NAME: P NIKHIL KRISHNA

ROLL NO: HU21CSEN0300328

USECASE - 2

Directory

In [2]:
```
import os
os.getcwd()
```

Out[2]:
```
'C:\\Users\\mpaga\\Desktop\\DL USECASE-2'
```

Importing Libraries

In [3]:
```
!pip install keras-preprocessing
import keras
from keras_preprocessing.sequence import pad_sequences
from keras.layers import Embedding, LSTM, Dense, Dropout
from keras.preprocessing.text import Tokenizer
from keras.callbacks import EarlyStopping
from keras.models import Sequential
import keras.utils as ku
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: keras-preprocessing in c:\users\mpaga\appdata\roaming
\python\python310\site-packages (1.1.2)
Requirement already satisfied: numpy>=1.9.1 in c:\users\mpaga\appdata\roaming\python
\python310\site-packages (from keras-preprocessing) (1.24.3)
Requirement already satisfied: six>=1.9.0 in c:\programdata\anaconda3\lib\site-packag
es (from keras-preprocessing) (1.16.0)
WARNING:tensorflow:From C:\Users\mpaga\AppData\Roaming\Python\Python310\site-packages
\keras\src\losses.py:2976: The name tf.losses.sparse_softmax_cross_entropy is depreca
ted. Please use tf.compat.v1.losses.sparse_softmax_cross_entropy instead.
```

In [5]:
```
import tensorflow as tf
tf.random.set_seed(1)
import pandas as pd
import numpy as np
import string, os

import warnings
warnings.filterwarnings("ignore")
warnings.simplefilter(action='ignore', category=FutureWarning)
```

Loading the Dataset

In [6]:
```
data= pd.read_csv('USECASE2.csv')
```

In [7]:
```
data.head(20)
```

Out[7]:

	Name	Description
0	Nikhil	Nikhil is a movie buff with a particular fondn...
1	Manusree	Manusree is a fashion designer who creates ele...
2	Sreya	Sreya runs a successful bakery chain in Hydera...
3	Vamsi	Vamsi loves watching movie.His excitement peak...
4	Arjun	Arjun has recently started his startup company...
5	Toretto	Toretto is a kind-hearted person who cares abo...
6	Letty	Letty combines her unparalleled driving skill...
7	Preeti	Preeti's cousin is a travel blogger who share...
8	Chanakya	Chanakya's digital marketing strategies are li...
9	Nisha	Nisha's confidence made our project a success
10	Anika	Anika secured a First Rank in her school when ...
11	Sahiti	Sahithi mesmerizes audiences with her captivat...
12	Aryan	Arya pioneers in the gaming industry, leading ...
13	Manohar	Manohar is a big fan of James Bond Flims 007
14	Isha	Isha is known for her creativity and talent.
15	Lekhya	Lekhya and Isha are inseparable and always tog...
16	Nainika	Nanikia is an amazing dancer who lights up the...
17	Nisman	Nisman's reels are like snippets from our own ...
18	Rashmitha	Rashmitha always shows full dedication to her ...
19	Ritika	Ritika, a self-proclaimed gym freak, has an in...

In [8]:
```
text=pd.DataFrame()
text['Description']=data.Description
```

In [9]:
```
text.shape
```

Out[9]:
```
(50, 1)
```

In [10]:
```
text.head(5)
```

Out[10]:

	Description
0	Nikhil is a movie buff with a particular fondn...
1	Manusree is a fashion designer who creates ele...
2	Sreya runs a successful bakery chain in Hydera...
3	Vamsi loves watching movie.His excitement peak...
4	Arjun has recently started his startup company...

In [12]:

```
text.describe()
```

Out[12]:

	Description
count	50
unique	50
top	Nikhil is a movie buff with a particular fondn...
freq	1

In [11]:

```
all_descriptions = []


all_descriptions= [h for h in text.Description if h != "Unknown"]
len(all_descriptions)
```

