### PROJECT TITLE: UNDERSTANDING SENTIMENT THROUGH TWITTER DATA

**Overview**

* This project developed an automated sentiment classification system to help Apple's product management team monitor customer feedback on Twitter. Using a dataset of 9,093 human-labeled tweets from CrowdFlower, we built a multi-class classifier to categorize sentiment as Positive, Negative, or Neutral, with particular focus on detecting customer complaints.
* **Data Preparation**: We implemented comprehensive text preprocessing using NLTK and scikit-learn, including lowercasing, URL removal, tokenization, lemmatization, and stopword removal. The SMOTE (Synthetic Minority Over-sampling Technique) from imbalanced-learn was applied to address severe class imbalance (only 7% negative examples), creating synthetic negative examples for robust model training. Text was vectorized using TF-IDF with 5,000 features and n-grams (1-2).
* **Modeling:** We compared Logistic Regression, Random Forest, and SMOTE-enhanced Logistic Regression using scikit-learn. After rigorous testing, SMOTE Logistic Regression was selected for its superior balance of recall and precision. Hyperparameter tuning focused on regularization strength (C=1.0) and class weighting strategies.
* **Evaluation: The** model achieved 61.7% accuracy on the test set with 50% negative tweets recall (our primary business metric) demonstrating excellent generalization with performance improvements across all metrics from validation to test. Using stratified train-validation-test splits (70-15-15), we validated that the model catches 5.5x more complaints than the original approach while maintaining balanced performance across sentiment classes.

This solution provides immediate business value by automating complaint detection, enabling faster response to customer issues, and delivering actionable insights for product strategy decisions.

**Business Understanding**

**Real-World Problem**

* Stakeholder: Sarah Chen, Senior Product Manager for iOS at Apple Inc.
* Business Challenge: Sarah's team faces significant difficulties in efficiently monitoring and responding to customer feedback about Apple products on Twitter. With thousands of tweets mentioning Apple daily, manual sentiment analysis is:
* Time-consuming: Requires 15-20 hours weekly of manual reading and categorization
* Inconsistent: Different team members apply sentiment labels inconsistently
* Slow: Delays of 2-3 days in identifying emerging product issues
* Not scalable: Impossible to process the full volume of tweets in real-time
* Reactive: Misses opportunities for proactive customer issue resolution

This problem directly impacts Apple's ability to maintain customer satisfaction and quickly address product quality issues.

**Objectives**

* Main Objective Build a machine learning system that automatically classifies Twitter sentiment about Apple products to enable real-time customer feedback monitoring and faster issue resolution**.**

**Specific objectives**

1. Achieve >45% negative recall to catch nearly half of all customer complaints automatically
2. Process tweets in real-time to provide immediate insights to product teams\*
3. Handle class imbalance effectively to detect rare but critical negative feedback
4. Deliver interpretable results that product managers can trust and act upon

**Challenges**

Some of the challenges that were encountered in analyzing the Twitter sentiments include:

● Imbalance in the dataset: The dataset was heavily biased towards the neutral tweets, with the negative tweets being majorly underrepresented (about 6.27% of the total dataset)

● Missing values in data: The dataset had a huge number of missing entries in one of the columns.

● Multi-class model bias: The multi-class model was heavily biased towards the neutral class, and this reduced the instances in which positive and negative tweets were predicted accurately.

**Success Metrics**

* Evaluation Metrics: Negative recall (primary), balanced accuracy, balanced F1-score.
* Success Threshold: Improve negative tweet recall to ~50% while maintaining balanced accuracy.
* Stakeholder Utilization: Sarah and her team will use the system to:
* Monitor real-time sentiment on iOS features and updates.
* Receive alerts for negative feedback needing immediate attention.
* Track sentiment trends post-product changes.
* Identify emerging issues early.
* Prioritize product improvements based on data.

The system integrates into existing workflows via dashboards and alerts, requiring no technical expertise**.**

**Data Understanding**

**Data Source & Description**

* Dataset: Twitter Sentiment Analysis (Apple vs. Google) from CrowdFlower via data.world. Collection Method: Human raters manually labeled tweet sentiment. Size: 9,093 tweets. Time Period: Historical tweets about Apple and Google products.
* The dataset contains 9,093 tweets across 3 columns. This provides a substantial amount of data for building a robust sentiment analysis model.

Column Overview:

1. tweet\_text: The actual content of the tweet (our primary feature).
2. emotion\_in\_tweet\_is\_directed\_at: The brand/product being mentioned (Apple, Google, etc.).
3. is\_there\_an\_emotion\_directed\_at\_a\_brand\_or\_product: The sentiment label (Positive/Negative/Neutral emotion).

