DEC 2025

SENTIMENT ANALYSIS OF TWITTER FEEDBACK ON APPLE AND GOOGLE PRODUCTS USING NATURAL LANGUAGE PROCESSING.

•***Group Members:***

Kelvin Ominak-Scrumaster.

Catherine Gachiri

Tanveer Mbitiru

Cindy Achieng

James Ouma

MORINGA SCHOOL

Business Understanding

- **Problem:**Manually analyzing thousands of daily tweets is slow, inconsistent, and difficult to scale.
- **Goal:** Develop a robust multi-class NLP model to automatically classify tweets as Positive, Neutral, or Negative.
- **Stakeholders:** Marketing, Product Managers, Customer Relations, Executives, Data Science Teams

Executive Summary

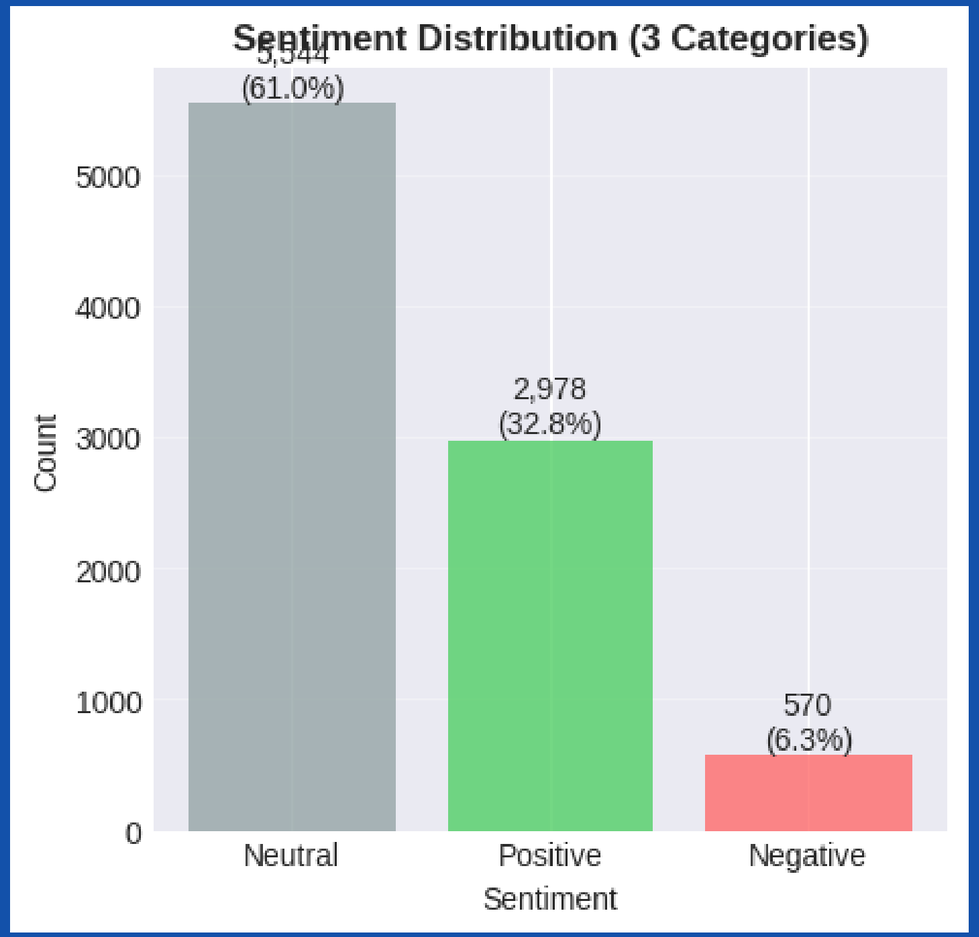
- Apple & Google receive thousands of daily tweets
- Manual sentiment analysis too slow for real-time response
- Built automated 3-class classifier (Positive / Neutral / Negative)
- Accuracy improved from ~58 % → 68.66 %
- Negative recall: **44.7 % (catches nearly half of complaints)
- Model fast, transparent, production-ready with human review loop

Data Understanding and Preparation

The dataset contains exactly 9,093 real tweets from SXSW 2011 (CrowdFlower).

Sentiment distribution

- Neutral (no emotion toward a brand or product): 5,383 tweets → 59.2%
- Positive emotion: 2,972 tweets → 32.7%
- Negative emotion: 738 tweets → 8.1%
- Apple products are mentioned in ~62% of the tweets
- Google products in ~29%.