Out[11]:

```
50
```

In [13]:

```
all_descriptions
```

```
Out[13]: ["Nikhil is a movie buff with a particular fondness for Fast and Furious, alongside h
         is love for cinema. When he's not enjoying high-octane action on screen, he's likely
         honing his badminton skills.",
          "Manusree is a fashion designer who creates elegant Indian ethnic wear. She's inspir
         ed by traditional craftsmanship and loves incorporating vibrant colors into her desig
         ns",
          'Sreya runs a successful bakery chain in Hyderabad with over 10 outlets. Her dedicat
         ion and passion for baking have made her a household name in the city',
          'Vamsi loves watching movie.His excitement peaks when he secures tickets for first-d
         ay-first-show screenings, eager to be among the first to witness the magic unfold.\n
         \n\n\n',
          "Arjun has recently started his startup company. He's known for his innovative ideas
         and strategic business acumen.",
          'Toretto is a kind-hearted person who cares about his family',
          'Letty  combines her unparalleled driving skills with a resilient spirit, making her
         a force to be reckoned with on and off the streets.',
          "Preeti's cousin  is a travel blogger who shares her experiences exploring different
         parts of India and the world. Her blog inspires others to travel and experience new c
         ultures.",
          "Chanakya's digital marketing strategies are like a symphony of success, orchestrati
         ng brand visibility and business growth with precision",
          "Nisha's confidence made our project a success",
          'Anika secured a First Rank in her school when she was in 9th class',
          'Sahithi mesmerizes audiences with her captivating content and InstaReels.',
          'Arya pioneers in the gaming industry, leading players on epic adventures through vi
         rtual realms where imagination knows no bounds.',
          'Manohar is a big fan of James Bond Flims 007',
          'Isha is known for her creativity and talent.',
          "Lekhya and Isha are inseparable and always together. They share everything, from se
         crets to snacks, and they're always there for each other no matter what",
          "Nanikia is an amazing dancer who lights up the stage with her moves. She's so good
         that she even participated in DHEE!",
          "Nisman's reels are like snippets from our own life, capturing moments that feel so
         relatable",
          'Rashmitha always shows full dedication to her studies .Her determination and resili
         ence have made her a role model among her peers.',
          'Ritika, a self-proclaimed gym freak, has an insatiable passion for fitness that set
         s her apart from the crowd.She is known for her cool and jovial personality',
          "Bharawaj is known for his cool and composed nature, a person who doesn't let small
         things ruffle his feelings",
          'Vijay always dream to become a successful superstar',
          "Vishnu always gets roasted by his friends. He feels sad but doesn't express it ",
          'Venu is a helpful friend who always lends a hand when you need it',
          'Gopal has to work on his anger issues',
          'Tejaswini is a smart student who loves learning new things',
          'Sreeleela dances with grace and joy, spreading happiness with every step',
          'Kanupriya and her best friend always explores new places.',
          'Rohit shapes cultural narratives in entertainment, captivating audiences with his c
         ompelling storytelling and visionary filmmaking',
          'Samantha shines on stage or screen, bringing characters to life with her love for a
         cting',
          'Revanth has a beautiful voice and loves to sing',
          'Kriti Sanon is known for her dedication and commitment to every event she undertake
         s',
          'Jason loves watching  action movies. He always observes different type of actions
         ',
          'Yashna loves her parents very much. She is gonna surprise them next week ',
          'Divya is a teacher who teaches Hindi at a primary school. She is passionate about h
         er language and culture',
          'Vennela is a student who is studying medicine at a top university',
          "Bhoomi's art designs are captivatingly intriguing, drawing viewers into a world of
         creativity and imagination",
          'Meenakshi',
          'Chaitanya is a hard worker who always gives his best effort',
          'Vishwa is a good friend who always been supportive',
          'Manognaa is learning to play badminton',
          'Karthikey showcasts his creativity and skill in transforming raw footage into capti
         vating content',
          'Rudhvik enjoys spending time with his close friends from 10th grade',
          'Vishwa is an introverted individual who values his solitude and prefers quiet conte
         mplation over social gatherings',
          'Sravya is a diligent student who devotes a significant amount of time to studying',
          'Rishit always completes a new game in 2 days',
          'Yashawi loves to travel, eagerly  wants to explore new destinations',
          'Rohan always sleeps during classes',
          'Sandeep once caught his friend hiding his pen in his bag',
          'Akshay, once a dull student, now owns many buildings']
```

DATASET PREPARTION

Dataset Cleaning

In [14]:
```python
def clean_text(txt):
    txt = "".join(v for v in txt if v not in string.punctuation).lower()
    txt = txt.encode("utf8").decode("ascii",'ignore')
    return txt

corpus = [clean_text(x) for x in all_descriptions]
corpus[:10]
```

