

# Micro IT Project 1: Sentiment Analysis

Kalpana Singh

June 13, 2025

- Sentiment analysis is a machine learning technique used to determine the emotional tone behind text data.
- This project focuses on analyzing customer reviews to classify sentiments as positive, negative, or neutral.
- The goal is to understand user feedback for products or services using natural language processing (NLP).

# Objectives of the Project

- Develop a system to automatically classify text sentiments using NLP techniques.
- Analyze a large dataset of customer reviews (1200 entries) to identify emotional trends.
- Visualize sentiment distribution to provide actionable insights for businesses.
- Gain hands-on experience with machine learning and data processing tools.

- **Python:** Core programming language for implementation.
- **NLTK (Natural Language Toolkit):** For sentiment analysis using the VADER model.
- **Pandas:** For handling and processing the CSV dataset.
- **Matplotlib & Seaborn:** For visualizing sentiment distribution.
- **VS Code:** As the development environment.

# System Design and Architecture

- **Input:** A CSV file with 1200 customer reviews in a text column.
- **Preprocessing:** Convert text to lowercase, remove punctuation, tokenize, and filter stopwords.
- **Sentiment Analysis:** Use VADER to compute sentiment scores and classify as Positive, Negative, or Neutral.
- **Output:** Display results in a table, visualize with a bar chart, and save to a new CSV.

Input → Preprocessing → Sentiment Analysis → Visualization  
→ Output

# Implementation Details

- **Data Generation:** Created a script (`generate_reviews_csv.py`) to produce 1200 synthetic reviews.
- **Script:** `sentiment_analysis.py` loads the CSV, preprocesses text, applies VADER, and visualizes results.
- **Preprocessing Steps:** Lowercase conversion, punctuation removal, tokenization, stopword removal.
- **Output:** Table of sentiments, bar chart of distribution, and `sentiment_results.csv`.

# This Internship Project Taught Me

- Practical application of NLP and machine learning in real-world scenarios.
- Handling large datasets and preprocessing text data effectively.
- Using visualization tools to interpret and present data insights.
- Debugging and resolving issues in Python, such as NLTK resource errors and file handling.
- Importance of clear documentation and structured project workflow.

- Integrate more advanced models like BERT for improved sentiment accuracy.
- Expand the dataset to include multilingual reviews for broader applicability.
- Develop a web interface for real-time sentiment analysis of user inputs.
- Incorporate sentiment trends over time to track changes in customer feedback.
- Deploy the system on a cloud platform for scalability and accessibility.