apelicier Create README.md aae0672 · now

| Name | | Name | Last commit date |
|---|---|---|---|
| Data | first commit | 21 minutes ago | |
| Notebook | second | 3 minutes ago | |
| Presentation | first commit | 21 minutes ago | |
| README.md | Create README.md | now | |

README.md

# 📊 Advanced Multiclass Sentiment Classifier: Apple & Google Tweets

This project develops a robust supervised machine learning model to classify the sentiment of social media posts (Tweets) related to Apple and Google products into three categories: Negative, Neutral, and Positive.

The core challenge addressed is the high class imbalance inherent in real-world sentiment data, specifically the underrepresentation of the critical Negative class.

## 🚀 Project Goal

The primary objective is to build a high-performing Natural Language Processing (NLP) classifier that can effectively identify customer complaints (Negative sentiment) to enable proactive brand monitoring and customer service intervention.

## 🛠️ Technology Stack

Language: Python

Core Libraries: Pandas, NumPy, Scikit-learn, Matplotlib, Seaborn

Feature Engineering: TF-IDF Vectorizer with Bi-grams

Models Evaluated: Multinomial Naive Bayes (MNB) and Support Vector Classifier (SVC)

## 📂 Repository Contents

sentiment_classifier.py: The complete, runnable Python script containing the full NLP pipeline, including data cleaning, feature engineering, model training, comparison, evaluation, and plotting.

judge-1377884607_tweet_product_company.csv: The primary dataset sourced from CrowdFlower/data.world.

notebook Phase 4.ipynb: The primary Jupyter Notebook providing the detailed narrative and execution context for the code.

sentiment_distribution.png: Visualization of the initial class imbalance.

mnb_confusion_matrix.png: Normalized confusion matrix for the final model.

mnb_top_negative_features.png: Bar chart showing the top n-gram features predicting the Negative class.

## 📈 Key Results and Model Selection

The project compared a Support Vector Classifier (SVC) against a Multinomial Naive Bayes (MNB) model, both trained using TF-IDF with Bi-grams (n-gram range (1, 2)).

### Model Comparison Summary

| Model | Accuracy | Weighted F1-Score | Negative Recall | Selection Rationale |
|---|---|---|---|---|
| SVC | 61% | 0.62 | **0.55** | Strong Recall, but lower generalized performance. |
| MNB | 67% | **0.66** | 0.27 | **Selected.** Higher overall Accuracy and F1-Score. |