Out[14]:
```
['nikhil is a movie buff with a particular fondness for fast and furious alongside hi
s love for cinema when hes not enjoying highoctane action on screen hes likely honing
his badminton skills',
 'manusree is a fashion designer who creates elegant indian ethnic wear shes inspired
by traditional craftsmanship and loves incorporating vibrant colors into her design
s',
 'sreya runs a successful bakery chain in hyderabad with over 10 outlets her dedicati
on and passion for baking have made her a household name in the city',
 'vamsi loves watching moviehis excitement peaks when he secures tickets for firstday
firstshow screenings eager to be among the first to witness the magic unfold\n\n\n
\n',
 'arjun has recently started his startup company hes known for his innovative ideas a
nd strategic business acumen',
 'toretto is a kindhearted person who cares about his family',
 'letty  combines her unparalleled driving skills with a resilient spirit making her
a force to be reckoned with on and off the streets',
 'preetis cousin  is a travel blogger who shares her experiences exploring different
parts of india and the world her blog inspires others to travel and experience new cu
ltures',
 'chanakyas digital marketing strategies are like a symphony of success orchestrating
brand visibility and business growth with precision',
 'nishas confidence made our project a success']
```

Generating Sequence of N-gram Tokens

In [15]:
```
all_descriptions[0]
```

Out[15]:
```
"Nikhil is a movie buff with a particular fondness for Fast and Furious, alongside hi
s love for cinema. When he's not enjoying high-octane action on screen, he's likely h
oning his badminton skills."
```

In [16]:
```
tokenizer = Tokenizer()

def get_sequence_of_tokens(corpus):
    ## tokenization
    tokenizer.fit_on_texts(corpus)
    total_words = len(tokenizer.word_index) + 1

    ## convert data to sequence of tokens
    input_sequences = []
    for line in corpus:
        token_list = tokenizer.texts_to_sequences([line])[0]
        for i in range(1, len(token_list)):
            n_gram_sequence = token_list[:i+1]
            input_sequences.append(n_gram_sequence)
    return input_sequences, total_words

inp_sequences, total_words = get_sequence_of_tokens(corpus)
inp_sequences[:10]
```

Out[16]:
```
[[82, 3],
 [82, 3, 1],
 [82, 3, 1, 83],
 [82, 3, 1, 83, 84],
 [82, 3, 1, 83, 84, 8],
 [82, 3, 1, 83, 84, 8, 1],
 [82, 3, 1, 83, 84, 8, 1, 85],
 [82, 3, 1, 83, 84, 8, 1, 85, 86],
 [82, 3, 1, 83, 84, 8, 1, 85, 86, 9],
 [82, 3, 1, 83, 84, 8, 1, 85, 86, 9, 87]]
```

In [23]:
```
text1 = tokenizer.sequences_to_texts([[82, 3]])
text2=  tokenizer.sequences_to_texts([[82, 3,1]])
text3=  tokenizer.sequences_to_texts([[82, 3,1,83]])
print(text1,text2,text3)
```

```
['nikhil is'] ['nikhil is a'] ['nikhil is a movie']
```

In [17]:
```
def generate_padded_sequences(input_sequences):
    max_sequence_len = max([len(x) for x in input_sequences])
    input_sequences = np.array(pad_sequences(input_sequences, maxlen=max_sequence_len,

    predictors, label = input_sequences[:,:-1],input_sequences[:,-1]
    label = ku.to_categorical(label, num_classes=total_words)
    return predictors, label, max_sequence_len

predictors, label, max_sequence_len = generate_padded_sequences(inp_sequences)
```

In [18]:
```
print(predictors, label)
```

```
[[  0   0   0 ...   0   0  82]
 [  0   0   0 ...   0  82   3]
 [  0   0   0 ...  82   3   1]
 ...
 [  0   0   0 ... 417  24 418]
 [  0   0   0 ...  24 418 419]
 [  0   0   0 ... 418 419 420]] [[0. 0. 0. ... 0. 0. 0.]
 [0. 1. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 1. 0. 0.]
 [0. 0. 0. ... 0. 1. 0.]
 [0. 0. 0. ... 0. 0. 1.]]
```

Buliding LSTM MODEL

In [21]:
```
def LSTM_model(max_sequence_len, total_words):
    input_len = max_sequence_len - 1
    model = Sequential()


    model.add(Embedding(total_words, 10, input_length=input_len))


    model.add(LSTM(100))
    model.add(Dropout(0.1))


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

    model.compile(loss='categorical_crossentropy', optimizer='adam',metrics=['accuracy

    return model
```

In [22]:
```
LSTMmodel = LSTM_model(max_sequence_len, total_words)
LSTMmodel.summary()
```

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_2 (Embedding)     (None, 31, 10)            4220

 lstm_2 (LSTM)               (None, 100)               44400

 dropout_2 (Dropout)         (None, 100)               0

 dense_2 (Dense)             (None, 422)               42622

=================================================================
Total params: 91242 (356.41 KB)
Trainable params: 91242 (356.41 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

