Social Media Sentiment Analysis using NLP

Project 6 – Coding Samurai Data Science Internship

Presented by: **Ridhwan S**

Project Overview

- Analyzed ~1.6M tweets using Natural Language Processing (NLP)
- Goal: Classify tweets as Positive or Negative
- Dataset: Sentiment140 dataset (tweets.csv)
- Tools: Python, Pandas, NLTK, Scikit-learn, Matplotlib, WordCloud

Workflow

1. Data Loading & Cleaning
2. Text Preprocessing (Tokenization, Lemmatization)
3. Feature Extraction (TF-IDF)
4. Sentiment Classification (Logistic Regression)
5. Evaluation & Visualization

Dataset Info

- Size: 1.6 million tweets
- Columns: target, id, date, query, user, tweet
- Target Classes:
  - 0 = Negative
  - 4 = Positive
- Balanced classes: 800K each

Data Cleaning

- Removed URLs, mentions, special characters
- Converted to lowercase
- Removed stopwords
- Tokenized and lemmatized tweets
- No missing or duplicate rows

Feature Engineering

- Applied **TF-IDF Vectorization**
- Max Features: 5000
- Transformed tweets into sparse matrix for model input

Model Building

- Classifier: Logistic Regression
- Train-Test Split: 80/20
- Used TF-IDF matrix as feature input
- Target: Sentiment labels (0 and 4)

Model Evaluation

- **Accuracy**: 77.38%
- **Confusion Matrix**:

  [[120000 39494]

  [32888 127618]]

- **Precision, Recall, F1-Score**: Balanced across classes

Visualizations

- Countplot for sentiment distribution
- Word Clouds for Positive vs Negative tweets
- Confusion Matrix Heatmap

Key Learnings

- NLP requires heavy preprocessing for accuracy
- Logistic Regression performed well on text data
- TF-IDF helps reduce noise and focus on relevant words

Future Scope

- Use advanced models like BERT or LSTM
- Perform multi-class sentiment analysis (add Neutral)
- Real-time tweet scraping & analysis (Twitter API or Streamlit)

Thank You!

Ridhwan S

Data Analyst Intern – Coding Samurai

LinkedIn | GitHub