ASSIGNMENT 4 DL - RNN

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

df_train = pd.read_csv('trainset.csv')
df_test = pd.read_csv('testset.csv')

train_set = df_train.iloc[:,1:2].values

from sklearn.preprocessing import MinMaxScaler
sc = MinMaxScaler()

train_set = sc.fit_transform(train_set)

x_train = train_set[0:1257]
y_train = train_set[1:1258]

x_train = np.reshape(x_train,(1257,1,1))

from keras.models import Sequential
from keras.layers import LSTM
from keras.layers import Dense
from keras.optimizers import Adam
from keras.losses import mean_squared_error

model = Sequential()
model.add(LSTM(units=4,activation='sigmoid'))
model.add(Dense(units=1))
model.compile(optimizer='Adam',loss='mean_squared_error')


model.fit(x_train,y_train,batch_size=32,epochs=200)


df_test = df_test.iloc[:,1:2].values


temp = df_test


temp = sc.transform(temp)
temp = np.reshape(temp,(125,1,1))
pred = model.predict(temp)
pred = sc.inverse_transform(pred)


plt.plot(pred,color = 'blue', label = 'Predicted Prices')
plt.plot(df_test, color='red', label = 'Real')
```

ASSIGNMENT 3 DL - CNN

```
import numpy as np

import pandas as pd

df_train = pd.read_csv('fashion-mnist_train.csv')

df_test = pd.read_csv('fashion-mnist_test.csv')

train = np.array(df_train,dtype='float32')

test = np.array(df_test,dtype='float32')

x_temp = train[:,1:]/255

y_temp = train[:,0]

x_test = test[:,1:]/255

y_test = test[:,0]


from sklearn.model_selection import train_test_split

x_train, x_valid , y_train, y_valid = train_test_split(x_temp,y_temp,test_size=0.2,random_state=0)

x_train = x_train.reshape(x_train.shape[0],28,28,1)

x_test = x_test.reshape(x_test.shape[0],28,28,1)

x_valid = x_valid.reshape(x_valid.shape[0],28,28,1)


from keras.models import Sequential

from keras.layers import Conv2D

from keras.layers import Dense

from keras.layers import MaxPooling2D

from keras.layers import Dropout

from keras.layers import Flatten

from keras.optimizers import Adam

from keras.losses import sparse_categorical_crossentropy


model = Sequential()

model.add(Conv2D(filters = 64, kernel_size=(3,3), input_shape = (28,28,1), activation='relu'))

model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.2))
model.add(Flatten())
model.add(Dense(units=64,activation='relu'))
model.add(Dense(units=32))
model.add(Dense(units=10, activation='sigmoid'))
model.compile(loss='sparse_categorical_crossentropy', optimizer='adam', metrics='accuracy')
model.fit(x_train,y_train,batch_size=512,epochs=10,validation_data=(x_valid,y_valid))

acc = model.evaluate(x_test,y_test)

print("MOdel accuracy is ",acc)

pred_x = model.predict(x_test)
pred_classes = np.argmax(pred_x,axis=1)

from sklearn.metrics import classification_report
classes = 10
targets = ["Class {}".format(i) for i in range(classes)]
print(classification_report(y_test,pred_classes,target_names=targets))


import matplotlib.pyplot as plt

index = 45  # Specify the index of the element you want to display

plt.imshow(x_test[index].reshape(28, 28))
plt.title('Predicted = {:0.1f} \n Actual = {:0.1f}'.format(pred_classes[index], y_test[index]))
plt.axis('off')

plt.show()
```

ASSIGNMENT 2 DL – Binary Classification

```
from keras.datasets import imdb
import numpy as np

(train_data, train_labels) , (test_data,test_labels) = imdb.load_data(num_words=100000)

def vectorize(sequences, dimension = 100000):
    results = np.zeros((len(sequences), dimension))
    for i, sequence in enumerate(sequences):
        results[i, sequence] = 1
    return results

x_train = vectorize(train_data)
x_test = vectorize(test_data)
y_train = np.array(train_labels, dtype='float16')
y_test = np.array(test_labels,dtype='float16')

from keras.models import Sequential
from keras.layers import Dense
from keras.optimizers import Adam
from keras.losses  import binary_crossentropy

model = Sequential()
model.add(Dense(units=16,activation='relu', input_shape=(100000,)))
model.add(Dense(units=8,activation='relu'))
model.add(Dense(units=1,activation='sigmoid'))
model.compile(optimizer='adam',loss='binary_crossentropy', metrics='accuracy')
model.fit(x_train,y_train,batch_size=8, epochs=20, validation_data=(x_test,y_test))
acc = model.evaluate(x_test,y_test)
```

ASSIGNMENT 1 DL – Linear Regression

```
import pandas as pd

df = pd.read_csv('HousingData.csv')

df.isnull().sum()

df["CRIM"].fillna(df["CRIM"].mean(), inplace=True)
df["ZN"].fillna(df["ZN"].mean(), inplace=True)
df["INDUS"].fillna(df["INDUS"].mean(), inplace=True)
df["CHAS"].fillna(df["CHAS"].mean(), inplace=True)
df["AGE"].fillna(df["AGE"].mean(), inplace=True)
df["LSTAT"].fillna(df["LSTAT"].mean(), inplace=True)

df.isnull().sum()

x = df.drop('MEDV',axis=1)
y = df['MEDV']

from sklearn.model_selection import train_test_split

x_train,x_test,y_train,y_test = train_test_split(x,y,random_state=42)

from keras.models import Sequential
from keras.layers import Dense
from keras.optimizers import Adam
from keras.losses import mean_squared_error

model = Sequential()
model.add(Dense(units=32,activation='relu'))
model.add(Dense(units=16,activation='relu'))
model.add(Dense(units=8,activation='relu'))
model.add(Dense(units=1,activation='linear'))

model.compile(optimizer='adam', loss='mean_squared_error', metrics= 'accuracy')

model.fit(x_train,y_train,epochs=32)
```