In [23]:
```
LSTMmodel.fit(predictors, label, epochs=100, verbose=5)
```

```
Epoch 1/100
WARNING:tensorflow:From C:\Users\mpaga\AppData\Roaming\Python\Python310\site-packages
\keras\src\utils\tf_utils.py:492: The name tf.ragged.RaggedTensorValue is deprecated.
Please use tf.compat.v1.ragged.RaggedTensorValue instead.

WARNING:tensorflow:From C:\Users\mpaga\AppData\Roaming\Python\Python310\site-packages
\keras\src\engine\base_layer_utils.py:384: The name tf.executing_eagerly_outside_func
tions is deprecated. Please use tf.compat.v1.executing_eagerly_outside_functions inst
ead.

Epoch 2/100
Epoch 3/100
Epoch 4/100
Epoch 5/100
Epoch 6/100
Epoch 7/100
Epoch 8/100
Epoch 9/100
Epoch 10/100
Epoch 11/100
Epoch 12/100
Epoch 13/100
Epoch 14/100
Epoch 15/100
Epoch 16/100
Epoch 17/100
Epoch 18/100
Epoch 19/100
Epoch 20/100
Epoch 21/100
Epoch 22/100
Epoch 23/100
Epoch 24/100
Epoch 25/100
Epoch 26/100
Epoch 27/100
Epoch 28/100
Epoch 29/100
Epoch 30/100
Epoch 31/100
Epoch 32/100
Epoch 33/100
Epoch 34/100
Epoch 35/100
Epoch 36/100
Epoch 37/100
Epoch 38/100
Epoch 39/100
Epoch 40/100
Epoch 41/100
Epoch 42/100
Epoch 43/100
Epoch 44/100
Epoch 45/100
Epoch 46/100
Epoch 47/100
Epoch 48/100
Epoch 49/100
Epoch 50/100
Epoch 51/100
Epoch 52/100
Epoch 53/100
Epoch 54/100
Epoch 55/100
Epoch 56/100
Epoch 57/100
Epoch 58/100
Epoch 59/100
Epoch 60/100
Epoch 61/100
Epoch 62/100
Epoch 63/100
Epoch 64/100
Epoch 65/100
Epoch 66/100
Epoch 67/100
Epoch 68/100
Epoch 69/100
Epoch 70/100
Epoch 71/100
Epoch 72/100
Epoch 73/100
Epoch 74/100
Epoch 75/100
Epoch 76/100
Epoch 77/100
Epoch 78/100
Epoch 79/100
Epoch 80/100
Epoch 81/100
Epoch 82/100
Epoch 83/100
Epoch 84/100
Epoch 85/100
Epoch 86/100
Epoch 87/100
Epoch 88/100
Epoch 89/100
Epoch 90/100
Epoch 91/100
Epoch 92/100
Epoch 93/100
Epoch 94/100
Epoch 95/100
Epoch 96/100
Epoch 97/100
Epoch 98/100
Epoch 99/100
Epoch 100/100
```

Out[23]:

```
<keras.src.callbacks.History at 0x2491dc3d630>
```

Performance of LSTM MODEL

In [24]:

```python
loss, accuracy = LSTMmodel.evaluate(predictors, label)
print("LSTM Loss:", loss)
print("LSTM Accuracy:", accuracy)
```

```
21/21 [==============================] - 1s 12ms/step - loss: 1.2382 - accuracy: 0.82
12
LSTM Loss: 1.238184928894043
LSTM Accuracy: 0.8211624622344971
```

Generating the text using Trained(LSTM) MODEL

In [25]:
```
def generate_text(seed_text, next_words, model, max_sequence_len):
    for _ in range(next_words):
        token_list = tokenizer.texts_to_sequences([seed_text])[0]
        token_list = pad_sequences([token_list], maxlen=max_sequence_len-1, padding='p
        predicted = np.argmax(model.predict(token_list,verbose=5), axis=-1)

        output_word = ""
        for word,index in tokenizer.word_index.items():
            if index == predicted:
                output_word = word
                break
        seed_text += " "+output_word
    return seed_text.title()
```

In [34]:
```
print(generate_text("Ritika", 5, LSTMmodel, max_sequence_len))
print(generate_text("Sreeleela", 5, LSTMmodel, max_sequence_len))
print(generate_text("Isha", 5, LSTMmodel, max_sequence_len))
print(generate_text("Nikhil", 5, LSTMmodel, max_sequence_len))
```

```
Ritika A Selfproclaimed Gym Freak Has
Sreeleela Dances With Grace And Joy
Isha Is Known For Her Creativity
Nikhil Is A Selfproclaimed Gym Freak
```

