

# Sentiment analysis

NATURAL LANGUAGE PROCESSING

Dr Esraa Afify  
Eng Mostafa Gamal

**Amr Tamer 205046**  
**Nabil Tamer 205079**  
**Moaz Gaafar 205087**

# Agenda



Introduction



Abstract



Methodology

# Abstract

- ▶ The abstract summarizes the sentiment analysis NLP project using Python. It provides a concise overview of the project, including its purpose, methodology, and key findings. The abstract briefly explains the significance of sentiment analysis and NLP in various domains, such as social media monitoring, customer feedback analysis, and market research. It also highlights the specific approach taken in the project and the results obtained. The abstract aims to provide a high-level understanding of the project to the readers.

# Introduction

- ▶ The introduction section provides an overview of the sentiment analysis NLP project using Python. It introduces the concept of sentiment analysis and its importance in understanding people's opinions and emotions from textual data. The section also highlights the relevance of natural language processing (NLP) techniques in sentiment analysis and discusses the goals and objectives of the project.

# Methodolgy

5

**Importing Libraries:** The code begins by importing necessary libraries such as `string`, `Counter` from `collections`, `matplotlib.pyplot`, and various modules from the Natural Language Toolkit (`nltk`).

**Downloading NLTK Resources:** The code uses `nltk.download()` to download required resources such as tokenizers, stopwords, WordNet, and the VADER sentiment lexicon.

**Reading and Cleaning the Text:** The code reads the contents of a file named 'read2.txt' and stores it in the `text` variable. It then converts the text to lowercase and removes punctuation using the `translate()` method.

**Tokenization and Stopword Removal:** The code tokenizes the cleaned text into individual words using the `word_tokenize()` function. It removes stopwords (commonly occurring words with little semantic meaning) using the `stopwords.words('english')` list from NLTK.

**Lemmatization:** The code performs lemmatization on the remaining words using the `WordNetLemmatizer` from NLTK, which reduces words to their base form (e.g., plural to singular, different verb tenses to base form).

**Identifying Emotions:** The code reads a file named 'Feelings.txt' which contains a list of words associated with specific emotions. It matches the lemmatized words with the words in the file and appends the corresponding emotion to the `Feels_list`.

# Methodolgy

6

**Sentiment Analysis:** The code defines a function `sentiment_analyse()` that uses the VADER (Valence Aware Dictionary and sentiment Reasoner) sentiment analyzer from NLTK to analyze the sentiment of the cleaned text. It calculates a sentiment score and prints the sentiment as "Negative Sentiment," "Positive Sentiment," or "Neutral Sentiment" based on the score

**Counting Emotions:** The code uses the `Counter()` function to count the occurrences of each emotion in the `Feels_list` and stores the results in the `w` variable.

**Visualization:** The code plots a bar graph using `matplotlib.pyplot` to visualize the frequency of different emotions. The graph is saved as 'graph.png' and displayed using `plt.show()`.



THANK  
YOU