CSE422

Lab05 Data Preprocessing

Name: Subhi Bhuiyan

ID: 17201116

Section: 03

#importing libraries

import pandas as pd

import numpy as np

heart\_dataset = pd.read\_csv('/content/sample\_data/heart failur classification dataset.csv')

heart\_dataset.head(3)

heart\_dataset.shape

heart\_dataset.isnull().sum()

dropping columns

heart\_dataset = heart\_dataset.drop(['time'], axis = 1)

heart\_dataset.shape

heart\_dataset.isnull().sum()

dropping rows

# Check how many values are missing in the serum\_sodium column

print("Number of rows with null values in serum\_sodium column: ", heart\_dataset['serum\_sodium'].isnull().sum())

# Subset the heart dataset

heart\_dataset\_subset = heart\_dataset[heart\_dataset['serum\_sodium'].notnull()]

# Print out the shape of the subset

print("Shape after removing null values: ", heart\_dataset\_subset.shape)

print("Shape of dataframe before dropping:", heart\_dataset.shape)

heart\_dataset = heart\_dataset.dropna(axis = 0, subset = ['serum\_sodium'])

print("Shape after dropping:", heart\_dataset.shape)

Encoding Categorical Features

heart\_dataset.info()

heart\_dataset['sex'].unique()

heart\_dataset['smoking'].unique()

from sklearn.preprocessing import LabelEncoder

# Set up the LabelEncoder object

enc = LabelEncoder()

heart\_dataset['sex']= enc.fit\_transform(heart\_dataset['sex'])

heart\_dataset['smoking']= enc.fit\_transform(heart\_dataset['smoking'])

heart\_dataset

Feature Scaling

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

y = pd.DataFrame(heart\_dataset['DEATH\_EVENT'])

X\_train, X\_test, y\_train, y\_test = train\_test\_split(heart\_dataset, heart\_dataset['DEATH\_EVENT'], test\_size = 0.25, random\_state=0, stratify = y)

print("X\_train shape:", X\_train.shape)

print("y\_train shape:", y\_train.shape)

print("X\_test shape:", X\_test.shape)

print("y\_test shape:", y\_test.shape)

from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()

scaler.fit(X\_train)

#transform data

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

X\_train\_scaled

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)))

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)))

# transform test data

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

from sklearn.neighbors import KNeighborsClassifier

knn=KNeighborsClassifier()

scaler = MinMaxScaler()

scaler.fit(X\_train)

#train

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

# scoring on the scaled test set

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