**Initial Observations:**

* Tweets mention various tech brands (iPhone, iPad, Google).
* Sentiment labels are provided, making this a supervised learning problem.
* Dataset Info:
* Total entries: 9,093.
* Non-null counts: tweet\_text (9,092), emotion\_in\_tweet\_is\_directed\_at (3,291), is\_there\_an\_emotion\_directed\_at\_a\_brand\_or\_product (9,093).
* All columns are of type object.

**Missing Values:**

* tweet\_text: 1 missing.
* emotion\_in\_tweet\_is\_directed\_at: 5,802 missing.
* is\_there\_an\_emotion\_directed\_at\_a\_brand\_or\_product: 0 missing.
* Duplicates: 22 duplicate rows (0.24% of data).

**Conclusion:** Complete sentiment labels (0% missing in target variable). One missing tweet text and significant missing brand data, though this doesn't impact our core sentiment analysis goal. 22 duplicate entries found that should be removed**.**

**Target Variable Initial Analysis**

* **Target Column:** is\_there\_an\_emotion\_directed\_at\_a\_brand\_or\_product**.**

| Sentiment Category | Count | Percentage |
| --- | --- | --- |
| * No emotion toward brand or product | * 5,389 | * 59.3% |
| * Positive emotion | * 2,978 | * 32.8% |
| * Negative emotion | * 570 | * 6.3% |
| * I can't tell | * 156 | * 1.7% |

**Severe Class Imbalance:** Neutral tweets dominate (59%). Negative sentiment is underrepresented (6%). "I can't tell" represents ambiguous labels.

**Insights:**

* Dominance of Neutral Sentiment: No emotion toward brand or product accounts for nearly 60% of all tweets.
* Significant Class Imbalance: Positive (32.8%), Negative (6.3%), "I can't tell" (1.7%).
* Business Implications: Need careful evaluation beyond accuracy; focus on precision, recall, F1 for minority classes. Use class weighting or oversampling.

**Brand Mention Analysis**

| Brand/Product | Count |
| --- | --- |
| * iPad | * 946 |
| * Apple | * 661 |
| * iPad or iPhone App | * 470 |
| * Google | * 430 |
| * iPhone | * 297 |
| * Other Google product or service | * 293 |
| * Android App | * 81 |
| * Android | * 78 |
| * Other Apple product or service | * 35 |

**Key Observations:** Apple products dominate. Specific hardware (iPad, iPhone) discussed more than general brands. 64% of tweets lack brand info.

**Insights:** Apple dominance; iPad most discussed. Strategic for Apple: Focus on iPad and app sentiment.

**Text Data Characteristics**

**Tweet Length Statistics**:

1. Mean: 104.96 characters.
2. Std: 27.19.
3. Min: 11, Max: 178.
4. 25%: 86, 50%: 109, 75%: 126.

**By Sentiment:**

| Sentiment | Mean Length | Std |
| --- | --- | --- |
| * I can't tell | * 103.98 | * 27.91 |
| * Negative emotion | * 109.45 | * 27.39 |
| * No emotion | * 104.19 | * 27.13 |
| * Positive emotion | * 105.56 | * 27.13 |

**Insights:** Negative tweets longer (more detail in complaints). Sufficient length for sentiment signals.

**Sample Tweets by Sentiment**

1. Negative Examples:

* ".@wesley83 I have a 3G iPhone. After 3 hrs tweeting at #RISE\_Austin, it was dead! I need to upgrade. Plugin stations at #SXSW." (Length: 127, Words: 23)
* Patterns: Complaints about functionality (battery, crashes).

1. Positive Examples:

* "@jessedee Know about @fludapp ? Awesome iPad/iPhone app that you'll likely appreciate for its design. Also, they're giving free Ts at #SXSW" (Length: 139, Words: 22)
* Patterns: Praise for design, excitement.

1. Neutral Examples:

* Informational, no emotion. Ambiguous Examples: Sarcasm or indirect mentions.
* Brand and Sentiment Cross-Analysis
* Apple products show positive dominance, but negative feedback on specific issues (e.g., battery, apps).