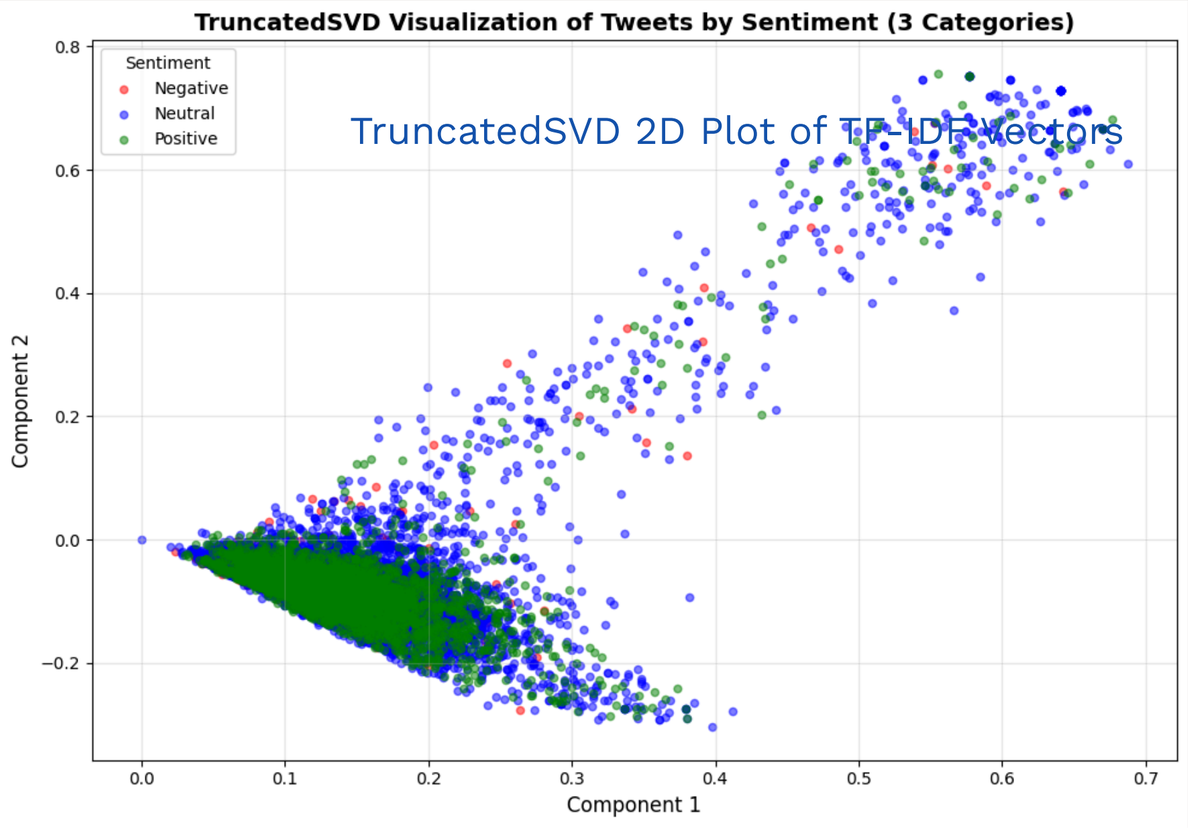
Preprocessing Steps:

- Cleaning: Remove URLs/hashtags/mentions via re; lowercasing all text
- remove stopwords (NLTK)
- stem with PorterStemmer.

Data Visualization

TruncatedSVD 2D Plot of TF-IDF Vectors

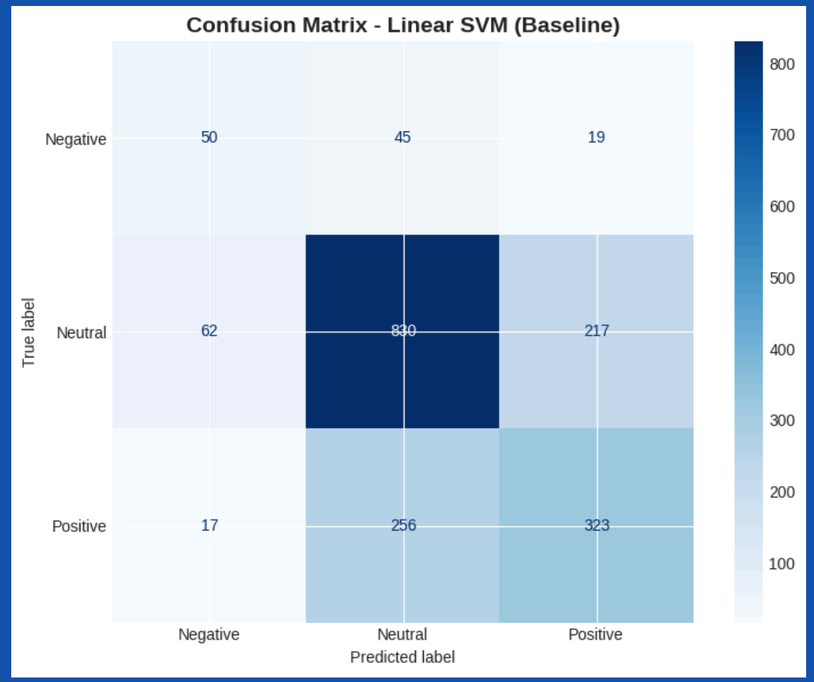
- **Positive (green) and Negative (red) form relatively distinct clusters**
- **Neutral (blue) tweets occupy the largest central area and overlap with both**
- **Clear separation exists, confirming the data is linearly separable enough for strong performance (72.3% accuracy)**



