SHRI VILEPARLE KELAVANI MANDAL'S
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC ACCREDITED with "A" GRADE (CGPA : 3.18)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE)

COURSE CODE: DJ19DSC501 **DATE:**

COURSE NAME: Machine Learning - II **CLASS: AY 2021-22**

LAB EXPERIMENT NO.5

Name: Bhuvi Ghosh **Sap ID: 60009210191** **Batch: D22**

AIM / OBJECTIVE:

Implement LSTM Sentiment Analysis on text dataset to evaluate customer reviews.

DESCRIPTION OF EXPERIMENT:

Python sentiment analysis is a methodology for analyzing a piece of text to discover the sentiment hidden within it. It accomplishes this by combining machine learning and natural language processing (NLP). Sentiment analysis allows you to examine the feelings expressed in a piece of text. It is essential for businesses to gauge customer response.

Preprocessing -

1) Normalization - Words which look different due to casing or written another way but are the same in meaning need to be process correctly. Normalisation processes ensure that these words are treated equally. For example, changing numbers to their word equivalents or converting the casing of all the text.
    a) Casing the Characters - Converting character to the same case so the same words are recognised as the same. (all lowercase)
    b) Removing - Stand alone punctuations, special characters and numerical tokens are removed as they do not contribute to sentiment which leaves only alphabetic characters. This step needs the use of tokenized words as they have been split appropriately for us to remove. We need to remove the special characters, numbers from the text. We can use the regular expression operations library of Python.

2) Tokenization - Tokenization is the process of breaking down chunks of text into smaller pieces. It converts text into tokens before transforming it into vectors. It is also easier to filter out unnecessary tokens. spaCy comes with a default processing pipeline that begins with tokenization, making this process a snap. In spaCy, you can do either sentence tokenization or word tokenization:

- Word tokenization breaks text down into individual words.
- Sentence tokenization breaks text down into individual sentences.

3) Stopwords - Stop words are the most commonly occuring words which are not relevant in the context of the data and do not contribute any deeper meaning to the phrase. In this case it contains no sentiment. We need to remove them as part of text preprocessing. nltk has a list of stopwords of every language.

4) Obtaining the stem words

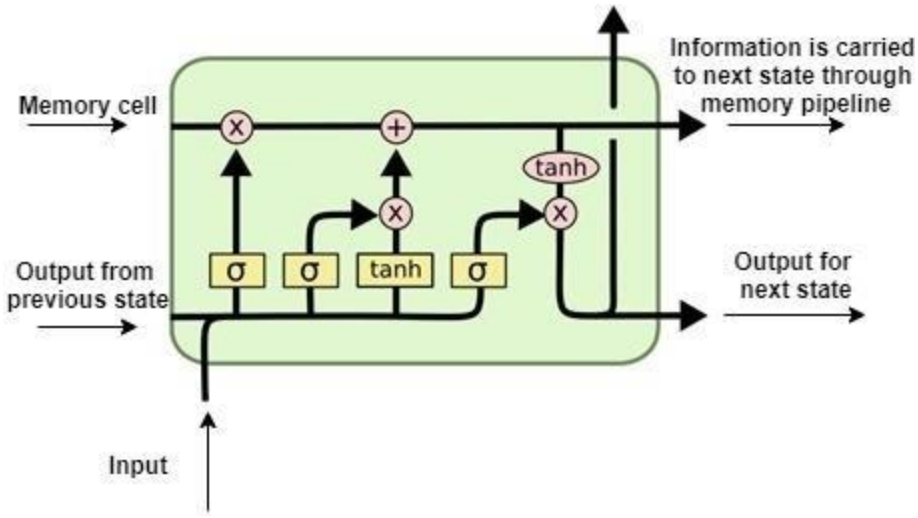
A stem is a part of a word responsible for its lexical meaning. The two popular techniques of obtaining the root/stem words are Stemming and Lemmatization

a) Stemming - Stemming is a technique used to extract the base form of the words by removing affixes from them. It is just like cutting down the branches of a tree to its stems. For example, the stem of the words eating, eats, eaten is eat.

b) Lemmitization - This process finds the base or dictionary form of the word known as the lemma. This is done through the use of vocabulary (dictionary importance of words) and morphological analysis (word structure and grammar relations)

5) Vectorization - use a count vectorizer from the Scikit-learn library to transform the text in data frame into a bag of words model, which will contain a sparse matrix of integers. The number of occurrences of each word will be counted.

Sentiment Analysis using LSTM: Use Keras



Hyperparameters to tune -

1. Layers - Explore additional hierarchical learning capacity by adding more layers and varied numbers of neurons in each layer
2. Number of inputs in dense layer - Dense layers improve overall accuracy and 5–10 units or nodes per layer is a good base
3. Dropout - Slow down learning with regularization methods like dropout on the recurrent LSTM connections.A good starting point is 20% but the dropout value should be kept small (up to 50%). The 20% value is widely accepted as the best compromise between preventing model overfitting and retaining model accuracy.