**Data Preparation**

**Handling Missing Values and Duplicates**

* Created brand\_specified flag for missing brand data.
* Removed 22 duplicates and 1 missing tweet text.
* Removed tweets with ≤2 words (lacking context).
* New size: 9,064 tweets.
* Mapping Sentiment Labels

**Mapped to three classes:** Positive, Negative, Neutral (merged "I can't tell" with Neutral).

| Sentiment | Count |
| --- | --- |
| * Neutral | * 5,527 |
| * Positive | * 2,968 |
| * Negative | * 569 |

Filtering for Apple-Related Tweets

* Used keywords (e.g., "apple", "iphone") and brand column.
* Apple-related: 5,633 tweets (62.15%).
* Distribution: Neutral (3,055), Positive (2,151), Negative (427).

**Text Preprocessing**

* Function: Lowercasing, remove URLs/mentions, tokenization, lemmatization, stopword removal, filter short words.
* Average length reduced: 105.0 → 67.9 characters (35.3% reduction).
* No empty tweets after cleaning.

**Examples:**

* Before: ".@wesley83 I have a 3G iPhone. After 3 hrs tweeting at #RISE\_Austin, it was dead! I need to upgrade. Plugin stations at #SXSW."
* After: "iphone hr tweeting rise\_austin dead need upgrade plugin station sxsw".

**Train-Validation-Test Split**

* Stratified splits (65%-15%-20% for full; adjusted for Apple dataset).
* Apple: Train (3,661), Val (845), Test (1,127).
* Preserves class distribution.

Conclusion: Data ready for modeling with clean features and labels.

**Modeling**

**Feature Extraction**

* Used TF-IDF Vectorizer (max\_features=5,000, ngram\_range=(1,2), min\_df=5, max\_df=0.7).
* Vocabulary: 1,810 features.
* Focused on Apple dataset for stakeholder relevance.
* Handling Class Imbalance
* Computed class weights: Negative (4.41), Neutral (0.61), Positive (0.87).

**Models Compared**

**Baseline Logistic Regression (No Weights):**

* Val Accuracy: 63.4%.
* Negative Recall: 9%.
* Issue: Biased toward majority class.

**Random Forest (No Weights):**

* Val Accuracy: 60.8%.
* Negative Recall: 8%.
* Similar issues; slightly better neutral F1.

**Logistic Regression (Auto-Balanced Weights):**

* Val Accuracy: 58.2%.
* Negative Recall: 50%.
* Trade-off: Lower accuracy, but 5.5x more complaints caught.

**SMOTE + Logistic Regression (Selected Model):**

**Pipeline: SMOTE oversampling + Logistic Regression.**

* Val Accuracy: 59.1%.
* Negative Recall: 48%.
* Best balance (F1: 36% for negative).

**Model Selection Rationale:** SMOTE provides robust handling of imbalance, better precision-recall trade-off.

**5. Evaluation**

**Final Model: SMOTE Logistic Regression**

**Validation Set**:

* Accuracy: 59.1%.
* Negative: Precision 29%, Recall 48%, F1 36%.
* Neutral: F1 65%.
* Positive: F1 58%.

**Test Set:**

* Accuracy: 61.7% (+2.6% from val).
* Negative: Precision 32%, Recall 50%, F1 39%.
* Neutral: F1 67%.
* Positive: F1 60%.

**Generalizes well (no overfitting).**

| Metric | Validation | Test | Difference |
| --- | --- | --- | --- |
| Accuracy | 0.591 | 0.617 | +0.026 |
| Neg Recall | 0.484 | 0.500 | +0.016 |
| Neg Precision | 0.287 | 0.321 | +0.034 |
| Neg F1 | 0.360 | 0.391 | +0.030 |

**Business Impact:**

* Achieves 50% negative recall (>45% goal). Catches 5.5x more complaints than baseline. Projected: 615 additional complaints monthly (assuming 1,500 negatives).

**Strengths:**

* Robust complaint detection, generalization. Limitations: 32% negative precision (needs human verification).

**Deployment**

Current State: Analysis in Jupyter Notebook; model saved as .pkl file.

**Next Steps:**

* Wrap in FastAPI for API endpoints.
* Dockerize with dependencies; use Docker Compose/Kubernetes for orchestration.
* Deploy via Streamlit, Gradio, or Hugging Face for interactive UI.
* Integrate CI/CD pipeline for updates.
* Integration: Real-time Twitter API feed; dashboards/alerts for stakeholders.
* Recommendations & Future Work
* Enhancements: Incorporate deep learning (e.g., RoBERTa) for better multi-class performance.
* Monitoring: Track model drift; retrain with new data.
* Expansion: Include more brands; add sarcasm detection.
* Business Value: Enables proactive issue resolution, boosting customer loyalty.

This report concludes the project, delivering a scalable sentiment analysis solution aligned with Apple's needs.