**Module 7**

**Name:** Upadhyay Sachin Naresh

**Batch ID:** AIWDEOB 300821

## **Topic:** Deep learning practical issues

**Q. Build an ANN model to predict Delivery Status for a logistic firm. (Refer FedEx dataset)**

# Importing required library

import pandas as pd

from sklearn.preprocessing import LabelEncoder from sklearn.model\_selection import train\_test\_split from sklearn.preprocessing import StandardScaler import seaborn as sns

import keras

from keras.models import Sequential from keras.layers import Dense

from keras.wrappers.scikit\_learn import KerasClassifier from sklearn.model\_selection import cross\_val\_score

# load the dataset (fedex dataset)

path = "C:/Users/usach/Desktop/AI assignments/Module 7/fedex.csv"

" fedex\_data = pd.read\_csv(path)

# Copy the path or mount file to Google drive in order to load in Google collab Check for presence of Nan values

fedex\_data.isnull().sum().sum()

# dropping the rows having Nan values and check

fedex\_data = fedex\_data.dropna() fedex\_data.isnull().sum().sum()

# Viewing top 5 FedEx data

fedex\_data.head(5)

# Checking Column Names

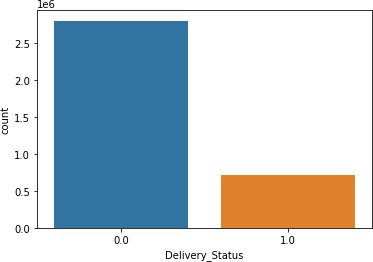
fedex\_data.columns

# Viewing target

fedex\_data['Delivery\_Status'].head(5)

# Summary of Delivery\_Status

sns.countplot(fedex\_data['Delivery\_Status'],label="Count")



# Combining source and destination column into new Path column since source and destination gives path

fedex\_data['Path'] = fedex\_data[['Source', 'Destination']].apply(lambda x: '-'.join(x), axis = 1) fedex\_data['Path'].head(5)

# Droping 'Source' and 'Destination' along with 'Carrier\_Name' since 'Carrier\_Num' is already pres ent, 'Carrier\_Num' is ID for 'Carrier\_Name' so keeping only 'Carrier\_Num'

fedex\_data.drop(['Source', 'Destination', 'Carrier\_Name'], axis = 1, inplace= True)

# Encoding categorical data, Since NN takes only numerical inputs

from sklearn.preprocessing import LabelEncoder le = LabelEncoder()

fedex\_data['Path'] = le.fit\_transform(fedex\_data['Path'])

# Separating Independent and dependent features

X = fedex\_data.iloc[:, fedex\_data.columns != 'Delivery\_Status'].values #independent y = fedex\_data['Delivery\_Status'].values #dependent

# Viewing Independent and dependent columns

print(X[:,:5]) print(y)

# Splitting the data into train and test with test size as 30%

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y,test\_size = 0.3, random\_state = 0) #Scaling independent variables

sc = StandardScaler()

X\_train = sc.fit\_transform(X\_train)

X\_test = sc.transform(X\_test)

# viewing scaled data before building model

X\_train[:,:5]

# Printing Shapes of input along with output values

print("X\_train",X\_train.shape) print("X\_test",X\_test.shape) print("y\_train",y\_train.shape) print("y\_test",y\_test.shape)

## Defining the model inside a function and

Adding the input and hidden layer followed by output layer also compiling the model

def build\_model():

model = Sequential() model.add(Dense(10,activation='relu',input\_shape=(12,))) #adding the second hidden layer

model.add(Dense(6, activation='relu')) #adding the output layer model.add(Dense(1, activation='sigmoid')) #compiling model

model.compile(optimizer='adam',loss='binary\_crossentropy',metrics=['accuracy']) return model

# Fitting model by using defined model function which as sequential layers.

classifier = KerasClassifier(build\_fn = build\_model , batch\_size = 512 , nb\_epoch = 20)

# Evaluating model using cross-validation technique having 10 Folds by using all available concurrent workers

accuracy = cross\_val\_score(estimator = classifier , X=X\_train, y=y\_train, cv =10 , n\_jobs = -1)

# Printing Mean accuracy of model

print("Mean accuracy:",format(100\*accuracy.mean(),".2f"),"%")