4. Learning Rate - This hyperparameter defines how quickly the network updates its parameters.
5. Decay Rate - weight decay can be added in the weight update rule that makes the weights decay to zero exponentially, if no other weight update is scheduled. After each update, the weights are multiplied by a factor slightly less than 1, thereby preventing them from growing to huge. This specifies regularization in the network.
6. Number of epochs

Workflow -

1. Preprocess data.
2. Split data into training and evaluation sets.
3. Select a model architecture.
4. Use training data to train model.
5. Use test data to evaluate the performance of model.

1. **Apply preprocessing techniques and LSTM on the dataset. Show accuracy achieved on the test dataset by providing classification report.**
2. **Perform LSTM hyper parameter tuning to improve accuracy score.**
3. **Show how LSTM model compares to built-in classifier provided by TextBlob.**

WRITEUP:

DATASET:

https://www.kaggle.com/datasets/sid321axn/amazon-alexa-reviews

This dataset consists of a nearly 3000 Amazon customer reviews (input text), star ratings, date of review, variant and feedback of various amazon Alexa products like Alexa Echo, Echo dots, Alexa Firesticks etc. for learning how to train Machine for sentiment analysis.

SOURCE CODE:

```
[ ] import pandas as pd
```

```
[ ]
    file_path = "/content/amazon_alexa.tsv"

    df = pd.read_csv(file_path, sep='\t')
```

```
[ ] df
```

	rating	date	variation	verified_reviews	feedback
0	5	31-Jul-18	Charcoal Fabric	Love my Echo!	1
1	5	31-Jul-18	Charcoal Fabric	Loved it!	1
2	4	31-Jul-18	Walnut Finish	Sometimes while playing a game, you can answer...	1
3	5	31-Jul-18	Charcoal Fabric	I have had a lot of fun with this thing. My 4 ...	1
4	5	31-Jul-18	Charcoal Fabric	Music	1
...	...	...	...	...	...
3145	5	30-Jul-18	Black Dot	Perfect for kids, adults and everyone in betwe...	1
3146	5	30-Jul-18	Black Dot	Listening to music, searching locations, check...	1
3147	5	30-Jul-18	Black Dot	I do love these things, i have them running my...	1
3148	5	30-Jul-18	White Dot	Only complaint I have is that the sound qualit...	1
3149	4	29-Jul-18	Black Dot	Good	1

3150 rows × 5 columns

```
[ ] df.isna().sum()
```

```
rating              0
date                0
variation           0
verified_reviews    0
feedback            0
dtype: int64
```

SHRI VILEPARLE KELAVANI MANDAL'S
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC ACCREDITED with "A" GRADE (CGPA : 3.18)

```
[ ] df.duplicated().sum()
```

```
715
```

```
[ ] df.drop_duplicates(inplace=True)
```

```
[ ] df.duplicated().sum()
```

```
0
```

```
[ ] !pip install nltk
```

```
Requirement already satisfied: nltk in /usr/local/lib/python3.10/dist-packages (3.8.1)
Requirement already satisfied: click in /usr/local/lib/python3.10/dist-packages (from nltk) (8.1.7)
Requirement already satisfied: joblib in /usr/local/lib/python3.10/dist-packages (from nltk) (1.3.2)
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.10/dist-packages (from nltk) (2023.6.3)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from nltk) (4.66.1)
```

```
[ ] import nltk
```

```
[ ] nltk.download("stopwords")
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
True
```

```
[ ] from nltk.corpus import stopwords
```

```
[ ] stopwords.words('english')

    import string
    string.punctuation
```

```
'!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~'
```

```
[ ] nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
True
```

```
[ ] from nltk.stem.porter import PorterStemmer
    ps = PorterStemmer()
    ps.stem('loving')
```

```
'love'
```

```
[ ] nltk.download("wordnet")
```

```
[nltk_data] Downloading package wordnet to /root/nltk_data...
True
```

```
[ ] from nltk.stem import WordNetLemmatizer
    lm = WordNetLemmatizer()
```

SHRI VILEPARLE KELAVANI MANDAL'S
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC ACCREDITED with "A" GRADE (CGPA : 3.18)

```python
def transform_text(text):
    try:
        text = text.lower()
        text = nltk.word_tokenize(text)

        y = []
        for i in text:
            if i.isalnum():
                y.append(i)

        text = y[:]
        y.clear()

        for i in text:
            if i not in stopwords.words('english') and i not in string.punctuation:
                y.append(i)

        text = y[:]
        y.clear()

        for i in text:
            y.append(lm.lemmatize(i))

        return " ".join(y)

    except AttributeError as e:
        print("Error in processing text:", e)
        print("Current value of text variable:", text)
        return None  # Return None to handle this case in your DataFrame
```

```python
df['transformed_reviews'] = df['verified_reviews'].apply(transform_text)
# df['transformed_content'] = df['Content'].apply(transform_text)
```

```python
# df['transformed_text'] = df['transformed_title'] + ' ' + df['transformed_content']

df = df.drop(['verified_reviews'], axis=1)
```

```python
df.head()
```

