

```
In [1]: # import the libraries
import pandas as pd
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from keras.models import Sequential
from keras.layers import Dense, Embedding, LSTM
#from keras.utils import to_categorical
#this import gave me a "ModuleNotFoundError: No module named 'keras.utils.np_utils'"
#After some search online, I changed the code to the following. It was due to an update
from tensorflow.python.keras.utils.np_utils import to_categorical
from sklearn.model_selection import train_test_split
```

C:\Users\chtan\anaconda3\lib\site-packages\scipy\\_\_init\_\_.py:155: UserWarning: A NumPy version >=1.18.5 and <1.25.0 is required for this version of SciPy (detected version 1.26.1)
... warnings.warn(f"A NumPy version >={np\_minversion} and <{np\_maxversion}"

```
In [2]: # Load the data that I've downloaded from the provided Kaggle page
df = pd.read_csv('Womens Clothing E-Commerce Reviews.csv')
```

```
In [3]: # Analyzing data
df.head()
```

	Unnamed: 0	Clothing ID	Age	Title	Review Text	Rating	Recommended IND	Positive Feedback Count	Division Name	Depa
0	0	767	33	NaN	Absolutely wonderful - silky and sexy and comf...	4	1	0	Initmates	I
1	1	1080	34	NaN	Love this dress! it's sooo pretty. i happen...	5	1	4	General	E
2	2	1077	60	Some major design flaws	I had such high hopes for this dress and reall...	3	0	0	General	B
3	3	1049	50	My favorite buy!	I love, love, love this jumpsuit. it's fun, fl...	5	1	0	General Petite	E
4	4	847	47	Flattering shirt	This shirt is very flattering to all due to th...	5	1	6	General	B

```
In [4]: df.isnull().sum()
```

```
Out[4]: Unnamed: 0 ..... 0
Clothing ID ..... 0
Age ..... 0
Title ..... 3810
Review Text ..... 845
Rating ..... 0
Recommended IND ..... 0
Positive Feedback Count ..... 0
Division Name ..... 14
Department Name ..... 14
Class Name ..... 14
dtype: int64
```

```
In [5]: # Removing irrelevant columns
df = df.drop(['Title', 'Unnamed: 0', 'Recommended IND', 'Positive Feedback Count', 'Clothing ID'])
df.head()
```

```
Out[5]:
```

	Review Text	Rating
<b>0</b>	Absolutely wonderful - silky and sexy and comf...	4
<b>1</b>	Love this dress! it's sooo pretty. i happen...	5
<b>2</b>	I had such high hopes for this dress and reall...	3
<b>3</b>	I love, love, love this jumpsuit. it's fun, fl...	5
<b>4</b>	This shirt is very flattering to all due to th...	5

	Review Text	Rating
<b>0</b>	Absolutely wonderful - silky and sexy and comf...	4
<b>1</b>	Love this dress! it's sooo pretty. i happen...	5
<b>2</b>	I had such high hopes for this dress and reall...	3
<b>3</b>	I love, love, love this jumpsuit. it's fun, fl...	5
<b>4</b>	This shirt is very flattering to all due to th...	5

```
In [6]: df = df.dropna(subset=['Review Text'])
df.isnull().sum()
```

```
Out[6]: Review Text ... 0
Rating ..... 0
dtype: int64
```

```
In [7]: # Lowercase the text
df['Review Text'] = df['Review Text'].str.lower()
df.head()
```

```
Out[7]:
```

	Review Text	Rating
<b>0</b>	absolutely wonderful - silky and sexy and comf...	4
<b>1</b>	love this dress! it's sooo pretty. i happen...	5
<b>2</b>	i had such high hopes for this dress and reall...	3
<b>3</b>	i love, love, love this jumpsuit. it's fun, fl...	5
<b>4</b>	this shirt is very flattering to all due to th...	5