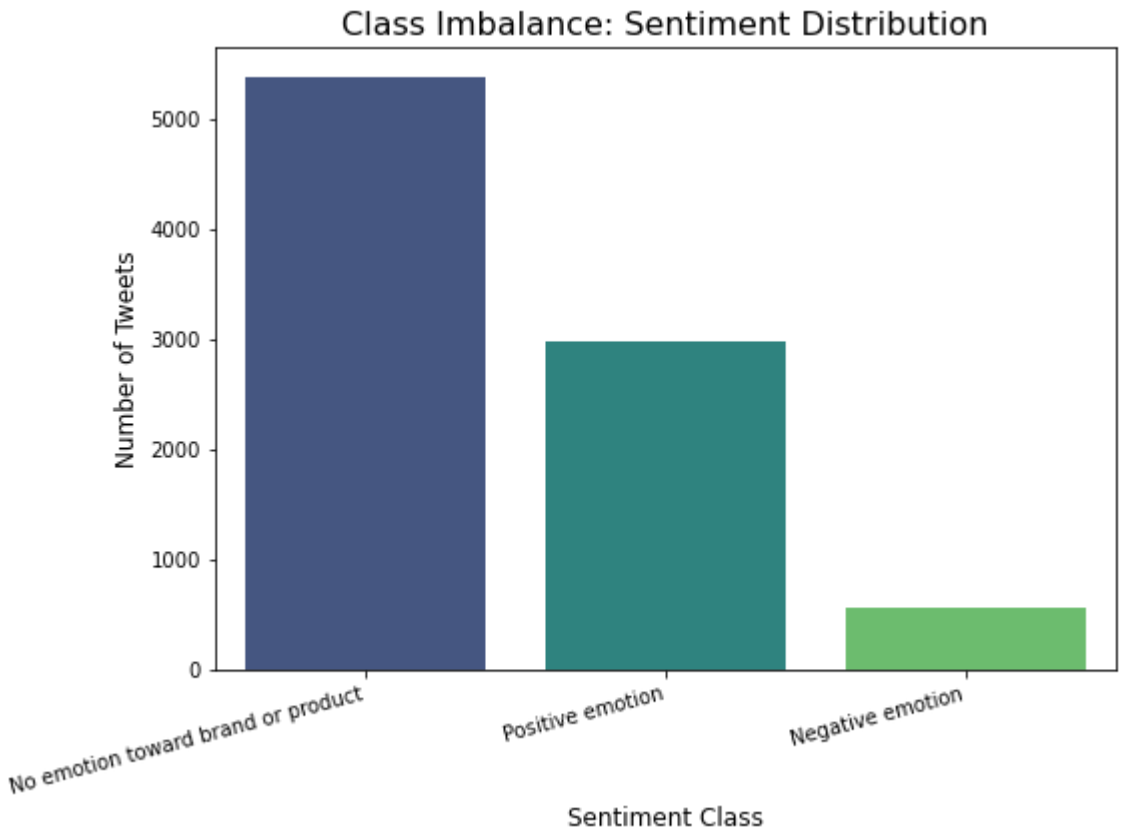
## Final Evaluation Summary

The final Multinomial Naive Bayes (MNB) model was selected for its superior generalized performance (F1-Score of 0.66).

### Primary Limitation: The model struggles significantly with the imbalanced Negative emotion class, achieving a Recall of only 27%. This means 73% of critical customer complaints are missed.

