1) POS Tagging (Part of Speech Tagging)

🔥 What is POS Tagging?

POS Tagging means identifying the **grammar role** of each word in a sentence.

It tells whether a word is:

- Noun
- Verb
- Adjective
- Pronoun
- Preposition
- etc.

📌 Example:
Sentence: **"Narendra Modi visited Delhi"**

POS Tagging result:

- Narendra → Proper Noun
- Modi → Proper Noun
- visited → Verb (past tense)
- Delhi → Proper Noun

✅ Why do we use POS Tagging in NLP?

⭐ 1. To understand sentence meaning

A word can have different meanings depending on its role.

Example:

- **"I book a ticket"** → book = Verb
- **"This is a book"** → book = Noun

POS Tagging helps NLP models understand the correct meaning.

⭐ 2. Used in Grammar Checking

Tools like Grammarly use POS tagging to detect grammar mistakes.

⭐ 3. Helpful in Machine Translation

For translating English to Hindi/French etc., POS tagging helps maintain correct grammar structure.

⭐ 4. Used in Chatbots and Question Answering

Chatbots need to identify actions and objects.

Example:
"Show me hotels in Delhi"

- Show = Verb (action)
- hotels = Noun (object)
- Delhi = Location

⭐ 5. Used in Text Summarization

POS tagging helps find important words (mostly nouns and verbs) to generate summaries.

✅ 2) NER (Named Entity Recognition)

🔥 What is NER?

NER stands for **Named Entity Recognition**.

It is used to identify **real-world names/entities** in text such as:

- Person Name
- Location
- Organization
- Date
- Time
- Money
- Country
- Product

📌 Example:
Sentence: **"Narendra Modi visited Delhi and met Elon Musk"**

NER output:

- Narendra Modi → PERSON
- Delhi → LOCATION
- Elon Musk → PERSON

✓ Why do we use NER in NLP?

⭐ 1. Information Extraction

NER helps extract important information from text.

Example:
"Apple launched iPhone in India on 5 Feb"

NER will identify:

- Apple → Organization
- iPhone → Product
- India → Location
- 5 Feb → Date

⭐ 2. Used in Resume Parsing

Companies use NER to extract:

- Name
- Skills
- Company Names
- Education details

⭐ 3. Used in Search Engines

Google uses NER to understand search queries.

Example:
"Modi Delhi visit"
It identifies Modi as PERSON and Delhi as LOCATION.

⭐ 4. Used in Chatbots and Voice Assistants

Example:
"Book a flight to Mumbai tomorrow"

NER extracts:

- Mumbai → Location
- tomorrow → Date

� � POS Tagging Code

import nltk from nltk.tokenize

import word_tokenize

nltk.download("punkt") nltk.download("averaged_perceptron_tagger")

sentence = "I love Artificial Intelligence and Machine Learning"

tokens = word_tokenize(sentence)

pos_tags = nltk.pos_tag(tokens) print(pos_tags)

✅ NER kya hota hai? (Theory)

NER = sentence me se entities identify karna:

- Person
- Location
- Organization
- Date

Example:
"Narendra Modi visited Delhi"
Output:

- Narendra Modi → PERSON
- Delhi → LOCATION

📌 Use:

- News analysis
- Resume parser
- AI assistant

🔥 NER Code + Output

import nltk

```
from nltk.tokenize import word_tokenize
from nltk import pos_tag, ne_chunk

nltk.download("punkt")
nltk.download("maxent_ne_chunker")
nltk.download("words")
nltk.download("averaged_perceptron_tagger")

sentence = "Narendra Modi visited Delhi and met Elon Musk"

tokens = word_tokenize(sentence)

tags = pos_tag(tokens)
entities = ne_chunk(tags)
print(entities)
```

✅ Output (example):

```
  (PERSON Narendra/NNP Modi/NNP)  visited/VBD (GPE Delhi/NNP) and/CC
met/VBD
  (PERSON Elon/NNP Musk/NNP))
```

PROJECT:

🔥 Chatbot

```
def chatbot(user_input):
    user_input = user_input.lower()
    if "hello" in user_input or "hi" in user_input:
        return "Hello! ☺ How can I help you?"

    elif "your name" in user_input:
        return "I am an AI Chatbot created using Python NLP."

    elif "what is ai" in user_input:
        return "AI means Artificial Intelligence. It is used to make machines smart."

    elif "what is nlp" in user_input:
        return "NLP means Natural Language Processing. It helps computers understand human language."

    elif "bye" in user_input or "exit" in user_input:
        return "Bye! Have a great day ☺"

    else:
        return "Sorry, I don't understand. Please ask something else."

print("Chatbot Started (type 'bye' to exit)")

while True:
    msg = input("\nYou: ")
    reply = chatbot(msg)
    print("Bot:", reply)
    if "bye" in msg.lower() or "exit" in msg.lower():
        break
```

SPAM EMAIL CLASSIFIER)

```
import pandas as pd
import nltk
import string

from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords

from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB

nltk.download("punkt")
nltk.download("stopwords")

# ----------------------------
# Dataset
# ----------------------------
data = {
    "text": [
        "Win money now!!! Click here",
        "Congratulations you have won a prize",
        "Free recharge offer available",
        "Call me when you are free",
        "Let's meet tomorrow at 5 pm",
        "Your bank account is hacked click this link",
        "Hello friend how are you",
        "Important meeting tomorrow",
        "Claim your lottery reward now",
        "Are you coming to class today"
    ],
    "label": [1,1,1,0,0,1,0,0,1,0]   # 1=Spam, 0=Not Spam
}

df = pd.DataFrame(data)

# ----------------------------
# Text Cleaning
# ----------------------------
def clean_text(text):
    text = text.lower()
    tokens = word_tokenize(text)
    tokens = [word for word in tokens if word not in string.punctuation]
    stop_words = stopwords.words("english")
    tokens = [word for word in tokens if word not in stop_words]
    return " ".join(tokens)
```

```
df["clean_text"] = df["text"].apply(clean_text)


# -----------------------------
# TF-IDF Vectorization
# -----------------------------
vectorizer = TfidfVectorizer()
X = vectorizer.fit_transform(df["clean_text"])
y = df["label"]


# -----------------------------
# Train Model
# -----------------------------
model = MultinomialNB()
model.fit(X, y)
# -----------------------------
# User Input Prediction
# -----------------------------
user_msg = input("\nEnter message: ")

clean_user = clean_text(user_msg)
user_vector = vectorizer.transform([clean_user])

prediction = model.predict(user_vector)

print("\nYour Message:", user_msg)
```

```
if prediction[0] == 1:
    print("Prediction: SPAM ⊘")
else:
    print("Prediction: NOT SPAM ✅")
```