Embeding

```
import nltk


paragraph = """AI, machine learning and deep learning are common terms in enterprise
              IT and sometimes used interchangeably, especially by companies in their marketing materials.
              But there are distinctions. The term AI, coined in the 1950s, refers to the simulation of human
              intelligence by machines. It covers an ever-changing set of capabilities as new technologies
              are developed. Technologies that come under the umbrella of AI include machine learning and
              deep learning. Machine learning enables software applications to become more accurate at
              predicting outcomes without being explicitly programmed to do so. Machine learning algorithms
              use historical data as input to predict new output values. This approach became vastly more
              effective with the rise of large data sets to train on. Deep learning, a subset of machine
              learning, is based on our understanding of how the brain is structured. Deep learning's
              use of artificial neural networks structure is the underpinning of recent advances in AI,
              including self-driving cars and ChatGPT."""



# Clearning the texts
import re # re libray will use for regular expression
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from nltk.stem import WordNetLemmatizer

ps = PorterStemmer()
wordnet=WordNetLemmatizer()
sentences = nltk.sent_tokenize(paragraph)

corpus = []

# Create the empty list name as corpus because after cleaned the data corpus will store
this

for i in range(len(sentences)):
    review = re.sub('[^a-zA-Z]', ' ', sentences[i])
    review = review.lower()
    review = review.split()
    review = [ps.stem(word) for word in review if not word in
set(stopwords.words('english'))]
    review = [wordnet.lemmatize(word) for word in review if not word in
set(stopwords.words('english'))]
    review = ' '.join(review)
    corpus.append(review)

# Creating the Bag of words model

# Also we called as document matrix
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer()
x_bow = cv.fit_transform(corpus).toarray()
```

```
from sklearn.feature_extraction.text import TfidfVectorizer

tf = TfidfVectorizer()

x_tf = tf.fit_transform(corpus).toarray()
```