()

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

print("Test set accuracy: {:.2f}".format(knn.score(X\_test, y\_test)))