POS Tagging

1) With NLTK

Input –

```
import nltk
from nltk.tokenize import word_tokenize
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')

def pos_tagger(text):
    words = word_tokenize(text)

    tagged_words = nltk.pos_tag(words)

    return tagged_words

sent = '''Two roads diverged in a yellow wood,
And sorry I could not travel both
And be one traveler, long I stood
And looked down one as far as I could
To where it bent in the undergrowth'''

tags = pos_tagger(sent)
print(tags)
```

Output –

```
[('Two', 'CD'), ('roads', 'NNS'), ('diverged', 'VBN'), ('in', 'IN'), ('a', 'DT'), ('yellow', 'JJ'), ('wo
od', 'NN'), (',', ','), ('And', 'CC'), ('sorry', 'NN'), ('I', 'PRP'), ('could', 'MD'), ('not', 'RB'),
('travel', 'VB'), ('both', 'DT'), ('And', 'CC'), ('be', 'VB'), ('one', 'CD'), ('traveler', 'NN'), (',',
','), ('long', 'RB'), ('I', 'PRP'), ('stood', 'VBD'), ('And', 'CC'), ('looked', 'VBD'), ('down', 'RB'),
('one', 'CD'), ('as', 'RB'), ('far', 'RB'), ('as', 'IN'), ('I', 'PRP'), ('could', 'MD'), ('To', 'TO'),
('where', 'WRB'), ('it', 'PRP'), ('bent', 'VBD'), ('in', 'IN'), ('the', 'DT'), ('undergrowth', 'NN')]
```

2) Without NLTK

Input –

```
def pos_tag(text):
    words = text.split()
    tagged_words = []

    for word in words:
        if word.endswith("ing"):
            pos = "VBG" # Verb, Gerund/Present Participle
        elif word.lower() in ["i", "me", "my", "mine", "myself", "you", "your", "yours", "yourself", "he", "him", "his", "himself",
                              "she", "her", "hers", "herself", "it", "its", "itself", "we", "us", "our", "ours", "ourselves",
                              "they", "them", "their", "theirs", "themselves"]:
            pos = "PRP"
        elif word.endswith("ed"):
            pos = "VBD" # Verb, Past Tense
        elif word.lower() in ["a", "an", "the"]:
            pos = "DT"
        elif word.isnumeric():
            pos = "CD" # Cardinal Digit
        elif word.endswith("er") or word.endswith("est"):
            pos = "JJ" #adjective
        else:
            pos = "NN" # Noun, singular or mass (default)

        tagged_words.append((word, pos))

    return tagged_words
```

```
text = '''Then took the other, as just as fair,
And having perhaps the better claim,
Because it was grassy and wanted wear;
Though as for that the passing there
Had worn them really about the same'''


#text = "I saw her at the park. She said it was hers."

tagged_text = pos_tag(text)

for word, pos in tagged_text:
    print(f"{word}/{pos}", end=" ")
```

Output –

```
Then/NN took/NN the/DT other,/NN as/NN just/NN as/NN fair,/NN And/NN having/VBG perhaps/NN the/DT better/JJ claim,/NN Because/NN it/PRP
was/NN grassy/NN and/NN wanted/VBD wear;/NN Though/NN as/NN for/NN that/NN the/DT passing/VBG there/NN Had/NN worn/NN them/PRP really/N
N about/NN the/DT same/NN
```

Repo Link

https://github.com/Shreyaww/Sem7_NLP/blob/main/Experiment%206.ipynb