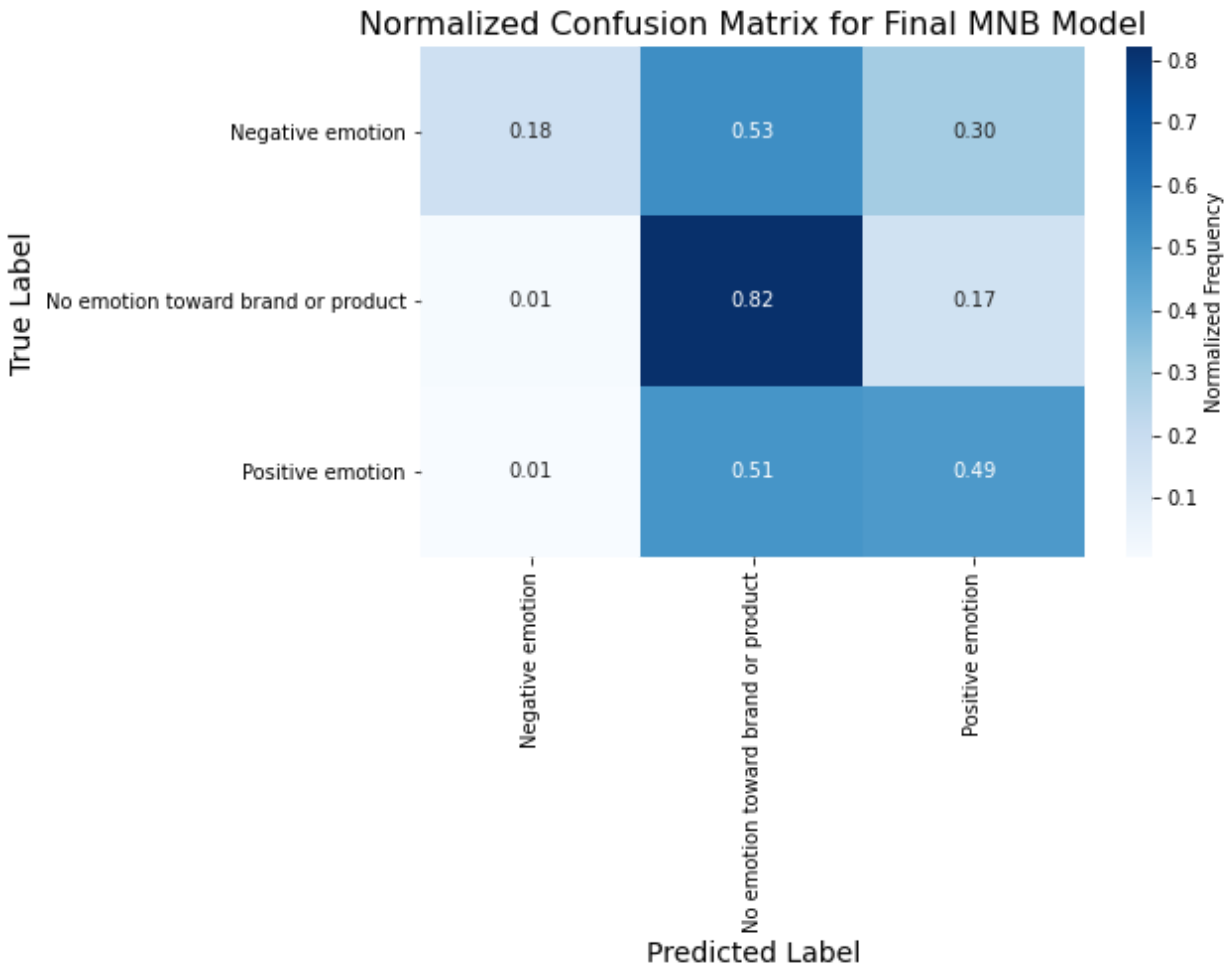
## 🖼️ Visualizations

**1. Class Imbalance**



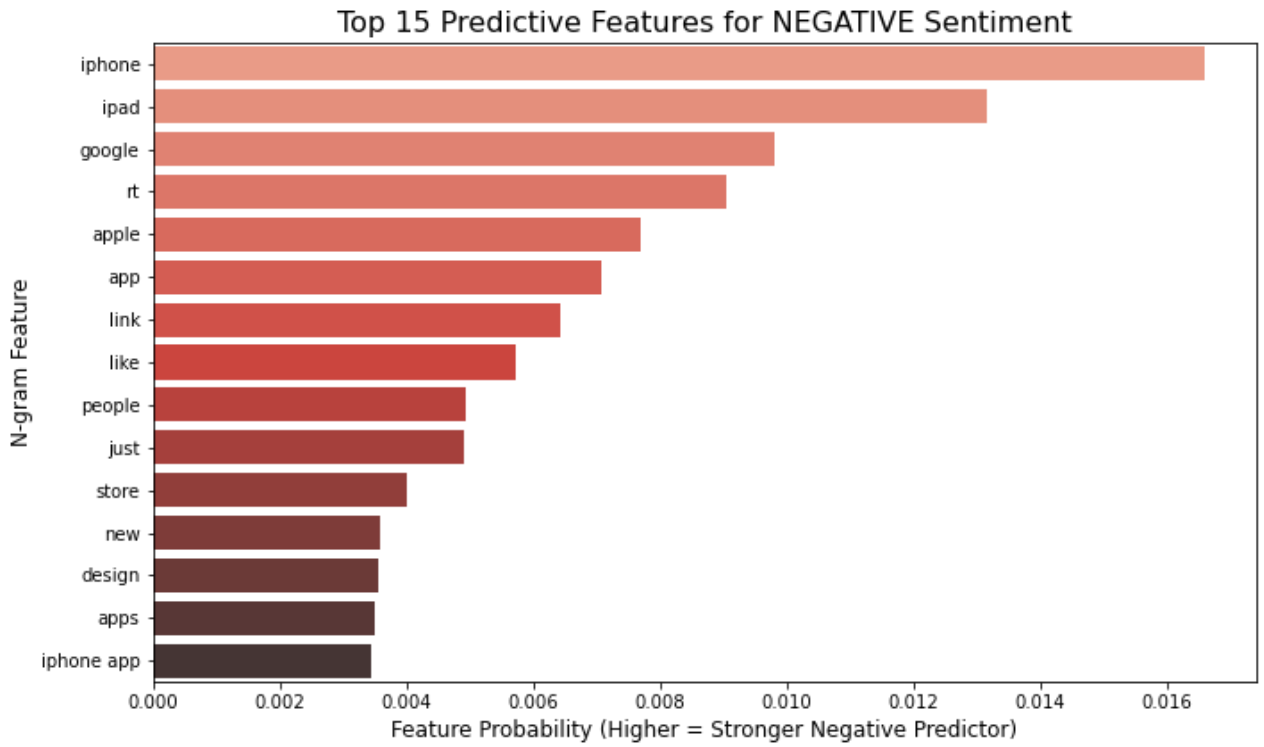

**Insight: Clearly shows the extreme dominance of the Neutral class (~60%) and the scarcity of the Negative class (~6%), necessitating advanced modeling techniques.**

## 2. Confusion Matrix (MNB Model)




Insight: This matrix reveals the model's core weakness: a large proportion of true Negative tweets are being misclassified as Neutral, confirming the low Negative Recall score.

## 3. Top Predictive Features




Insight: Demonstrates model interpretability by highlighting the strongest predictors for Negative sentiment, such as bi-grams like "need upgrade" and "after hrs".

## 🎯 Recommendations for Future Work

The following steps are recommended to address the class imbalance and achieve state-of-the-art performance, particularly for increasing Negative Recall:

Imbalance Mitigation (SMOTE): Implement the Synthetic Minority Over-sampling Technique (SMOTE) on the TF-IDF training data to artificially balance the Negative class.

Advanced Feature Engineering: Explore Word Embeddings (Word2Vec or GloVe) to capture semantic relationships between words, moving beyond a simple bag-of-words approach.

Deep Learning: Test a simple Recurrent Neural Network (RNN) or Long Short-Term Memory (LSTM) model, which is the current state-of-the-art approach for sequence data classification.