BULIDING RNN MODEL

In [35]:
```
from keras.models import Sequential
from keras.layers import Embedding, SimpleRNN, Dense, Dropout

def RNN_model(max_sequence_len, total_words):
    input_len = max_sequence_len - 1
    model = Sequential()


    model.add(Embedding(total_words, 10, input_length=input_len))


    model.add(SimpleRNN(100))
    model.add(Dropout(0.1))


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

    model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accurac

    return model
```

In [36]:
```
RNNmodel = RNN_model(max_sequence_len, total_words)
RNNmodel.summary()
```

```
Model: "sequential_4"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 embedding_4 (Embedding)     (None, 31, 10)            4220      
                                                                 
 simple_rnn_1 (SimpleRNN)    (None, 100)               11100     
                                                                 
 dropout_4 (Dropout)         (None, 100)               0         
                                                                 
 dense_4 (Dense)             (None, 422)               42622     
                                                                 
=================================================================
Total params: 57942 (226.34 KB)
Trainable params: 57942 (226.34 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

In [37]:
```
RNNmodel.fit(predictors, label, epochs=100, verbose=5)
```

```
Epoch 1/100
Epoch 2/100
Epoch 3/100
Epoch 4/100
Epoch 5/100
Epoch 6/100
Epoch 7/100
Epoch 8/100
Epoch 9/100
Epoch 10/100
Epoch 11/100
Epoch 12/100
Epoch 13/100
Epoch 14/100
Epoch 15/100
Epoch 16/100
Epoch 17/100
Epoch 18/100
Epoch 19/100
Epoch 20/100
Epoch 21/100
Epoch 22/100
Epoch 23/100
Epoch 24/100
Epoch 25/100
Epoch 26/100
Epoch 27/100
Epoch 28/100
Epoch 29/100
Epoch 30/100
Epoch 31/100
Epoch 32/100
Epoch 33/100
Epoch 34/100
Epoch 35/100
Epoch 36/100
Epoch 37/100
Epoch 38/100
Epoch 39/100
Epoch 40/100
Epoch 41/100
Epoch 42/100
Epoch 43/100
Epoch 44/100
Epoch 45/100
Epoch 46/100
Epoch 47/100
Epoch 48/100
Epoch 49/100
Epoch 50/100
Epoch 51/100
Epoch 52/100
Epoch 53/100
Epoch 54/100
Epoch 55/100
Epoch 56/100
Epoch 57/100
Epoch 58/100
Epoch 59/100
Epoch 60/100
Epoch 61/100
Epoch 62/100
Epoch 63/100
Epoch 64/100
Epoch 65/100
Epoch 66/100
Epoch 67/100
Epoch 68/100
Epoch 69/100
Epoch 70/100
Epoch 71/100
Epoch 72/100
Epoch 73/100
Epoch 74/100
Epoch 75/100
Epoch 76/100
Epoch 77/100
Epoch 78/100
Epoch 79/100
Epoch 80/100
Epoch 81/100
Epoch 82/100
Epoch 83/100
Epoch 84/100
Epoch 85/100
Epoch 86/100
Epoch 87/100
Epoch 88/100
Epoch 89/100
Epoch 90/100
Epoch 91/100
Epoch 92/100
Epoch 93/100
Epoch 94/100
Epoch 95/100
Epoch 96/100
Epoch 97/100
Epoch 98/100
Epoch 99/100
Epoch 100/100
```

Out[37]:

```
<keras.src.callbacks.History at 0x2491db67400>
```

Performance Metrics of RNN Model

In [38]:

```
loss, accuracy = RNNmodel.evaluate(predictors, label)
print("RNN Loss:", loss)
print("RNN Accuracy:", accuracy)
```

```
21/21 [==============================] - 1s 6ms/step - loss: 0.1000 - accuracy: 0.998
5
RNN Loss: 0.10002611577510834
RNN Accuracy: 0.9985097050666809
```

Generating Text using Trained(RNN) Model

In [39]:
```
print(generate_text("Manohar", 5, RNNmodel, max_sequence_len))
print(generate_text("Ritika", 5, RNNmodel, max_sequence_len))
print(generate_text("Isha", 5, RNNmodel, max_sequence_len))
```

```
Manohar Is A Big Fan Of
Ritika A Selfproclaimed Gym Freak Has
Isha Is Known For Her Creativity
```

Converting IPYNB to HTML

In [40]:
```
!jupyter nbconvert --to html "DL_USECASE2.ipynb"
```

```
[NbConvertApp] Converting notebook DL_USECASE2.ipynb to html
[NbConvertApp] Writing 650083 bytes to DL_USECASE2.html
```

In [ ]:
```
```