**Practical 11**

**Aim:**

**Implement multiclass classification with neural network on Iris flower species.**

**Code:**

from sklearn.datasets import load\_iris

from sklearn.cluster import KMeans

import pandas as pd

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

from keras.models import Sequential

from keras.layers import Dense

from keras.utils import np\_utils

from sklearn.preprocessing import LabelEncoder

dataset = load\_iris()

X = dataset.data

Y = dataset.target

LE = LabelEncoder()

LE.fit(Y)

e\_Y = LE.transform(Y)

Y = np\_utils.to\_categorical(e\_Y)

X\_train,X\_test,Y\_train,Y\_test = train\_test\_split(X,Y,train\_size=0.8)

model = Sequential()

model.add(Dense(7,input\_dim=4,activation='sigmoid'))

model.add(Dense(3,activation='softmax'))

model.compile(loss='binary\_crossentropy',

optimizer='adam',

metrics=['accuracy'])

import matplotlib.pyplot as plt

history = model.fit(X\_train,Y\_train,epochs=20,batch\_size=5,validation\_split=0.2)

plt.plot(history.history['acc'])

plt.plot(history.history['val\_acc'])

plt.title('Model accuracy')

plt.ylabel('Accuracy')

plt.xlabel('Epoch')

plt.legend(['Train', 'Test'], loc='upper left')

plt.show()

# Plot training & validation loss values

plt.plot(history.history['loss'])

plt.plot(history.history['val\_loss'])

plt.title('Model loss')

plt.ylabel('Loss')

plt.xlabel('Epoch')

plt.legend(['Train', 'Test'], loc='upper left')

plt.show()

\_, accuracy = model.evaluate(X\_test,Y\_test)

**Output:**



