1. Perform all the preprocessing steps for data tokenisation, normalisation, and stopword removal in the following text. Further determine the type/token ratio, and rank the words based on their frequencies

> "Hi there! Did you hear about the new café that just opened downtown?" asked Sarah. "No, I haven't. What's so special about it?" replied John, looking curious. "It's supposed to have the best espresso in town—people are raving about it!" Sarah responded enthusiastically. "Really? I’m always on the lookout for good coffee. When should we check it out?" John asked. "How about this Saturday? I heard they have a live jazz band in the evening," Sarah suggested. "That sounds fantastic! I’m in. Let’s meet there at 7 PM, then," John agreed. "Perfect! I'll text you the address later. By the way, did you see the new movie trailer for 'The Last Horizon'? It looks amazing, don't you think?" Sarah inquired. "Oh, I did! It looks intense. I can’t wait to see it," John said with excitement.

Solution:

- **Tokenisation: Initial Tokens (including punctuation, capitalization, and numbers):** ['hi', 'there', '!', 'did', 'you', 'hear', 'about', 'the', 'new', 'café', 'that', 'just', 'opened', 'downtown', '?', "''", 'asked', 'sarah', '.', '``', 'no', ',', 'i', 'have', "n't", '.', 'what', "'s", 'so', 'special', 'about', 'it', '?', "''", 'replied', 'john', ',', 'looking', 'curious', '.', '``', 'it', "'s", 'supposed', 'to', 'have', 'the', 'best', 'espresso', 'in', 'town—people', 'are', 'raving', 'about', 'it', '!', "''", 'sarah', 'responded', 'enthusiastically', '.', '``', 'really', '?', 'i', '’', 'm', 'always', 'on', 'the', 'lookout', 'for', 'good', 'coffee', '.', 'when', 'should', 'we', 'check', 'it', 'out', '?', "''", 'john', 'asked', '.', '``', 'how', 'about', 'this', 'saturday', '?', 'i', 'heard', 'they', 'have', 'a', 'live', 'jazz', 'band', 'in', 'the', 'evening', ',', "''", 'sarah', 'suggested', '.', '``', 'that', 'sounds', 'fantastic', '!', 'i', '’', 'm', 'in', '.', 'let', '’', 's', 'meet', 'there', 'at', '7', 'pm', ',', 'then', ',', "''", 'john', 'agreed', '.', '``', 'perfect', '!', 'i', "'ll", 'text', 'you', 'the', 'address', 'later', '.', 'by', 'the', 'way', ',', 'did', 'you', 'see', 'the', 'new', 'movie', 'trailer', 'for', "'the", 'last', 'horizon', "'", '?', 'it', 'looks', 'amazing', ',', 'do', "n't", 'you', 'think', '?', "''", 'sarah', 'inquired', '.', '``', 'oh', ',', 'i', 'did', '!', 'it', 'looks', 'intense', '.', 'i', 'can', '’', 't', 'wait', 'to', 'see', 'it', ',', "''", 'john', 'said', 'with', 'excitement', '.']

Filtered Tokens (stopwords and punctuation removed, numbers retained): ['hi', 'hear', 'new', 'café', 'opened', 'downtown', 'asked', 'sarah', 'special', 'replied', 'john', 'looking', 'curious', 'supposed', 'best', 'espresso', 'raving', 'sarah', 'responded', 'enthusiastically', 'really', 'always', 'lookout', 'good', 'coffee', 'check', 'john', 'asked', 'saturday', 'heard', 'live', 'jazz', 'band', 'evening', 'sarah', 'suggested', 'sounds', 'fantastic', 'let', 'meet', '7', 'pm', 'john', 'agreed', 'perfect', 'text', 'address', 'later', 'way', 'see', 'new', 'movie', 'trailer', 'last', 'horizon', 'looks', 'amazing', 'think', 'sarah', 'inquired', 'oh', 'looks', 'intense', 'wait', 'see', 'john', 'said', 'excitement']

- Total tokens after stopword removal = 68
- Unique types (distinct words) = 58
- Type–Token Ratio (TTR) = 58/68 = 0.85

Words ranked by frequency:
sarah: 4

john: 4
new: 2
asked: 2
see: 2
looks: 2
hi: 1
hear: 1
café: 1
All the rest: 1

2. **Stem the following words using Porter Stemmer**
    a. **Excitement**
    b. **Misunderstanding**
    c. **Transcontinental**
    d. **Substitutional**

Excitement

suffix -ment matches; condition m > 1 holds → remove -ment
→ excite
terminal -e removal rule: if m > 1 remove final e → excit

Misunderstanding

suffix -ing matches and the stem (misunderstand) contains a vowel → remove -ing → misunderstand

Transcontinental
suffix -al is in the suffix list;
condition m > 1 holds → remove -al → transcontinent

Substitutional
suffix -tional matches the mapping -tional → -tion → replace -tional by -tion → substitution
suffix -ion is in the list but only removed if preceded by s or t and m > 1.
Here substitution ends …t ion (preceded by t) and m > 1 → remove -ion → substitut

3. **Determine the Levenshtein distance for the following:**
    a. **“Anthropomorphization” and “Anthropomorphism"**
    b. **“Overcompensation" and "Overcompensate”**
    c. **“Counterproductive" and "Counterproductivity”**

"Anthropomorphization" and "Anthropomorphism"

- Levenshtein Distance: 6

The minimum edits required to transform "Anthropomorphization" (20 characters) into "Anthropomorphism" (16 characters) are:

1. Substitution: 'z' $\rightarrow$'s' (1 edit)
2. Deletion: 'a' (1 edit)
3. Deletion: 't' (1 edit)
4. Deletion: 'i' (1 edit)
5. Deletion: 'o' (1 edit)
6. Deletion: 'n' (1 edit)

Overcompensation" and "Overcompensate"

- Levenshtein Distance: 3

The minimum edits required to transform "Overcompensation" (16 characters) into "Overcompensate" (14 characters)

1. Substitution: 'i' $\rightarrow$ 'e' (1 edit)
2. Deletion: 'o' (1 edit)
3. Deletion: 'n' (1 edit)

"Counterproductive" and "Counterproductivity"

- Levenshtein Distance: 3

The minimum edits required to transform "Counterproductive" (17 characters) into "Counterproductivity" (19 characters)

1. Substitution: 'e' $\rightarrow$ 'i' (1 edit)
2. Insertion: 't' (1 edit)
3. Insertion: 'y' (1 edit)

4. **Determine the Hamming distances for the following:**
    a. **M4ch1n3L3arn1ng and M4ch1n3L3arN1ng**
    b. **GATTACA and GATCTAC**

M4ch1n3L3arn1ng and M4ch1n3L3arN1ng

The only mismatch occurs at position 12 ('n' vs. 'N').

Hamming Distance = 1

GATTACA and GATCTAC

The mismatches occur at positions 4 ('T' vs. 'C'), 5 ('A' vs. 'T'), and 6 ('C' vs. 'A').

Hamming Distance = 3

Exercise 2

Your task is to build an information retrieval system for a small collection of documents. The system should:

1. **Calculate the TF-IDF for each document in the corpus.**
2. **Use the Vector Space Model (VSM) to rank documents based on their cosine similarity to a given query.**
3. ~~**Incorporate query expansion using relevant terms from feedback.**~~

Corpus of Documents:

1. **D1: "Machine learning models are powerful tools for data-driven decision-making."**
2. **Document 2: "Deep learning, a subset of machine learning, has revolutionized many industries."**
3. **Document 3: "Neural networks are essential for deep learning, which is a branch of artificial intelligence."**
4. **Document 4: "Machine learning can be applied to both supervised and unsupervised tasks."**
5. **Document 5: "Data science involves machine learning, statistics, and big data technologies."**

Preprocessed Documents (Tokens):

1. D1: ['machine', 'learning', 'models', 'powerful', 'tools', 'data', 'driven', 'decision', 'making']
2. D2: ['deep', 'learning', 'subset', 'machine', 'learning', 'revolutionized', 'many', 'industries']
3. D3: ['neural', 'networks', 'essential', 'deep', 'learning', 'branch', 'artificial', 'intelligence']
4. D4: ['machine', 'learning', 'applied', 'supervised', 'unsupervised', 'tasks']
5. D5: ['data', 'science', 'involves', 'machine', 'learning', 'statistics', 'big', 'data', 'technologies']

Vocabulary (V): 30 unique words/tokens.

Term (t)	DF(t)	IDF(t) = log10(5/DF)

learning	5	0.000
machine	4	0.097
deep	2	0.398
data	2	0.398
models	1	0.699
neural	1	0.699
networks	1	0.699
science	1	0.699
(All other 1-DF terms)	1	0.699

TF-IDF of document 1

Term (t)	TF(t, D1)	IDF(t)	TF-IDF(t, D1)
machine	1	0.097	0.097
learning	1	0.000	0.000
models	1	0.699	0.699
data	1	0.398	0.398
powerful	1	0.699	0.699
Rest: driven, decision, making, tools..	1	0.699	0.699

TF-IDF of document 2

Term (t)	TF(t, D2)	IDF(t)	TF-IDF(t, D2)
learning	2	0.000	0.000
deep	1	0.398	0.398
machine	1	0.097	0.097
subset	1	0.699	0.699
(Others: revolutionized, industries, many)	1	0.699	0.699

(Do the same for document 3, and 4)

TF-IDF for all documents (do it for all the terms -- will be a 30 x 5 table)

Term (t)	D1	D2	D3	D4	D5
learning	0.000	0.000	0.000	0.000	0.000
machine	0.097	0.097	0.000	0.097	0.097
deep	0.000	0.398	0.398	0.000	0.000
data	0.398	0.000	0.000	0.000	0.796
models	0.699	0.000	0.000	0.000	0.000
neural	0.000	0.000	0.699	0.000	0.000
networks	0.000	0.000	0.699	0.000	0.000
science	0.000	0.000	0.000	0.000	0.699
supervised	0.000	0.000	0.000	0.699	0.000
unsupervised	0.000	0.000	0.000	0.699	0.000

2. Use the Vector Space Model (VSM) to rank documents based on their cosine similarity to a given query.

Vector	Length
Query	$(\|\vec{Q}\| = \sqrt{5} \approx 2.236)$
Doc1	$(\|\vec{D}_1\| = \sqrt{10} \approx 3.162)$
Doc2	$(\|\vec{D}_2\| = \sqrt{12} \approx 3.464)$
Doc3	$(\|\vec{D}_3\| = \sqrt{14} \approx 3.742)$
Doc4	$(\|\vec{D}_4\| = \sqrt{11} \approx 3.317)$
Doc5	$(\|\vec{D}_5\| = \sqrt{11} \approx 3.317)$

Compute Dot Products

- Doc1: 1∗1+1∗1+1∗1+1∗1+1∗1=5
- Doc2: 1∗1+1∗2+0+0+0=3
- Doc3: 0+1∗1+1∗1+0+0=2
- Doc4: 1∗1+1∗1+0+0+0=2
- Doc5: 1∗1+1∗1+0+0+0=2

Document	Cosine Similarity
Doc1	0.707
Doc2	0.387
Doc3	0.239
Doc4	0.270
Doc5	0.270

5. Corpus:
○ I saw the boy
○ the man is working
○ I walked in the street
Compute the following:
● Unigram Counts

- **Maximum Likelihood for Bigrams**
- **Compute the probability of the sentence: I saw the man**
- **Compute the probability of the sentence: I saw the man in the street**

- **Another corpus:**
  - **I love machine learning**
  - **I love deep learning**
  - **Deep learning is great**
  - **Machine learning is fun**

Compute MLE for Trigrams for this corpus

Word	Count
<s>	3
I	2
saw	1
the	3
boy	1
man	1
is	1
working	1
walked	1
in	1
street	1
</s>	3

Bigram	Count

<s> I	2
<s> the	1
I saw	1
I walked	1
saw the	1
the boy	1
the man	1
man is	1
is working	1
walked in	1
in the	1
the street	1
boy </s>	1
working </s>	1
street </s>	1

P(I | <s>) = 2/3 ≈ 0.667

P(the | <s>) = 1/3 ≈ 0.333

P(saw | I) = 1/2 = 0.5

P(walked | I) = 1/2 = 0.5

P(the | saw) = 1/1 = 1

P(man | the) = 1/3 ≈ 0.333

P(boy | the) = 1/3 ≈ 0.333

P(street | the) = 1/3 ≈ 0.333

P(</s> | boy) = 1/1 = 1

P(</s> | working) = 1/1 = 1

P(</s> | street) = 1/1 = 1

Trigram	Count	Prefix Bigram	Count
<s> <s> I	2	<s> <s>	4
<s> <s> Deep	1	<s> <s>	4
<s> <s> Machine	1	<s> <s>	4
<s> I love	2	<s> I	2
<s> Deep learning	1	<s> Deep	1
<s> Machine learning	1	<s> Machine	1
I love machine	1	I love	2
I love deep	1	I love	2
love machine learning	1	love machine	1
love deep learning	1	love deep	1
machine learning </s>	1	machine learning	1
deep learning </s>	1	deep learning	1
Deep learning is	1	Deep learning	1
Machine learning is	1	Machine learning	1
learning is great	1	learning is	2
learning is fun	1	learning is	2
is great </s>	1	is great	1
is fun </s>	1	is fun	1

Now calculate the probabilities based on these values.