	rating	date	variation	feedback	transformed_reviews
0	5	31-Jul-18	Charcoal Fabric	1	love echo
1	5	31-Jul-18	Charcoal Fabric	1	loved
2	4	31-Jul-18	Walnut Finish	1	sometimes playing game answer question correct...
3	5	31-Jul-18	Charcoal Fabric	1	lot fun thing 4 yr old learns dinosaur control...
4	5	31-Jul-18	Charcoal Fabric	1	music

```python
# import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
```

```python
tfidf_vectorizer = TfidfVectorizer()
X = tfidf_vectorizer.fit_transform(df['transformed_reviews'])
```

```python
y = df['variation']
```

```python
# X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```python
from sklearn.metrics import classification_report,confusion_matrix,accuracy_score
from keras.models import Sequential
```

SHRI VILEPARLE KELAVANI MANDAL'S
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC ACCREDITED with "A" GRADE (CGPA : 3.18)

```
[ ] from sklearn.metrics import classification_report,confusion_matrix,accuracy_score
    from keras.models import Sequential
    from keras.preprocessing.text import Tokenizer
    from keras.preprocessing.sequence import pad_sequences
    from keras.layers import Dense, Embedding, LSTM, SpatialDropout1D
```

```
def sentiments(df):
  if df['rating'] > 3.0:
    return 'Positive'
  elif df['rating'] <= 3.0:
    return 'Negative'
df['sentiment'] = df.apply(sentiments, axis=1)
```

```
df.head(2)
```

	rating	date	variation	feedback	transformed_reviews	sentiment
0	5	31-Jul-18	Charcoal Fabric	1	love echo	Positive
1	5	31-Jul-18	Charcoal Fabric	1	loved	Positive

```
[ ] from sklearn.preprocessing import LabelEncoder
```

```
[ ]
    lb=LabelEncoder()
    df['sentiment'] = lb.fit_transform(df['sentiment'])
```

```
[ ] df.head(2)
```

```
[ ] df.head(2)
```

	rating	date	variation	feedback	transformed_reviews	sentiment
0	5	31-Jul-18	Charcoal Fabric	1	love echo	1
1	5	31-Jul-18	Charcoal Fabric	1	loved	1

```
[ ] import tensorflow as tf
    from tensorflow import keras
    from tensorflow.keras.layers import Embedding, LSTM, Dense
    from tensorflow.keras.preprocessing.text import Tokenizer
    from tensorflow.keras.preprocessing.sequence import pad_sequences
```

```
    tokenizer = Tokenizer(num_words=10000)
    tokenizer.fit_on_texts(df['transformed_reviews'])

    sequences = tokenizer.texts_to_sequences(df['transformed_reviews'])

    max_length = 100
    padded_sequences = pad_sequences(sequences, maxlen=max_length, padding='post')
```

```
[ ] X = padded_sequences
    y = df['sentiment']

    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

SHRI VILEPARLE KELAVANI MANDAL'S
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC ACCREDITED with "A" GRADE (CGPA : 3.18)

```
[ ] model = keras.Sequential()
    model.add(Embedding(input_dim=10000, output_dim=64, input_length=max_length))
    model.add(LSTM(64))
    model.add(Dense(1, activation='tanh'))
    model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
```

```
    model.fit(X_train, y_train, epochs=5, batch_size=64)
```

```
Epoch 1/5
31/31 [==============================] - 5s 85ms/step - loss: 0.8925 - accuracy: 0.6011
Epoch 2/5
31/31 [==============================] - 3s 94ms/step - loss: 0.4004 - accuracy: 0.8650
Epoch 3/5
31/31 [==============================] - 4s 112ms/step - loss: 0.3942 - accuracy: 0.8670
Epoch 4/5
31/31 [==============================] - 3s 86ms/step - loss: 0.3917 - accuracy: 0.8676
Epoch 5/5
31/31 [==============================] - 3s 85ms/step - loss: 0.3917 - accuracy: 0.8676
<keras.src.callbacks.History at 0x7efee6f02440>
```

```
[ ] test_loss, test_accuracy = model.evaluate(X_test, y_test)
    print(f'Test accuracy: {test_accuracy}')
```

```
16/16 [==============================] - 1s 17ms/step - loss: 0.4089 - accuracy: 0.8563
Test accuracy: 0.8562628626823425
```

```
[ ] new_reviews = ["This is a great product.", "I didn't like it."]
    new_sequences = tokenizer.texts_to_sequences(new_reviews)
    new_padded_sequences = pad_sequences(new_sequences, maxlen=max_length, padding='post')
    predictions = model.predict(new_padded_sequences)
```

```
1/1 [==============================] - 0s 430ms/step

[ ] predictions

array([[0.8618436],
       [0.8618436 ]], dtype=float32)

[ ]
```