NLP

1] Text Preprocessing

To prepare the text data for the model building we perform text preprocessing. It is the very first step of NLP projects.

Remove HTML Tags
HTML tags are not important in model building. We have to remove HTML tags.
For removing HTML tags we use regex.

```
import re
def remove_html_tags(text):
    pattern = re.compile('')
    return pattern.sub(r'', text)
df['text'].apply(remove_html_tags)
```

Remove URLs
URLs are not important in model building. We have to remove URLs.
For removing URLs we can use regex.

```
def remove_url(text):
        pattern = re.compile('https?://S+|www.S+')
        return pattern.sub(r'', text)
df['text'].apply(remove_url)
```

Punctuation Removal:
In this step, all the punctuations from the text are removed.If we did not remove punctuation then punctuation is also considered one word . string library of Python contains some pre-defined list of punctuations such as ‘!”#$
%&'()*+,-./:;?@[\]^_`{|}~’

Lowering the text:
It is one of the most common preprocessing steps where the text is converted into the same case preferably lower case.
Tokenization:
In this step, the text is split into smaller units. We can use either sentence tokenization or word tokenization based on our problem statement. Sentence tokenisation breaks the sentence into word tokens and word tokenisation breaks the word into individual letter tokens. Tokenization can be done using NLTK Library

Stop word removal:
Stopwords are the commonly used words and are removed from the text as they do not add any value to the analysis. These words carry less or no meaning.
NLTK library consists of a list of words that are considered stopwords for the English language. Some of them are : [i, me, my, myself, we, our, ours,

ourselves, you, you’re, you’ve, you’ll, you’d, your, yours, yourself,etc]

```
import nltk
                                                  (importing nlp library)
stopwords = nltk.corpus.stopwords.words('english')
                                      (Stop words present in the library)
def remove_stopwords(text):
                                      (defining the function to remove stopwords from
tokenized text)
   output= [i for i in text if i not in stopwords]
   return output
data['no_stopwords']= data['msg_tokenied'].apply(lambda
x:remove_stopwords(x))                            (applying the function)
```

Stemming:
It is also known as the text standardization step where the words are stemmed or diminished to their root/base form. **For example**, words like ‘programmer’, ‘programming, ‘program’ will be stemmed to ‘program’.
But the **disadvantage** of stemming is that it stems the words such that its root form loses the meaning or it is not diminished to a proper English word.
crazy-> crazi
available-> avail
entry-> entri
early-> earli

Lemmatization:
It stems the word but makes sure that it does not lose its meaning. Lemmatization has a pre-defined dictionary that stores the context of words and checks the word in the dictionary while diminishing.
In Lemmatization we search words in wordnet.

```
nltk.download(‘wordnet’)
from nltk.stem import WordNetLemmatizer
WordNetLemmatizer = WordNetLemmatizer()
sent = ‘History is the best subject for teaching’
tokens = nltk.word_tokenize(sent)
for word in tokens:
print(word,’—->’, WordNetLemmatizer.lemmatize(word, pos=’v’))
```

The difference between Stemming and Lemmatization can be understood with the example provided

Original Word	After Stemming	After Lemmatization
goose	goos	goose
geese	gees	goose

After all the text processing steps are performed, the final acquired data is converted into the numeric form using Bag of words or TF-IDF.

2] Featured Engineering

After the initial text is cleaned, we need to transform it into its features to be used for modeling. Document data is not computable so it must be transformed into numerical data such as a vector space model. This transformation task is generally called text vectorization of document data.

Corpus(c) — The total number of words present in the whole dataset is known as Corpus.
Vocabulary (V) — Total number of unique words available in the corpus.
Document (D) — There are multiple records in a dataset so a single record or review is referred to as a document.
Word (w) — Words that are used in a document are known as Word.

Bag of Words

It is one of the most used text vectorization techniques. A bag-of-words is a representation of text that describes the occurrence of words within a document.
Specially used in the Text Classification task.
We can directly use CountVectorizer class by Scikit-learn.

```
cV = CountVectorizer()
bow = cv.fit transform(df['text'])
print(sorted(cv.vocabulary_))
print (bow[0]. toarray())
```

3] Modelling/Model Building

In the modeling step, we try to make a model based on data. here also we can use multiple approaches to build the model based on the problem statement.

4] Model Evaluation

Intrinsic evaluation – In this evaluation, we use multiple metrics to check our model such as Accuracy, Recall, Confusion Metrics, Perplexity, etc.