### **Exercises**

Provide hands-on experience with Python basics, string manipulation, and data handling using popular libraries, building a foundation for NLP applications.

**1 Practice Python Basics**

These exercises focus on essential Python skills, including lists, strings, and dictionaries. The objective is to reinforce Python syntax and basic operations that are foundational for handling text data in NLP.

**Exercise 1**: Create a list of sentences and count the total number of words across all sentences.

python

sentences = ["Data science is intriguing.", "NLP is a vital part of AI.", "Python is widely used in NLP."]

# Count total words across all sentences

total\_words = sum(len(sentence.split()) for sentence in sentences)

print("Total number of words:", total\_words)

# Expected Output: Total number of words: 12

**Exercise 2**: Using a dictionary, store the names of fruits and their corresponding quantities, then print each fruit's quantity if it exceeds 10.

python

*fruit\_quantities = {"apples": 5, "bananas": 15, "oranges": 12, "pears": 8}*

# Print fruits with quantities greater than 10

*for fruit, quantity in fruit\_quantities.items():*

*if quantity > 10:*

*print(f"{fruit}: {quantity}")*

# Expected Output:

# bananas: 15

# oranges: 12

**Exercise 3**: Convert a string into a list of characters, reverse the list, and then join it back into a reversed string.

python

*text = "NLP is powerful"*

# Reverse the string

*reversed\_text = "".join(list(text)[::-1])*

*print("Reversed Text:", reversed\_text)*

# Expected Output: Reversed Text: lufrewop si PLN

**2 String Manipulation and Cleaning with Regex**

This activity introduces cleaning raw text data by using string functions and regular expressions, focusing on removing unwanted patterns like punctuation, numbers, and special characters.

**Exercise 1**: Remove all numbers and special characters from a sentence.

python

*import re*

*raw\_text = "Data Science 101: NLP & ML! Let's start."*

# Remove numbers and special characters

*cleaned\_text = re.sub(r'[^A-Za-z\s]', '', raw\_text)*

*print("Cleaned Text:", cleaned\_text)*

# Expected Output: Cleaned Text: Data Science NLP ML Lets start

**Exercise 2**: Find and extract all words that start with a capital letter.

python

text = "Machine Learning and Natural Language Processing are core in AI."

# Find all capitalized words

capitalized\_words = re.findall(r'\b[A-Z][a-z]\*\b', text)

print("Capitalized Words:", capitalized\_words)

# Expected Output: Capitalized Words: ['Machine', 'Learning', 'Natural', 'Language', 'Processing', 'AI']

**Exercise 3**: Remove extra whitespace (more than one space) from a text.

python

text = "Data science and NLP are closely related."

# Remove extra whitespace

cleaned\_text = re.sub(r'\s+', ' ', text).strip()

print("Text with Single Spacing:", cleaned\_text)

# Expected Output: Text with Single Spacing: Data science and NLP are closely related.

**3 DataFrame Creation with pandas**

This activity focuses on creating and manipulating a pandas DataFrame, a useful data structure for handling tabular data, which often comes in handy in NLP when organizing text and its associated metadata.

**Exercise 1**: Create a DataFrame with sample text and its length.

python

import pandas as pd

# Create a sample list of text data

data = {

"Text": ["Data science is important.", "Machine learning enables predictive analysis.",

"NLP handles human language."],

"Length": [len(text) for text in ["Data science is important.",

"Machine learning enables predictive analysis.",

"NLP handles human language."]]

}

# Create DataFrame

df = pd.DataFrame(data)

print(df)

# Expected Output:

# Text Length

# 0 Data science is important. 25

# 1 Machine learning enables predictive analysis. 40

# 2 NLP handles human language. 26

**Exercise 2**: Add a column that indicates whether each text contains the word "NLP".

python

df["Contains\_NLP"] = df["Text"].apply(lambda x: "NLP" in x)

print(df)

# Expected Output:

# Text Length Contains\_NLP

# 0 Data science is important. 25 False

# 1 Machine learning enables predictive analysis. 40 False

# 2 NLP handles human language. 26 True

**4 Word Frequency Counting with Dictionaries**

This exercise demonstrates how to count word frequencies in a sample text and store the results in a dictionary, a core NLP technique for understanding text content.

**Exercise 1**: Count the frequency of each word in a short paragraph.

python

text = "Deep learning and machine learning are integral to artificial intelligence. Machine learning is a subset of AI."

word\_counts = {}

# Tokenize and count word frequency

for word in text.lower().split():

word = re.sub(r'[^a-z]', '', word) # Remove punctuation

word\_counts[word] = word\_counts.get(word, 0) + 1

print("Word Frequencies:", word\_counts)

# Expected Output: Word Frequencies: {'deep': 1, 'learning': 3, 'and': 1, 'machine': 2, 'are': 1, 'integral': 1, 'to': 1, 'artificial': 1, 'intelligence': 1, 'is': 1, 'a': 1, 'subset': 1, 'of': 1, 'ai': 1}

**Exercise 2**: Filter out words that appear only once from the frequency dictionary.

python

# Filter out words with frequency of 1

frequent\_words = {word: count for word, count in word\_counts.items() if count > 1}

print("Frequent Words:", frequent\_words)

# Expected Output: Frequent Words: {'learning': 3, 'machine': 2}

**5 Summarize and Output Cleaned Text**

This exercise demonstrates how to present cleaned, preprocessed text in a formatted summary, preparing it for further NLP tasks or reporting.

**Exercise 1**: Create a summary of cleaned text by outputting unique words and their frequencies.

python

# Using the word\_counts dictionary from previous exercise

unique\_words = sorted(word\_counts.keys())

print("Summary of Cleaned Text:")

print("Unique Words:", unique\_words)

print("Word Frequencies:", word\_counts)

# Expected Output:

# Summary of Cleaned Text:

# Unique Words: ['a', 'ai', 'and', 'are', 'artificial', 'deep', 'integral', 'intelligence', 'is', 'learning', 'machine', 'of', 'subset', 'to']

# Word Frequencies: {'deep': 1, 'learning': 3, 'and': 1, 'machine': 2, 'are': 1, 'integral': 1, 'to': 1, 'artificial': 1, 'intelligence': 1, 'is': 1, 'a': 1, 'subset': 1, 'of': 1, 'ai': 1}

**Exercise 2**: Format and output text with stopwords removed.

python

stopwords = {"and", "are", "to", "is", "a", "of"}

cleaned\_text = " ".join([word for word in text.lower().split() if word not in stopwords])

print("Text without Stopwords:", cleaned\_text)

# Expected Output: Text without Stopwords: deep learning machine learning integral artificial intelligence machine learning subset ai