In [1]:
```
import nltk
nltk.download('punkt')
nltk.download('wordnet')
nltk.download('averaged_perceptron_tagger')
nltk.download('stopwords')
from nltk import sent_tokenize
from nltk import word_tokenize
from nltk.corpus import stopwords
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\Aniket\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping tokenizers\punkt.zip.
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\Aniket\AppData\Roaming\nltk_data...
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     C:\Users\Aniket\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping taggers\averaged_perceptron_tagger.zip.
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\Aniket\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping corpora\stopwords.zip.
```

In [2]:
```
text='Real madrid is set to win the UCL for the season . Benzema might win
```

In [3]:
```
tokens_sents = nltk.sent_tokenize(text)
print(tokens_sents)
```

```
['Real madrid is set to win the UCL for the season .', 'Benzema might win
Balon dor .', 'Salah might be the runner up']
```

In [4]:
```
tokens_words = nltk.word_tokenize(text)
print(tokens_words)
```

```
['Real', 'madrid', 'is', 'set', 'to', 'win', 'the', 'UCL', 'for', 'the',
'season', '.', 'Benzema', 'might', 'win', 'Balon', 'dor', '.', 'Salah', 'm
ight', 'be', 'the', 'runner', 'up']
```

In [5]:
```
from nltk.stem import PorterStemmer
from nltk.stem.snowball import SnowballStemmer
from nltk.stem import LancasterStemmer
```

In [6]:
```
stem=[]
for i in tokens_words:
  ps = PorterStemmer()
  stem_word= ps.stem(i)
  stem.append(stem_word)
print(stem)
```

```
['real', 'madrid', 'is', 'set', 'to', 'win', 'the', 'ucl', 'for', 'the',
'season', '.', 'benzema', 'might', 'win', 'balon', 'dor', '.', 'salah', 'm
ight', 'be', 'the', 'runner', 'up']
```

In [7]:
```
import nltk
from nltk.stem import WordNetLemmatizer
lemmatizer = WordNetLemmatizer()
lemmatized_output = ' '.join([lemmatizer.lemmatize(w) for w in stem])
print(lemmatized_output)
```

```
real madrid is set to win the ucl for the season . benzema might win balon
dor . salah might be the runner up
```

In [8]:
```
leme=[]
for i in stem:
  lemetized_word=lemmatizer.lemmatize(i)
  leme.append(lemetized_word)
print(leme)
```

```
['real', 'madrid', 'is', 'set', 'to', 'win', 'the', 'ucl', 'for', 'the',
'season', '.', 'benzema', 'might', 'win', 'balon', 'dor', '.', 'salah', 'm
ight', 'be', 'the', 'runner', 'up']
```

In [9]:
```
print("Parts of Speech: ",nltk.pos_tag(leme))
```

```
Parts of Speech:  [('real', 'JJ'), ('madrid', 'NN'), ('is', 'VBZ'), ('se
t', 'VBN'), ('to', 'TO'), ('win', 'VB'), ('the', 'DT'), ('ucl', 'NN'), ('f
or', 'IN'), ('the', 'DT'), ('season', 'NN'), ('.', '.'), ('benzema', 'N
N'), ('might', 'MD'), ('win', 'VB'), ('balon', 'NN'), ('dor', 'NN'), ('.',
'.'), ('salah', 'NN'), ('might', 'MD'), ('be', 'VB'), ('the', 'DT'), ('run
ner', 'NN'), ('up', 'RP')]
```

In [10]:
```
sw_nltk = stopwords.words('english')
print(sw_nltk)
```

```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "yo
u're", "you've", "you'll", "you'd", 'your', 'yours', 'yourself', 'yourselv
es', 'he', 'him', 'his', 'himself', 'she', "she's", 'her', 'hers', 'hersel
f', 'it', "it's", 'its', 'itself', 'they', 'them', 'their', 'theirs', 'the
mselves', 'what', 'which', 'who', 'whom', 'this', 'that', "that'll", 'thes
e', 'those', 'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'hav
e', 'has', 'had', 'having', 'do', 'does', 'did', 'doing', 'a', 'an', 'th
e', 'and', 'but', 'if', 'or', 'because', 'as', 'until', 'while', 'of', 'a
t', 'by', 'for', 'with', 'about', 'against', 'between', 'into', 'through',
'during', 'before', 'after', 'above', 'below', 'to', 'from', 'up', 'down',
'in', 'out', 'on', 'off', 'over', 'under', 'again', 'further', 'then', 'on
ce', 'here', 'there', 'when', 'where', 'why', 'how', 'all', 'any', 'both',
'each', 'few', 'more', 'most', 'other', 'some', 'such', 'no', 'nor', 'no
t', 'only', 'own', 'same', 'so', 'than', 'too', 'very', 's', 't', 'can',
'will', 'just', 'don', "don't", 'should', "should've", 'now', 'd', 'll',
'm', 'o', 're', 've', 'y', 'ain', 'aren', "aren't", 'couldn', "couldn't",
'didn', "didn't", 'doesn', "doesn't", 'hadn', "hadn't", 'hasn', "hasn't",
'haven', "haven't", 'isn', "isn't", 'ma', 'mightn', "mightn't", 'mustn',
"mustn't", 'needn', "needn't", 'shan', "shan't", 'shouldn', "shouldn't",
'wasn', "wasn't", 'weren', "weren't", 'won', "won't", 'wouldn', "would
n't"]
```

In [12]:
```
words = [word for word in text.split() if word.lower() not in sw_nltk]
new_text = " ".join(words)
print(new_text)
```

```
Real madrid set win UCL season . Benzema might win Balon dor . Salah might
runner
```

In [ ]: