Feedforward\_NN

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

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

from sklearn.preprocessing import StandardScaler

import tensorflow as tf

from imblearn.over\_sampling import RandomOverSampler

df = pd.read\_csv("diabetes.csv")

df.head()

for i in range(len(df.columns[:-1])):

   label = df.columns[i]

   plt.hist(df[df['Outcome'] ==1][label], color='blue', label ="Diabetes",alpha =0.7, density = True, bins =15)

   plt.hist(df[df['Outcome'] ==0][label], color='red', label ="No diabetes", alpha =0.7, density =True, bins =15)

   plt.title(label)

   plt.ylabel("Probability")

   plt.xlabel(label)

   plt.legend()

   plt.show()

X = df[df.columns[:-1]].values

y = df[df.columns[-1]].values

scaler = StandardScaler()

X = scaler.fit\_transform(X)

data = np.hstack((X,np.reshape(y,(-1,1))))

transformed\_df = pd.DataFrame( data, columns = df.columns)

over = RandomOverSampler()

X,y = over.fit\_resample(X, y)

data = np.hstack((X,np.reshape(y,(-1,1))))

transformed\_df = pd.DataFrame( data, columns = df.columns)

X\_train, X\_temp, y\_train, y\_temp = train\_test\_split(X,y, test\_size=0.4, random\_state= 0)

X\_valid, X\_test, y\_valid, y\_test = train\_test\_split(X\_temp,y\_temp, test\_size=0.5, random\_state= 0)

model = tf.keras.Sequential([

                              tf.keras.layers.Dense(16, activation='relu'),

                              tf.keras.layers.Dense(16, activation='relu'),

                              tf.keras.layers.Dense(1, activation='sigmoid')

])

model.compile(optimizer=tf.keras.optimizers.Adam(learning\_rate =0.01),

              loss=tf.keras.losses.BinaryCrossentropy(),

              metrics =['accuracy'])

model.evaluate(X\_train,y\_train)

model.evaluate(X\_valid,y\_valid)

model.evaluate(X\_test,y\_test)