	Review Text	Rating
<b>0</b>	absolutely wonderful - silky and sexy and comf...	4
<b>1</b>	love this dress! it's sooo pretty. i happen...	5
<b>2</b>	i had such high hopes for this dress and reall...	3
<b>3</b>	i love, love, love this jumpsuit. it's fun, fl...	5
<b>4</b>	this shirt is very flattering to all due to th...	5

```
In [8]: # Remove the special characters
df['Review Text'] = df['Review Text'].str.replace('[^\w\s]', '')
df.head()
```

```
C:\Users\chtan\AppData\Local\Temp\ipykernel_36256\2518602467.py:2: FutureWarning: The default value of regex will change from True to False in a future version.  
... df['Review Text'] = df['Review Text'].str.replace('^\w\s+', '')
```

Out[8]:

	Review Text	Rating
0	absolutely wonderful silky and sexy and comfo...	4
1	love this dress its sooo pretty i happened t...	5
2	i had such high hopes for this dress and reall...	3
3	i love love love this jumpsuit its fun flirty ...	5
4	this shirt is very flattering to all due to th...	5

In [9]:

```
def classify_sentiment(rating):  
    if rating <= 2:  
        return 'Negative'  
    elif rating == 3:  
        return 'Neutral'  
    else:  
        return 'Positive'  
  
df['Sentiment'] = df['Rating'].apply(classify_sentiment)  
df.head()
```

Out[9]:

	Review Text	Rating	Sentiment
0	absolutely wonderful silky and sexy and comfo...	4	Positive
1	love this dress its sooo pretty i happened t...	5	Positive
2	i had such high hopes for this dress and reall...	3	Neutral
3	i love love love this jumpsuit its fun flirty ...	5	Positive
4	this shirt is very flattering to all due to th...	5	Positive

In [11]:

```
# Tokenize  
tokenizer = Tokenizer(num_words=5000, split=" ")  
tokenizer.fit_on_texts(df['Review Text'].values)  
  
X = tokenizer.texts_to_sequences(df['Review Text'].values)  
# padding text vector so they all have the same length  
X = pad_sequences(X)  
y = pd.get_dummies(df['Sentiment']).values
```

In [12]:

```
# Splitting data into training and testing sets  
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=4)
```

In [13]:

```
# Building model  
model = Sequential()  
model.add(Embedding(5000, 256, input_length=X.shape[1]))  
model.add(LSTM(256, dropout=0.2, recurrent_dropout=0.2))  
model.add(Dense(3, activation='softmax')) # 3 classes: 'Positive', 'Neutral', 'Negative'
```

In [14]:

```
# Compiling model  
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
```

```
In [15]: # Training model  
model.fit(X_train, y_train, epochs=5, batch_size=32, verbose=2)
```

```
Epoch 1/5  
566/566 - 266s - loss: 0.5196 - accuracy: 0.7961 - 266s/epoch - 470ms/step  
Epoch 2/5  
566/566 - 266s - loss: 0.3987 - accuracy: 0.8354 - 266s/epoch - 470ms/step  
Epoch 3/5  
566/566 - 268s - loss: 0.3384 - accuracy: 0.8628 - 268s/epoch - 474ms/step  
Epoch 4/5  
566/566 - 265s - loss: 0.2824 - accuracy: 0.8868 - 265s/epoch - 469ms/step  
Epoch 5/5  
566/566 - 266s - loss: 0.2344 - accuracy: 0.9080 - 266s/epoch - 469ms/step  
<keras.src.callbacks.History at 0x1d252c25cd0>
```

Out[15]:

```
In [16]: # Evaluating model  
score, acc = model.evaluate(X_test, y_test, verbose=2)  
print("Model Accuracy: %.2f%%" % (acc*100))
```

```
142/142 - 8s - loss: 0.5825 - accuracy: 0.8106 - 8s/epoch - 60ms/step  
Model Accuracy: 81.06%
```

In [17]: # Function to predict sentiment of a review

```
def predict_sentiment(review):  
    review_seq = tokenizer.texts_to_sequences([review])  
    review_padded = pad_sequences(review_seq, maxlen=X.shape[1])  
    prediction = model.predict_classes(review_padded)  
    sentiments = ['Positive', 'Neutral', 'Negative']  
    return sentiments[prediction[0]]
```