ANALYZING TWITTER SENTIMENT AROUND A PRODUCT LAUNCH USING NLP TECHNIQUES

Executive Summary

This case study demonstrates how Natural Language Processing (NLP) can be used to analyze public sentiment on Twitter regarding a recent product launch. It walks students through tweet extraction, text preprocessing, sentiment classification using Vader/TextBlob, visualization of results, and interpretation of public perception trends over time. The project helps learners apply real-world text data to business analysis scenarios.

1. Introduction

Understanding how people feel about a product launch is critical for marketing and reputation management. With platforms like Twitter offering real-time public opinion, NLP tools allow companies to monitor sentiment and adapt strategies accordingly. This project helps students gain hands-on experience with text mining, sentiment analysis, and exploratory data storytelling.

2. Problem Statement

Analyze the sentiment of tweets containing a given product name or hashtag within a 7-day window before and after its launch. Determine how public sentiment trends shifted and identify key words driving negative or positive responses.

3. Dataset Overview

- **Source**: Live tweets using Tweepy API or sample from Kaggle
- **Size**: ~10,000 tweets
- **Fields**:
  - tweet_text
  - created_at
  - username
  - retweets
  - likes

4. Methodology

Step 1: Data Collection (optional live version)

```
import tweepy

api.search_tweets(q="#ProductX", lang="en", count=100)
```

Or load from sample CSV.

Step 2: Preprocessing

- Lowercase text
- Remove URLs, mentions, hashtags, punctuation
- Tokenize and remove stopwords
- Apply lemmatization

Step 3: Sentiment Scoring

Option 1: Using TextBlob

```
from textblob import TextBlob


def get_sentiment(text):
    score = TextBlob(text).sentiment.polarity
    if score > 0: return 'Positive'
    elif score < 0: return 'Negative'
    else: return 'Neutral'
```

Option 2: Using Vader

```
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer

analyzer = SentimentIntensityAnalyzer()
```

5. Results and Interpretation

Sentiment Distribution

Sentiment	Count	Percentage

Positive	4,170	41.7%
Neutral	3,150	31.5%
Negative	2,680	26.8%

Launch Impact

- **Before Launch**: Mostly curiosity, neutral or mildly positive
- **After Launch**: Spike in negative sentiment tied to delivery delays and pricing
- **Top Positive Words**: "smooth", "love", "fast", "innovation"
- **Top Negative Words**: "late", "overpriced", "buggy", "disappointed"

6. Visualizations

- **Pie chart** of sentiment categories
- **Line plot** of daily sentiment changes
- **Word cloud** for positive and negative tweets
- **Bar chart** for most frequent complaint terms

7. Conclusion

NLP sentiment analysis effectively captured the shift in public mood around the product launch. This kind of analysis supports PR, sales, and product teams with data-driven insights. The case study demonstrates how students can turn unstructured data into actionable business intelligence.

8. Learning Outcomes for Students

- Perform text preprocessing for NLP tasks
- Apply lexicon-based sentiment classifiers
- Create time-based sentiment trend plots
- Translate NLP findings into business narratives

9. Suggested Enhancements

- Build sentiment timeline dashboards in Tableau or Power BI
- Apply topic modeling (LDA) to identify underlying themes

- Train a custom classifier (Naive Bayes, SVM) using labeled tweets
- Integrate with Slack to send alerts on sentiment spikes

10. References

- Bird, Klein, & Loper. *Natural Language Processing with Python*
- Vader Sentiment Docs
- TextBlob Documentation
- Twitter API v2 Docs
- Kaggle: Twitter Sentiment Analysis Datasets