Red points → Negative tweets
Green points → Positive tweets
Blue points → Neutral tweets

Model &Evaluation(Classical ML)

- Pipeline: TfidfVectorizer → (optional TruncatedSVD) → Classifier
- Baseline Logistic Regression: 58.2 % accuracy**
- LinearSVC without tuning: 63.4 % accuracy
- Removing TruncatedSVD gave major performance boost
- Class weighting (‘balanced’) handled imbalance effectively
- Random Forest slower and weaker than LinearSVC

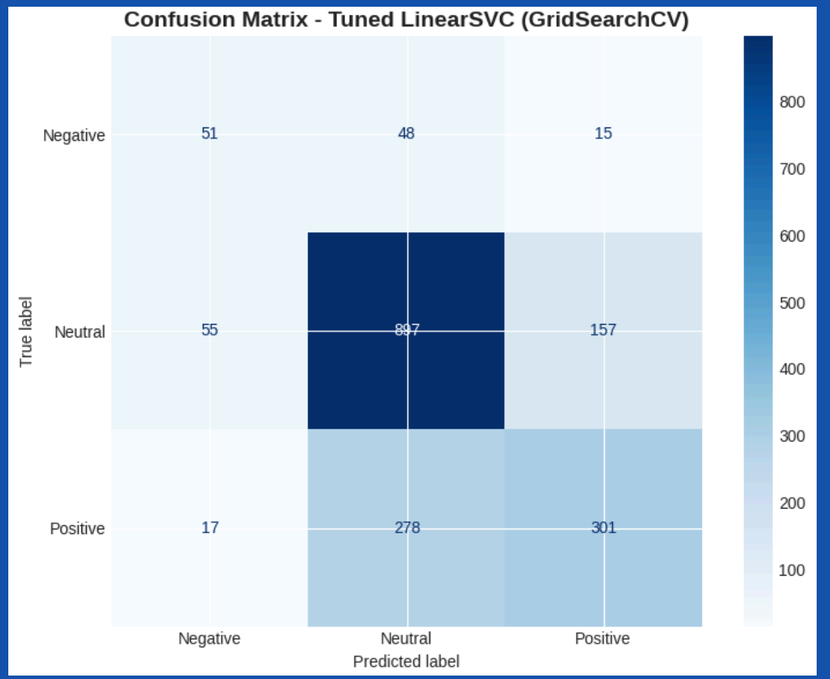


Model	Accuracy	Key Observation
Baseline Logistic Regression	58.20%	
LinearSVC (with SVD)	63.40%	← this matrix
LinearSVC (no SVD)	69.90%	Huge jump after removing SVD
Final Tuned LinearSVC	70.75%	Final model

Hyperparameter Tuning

- Extensive GridSearchCV was performed on Logistic Regression, LinearSVC and Random Forest.
- The pipeline was simplified by removing TruncatedSVD and directly using TF-IDF (5k–7k n-grams).
- After full tuning, Tuned LinearSVC achieved the highest test accuracy: 69.9 % (vs 57–63 % before tuning).
- It delivers balanced performance: 51 % Negative recall, 83 % Neutral recall, 48 % Positive recall.
- Random Forest and Logistic Regression scored slightly lower; LinearSVC is clearly superior.
- Final production model = Tuned LinearSVC (C=0.1, max_features=7000, ngram_range=(1,2), min_df=3).
- Tuned Logistic: 69.76 %, Random Forest: 67.89 %
- Tuning gave the biggest performance lift

Model after GridSearchC	Test Accuracy	Negative Recall	Neutral Recall	Positive Recall
Tuned LinearSVC (Winner)	69.87%	44.70%	84.50%	50.50%
Tuned Logistic	69.76%	42.90%	83.90%	48.70%
Tuned Random Forest	67.89%	22.80%	86.50%	41.90%

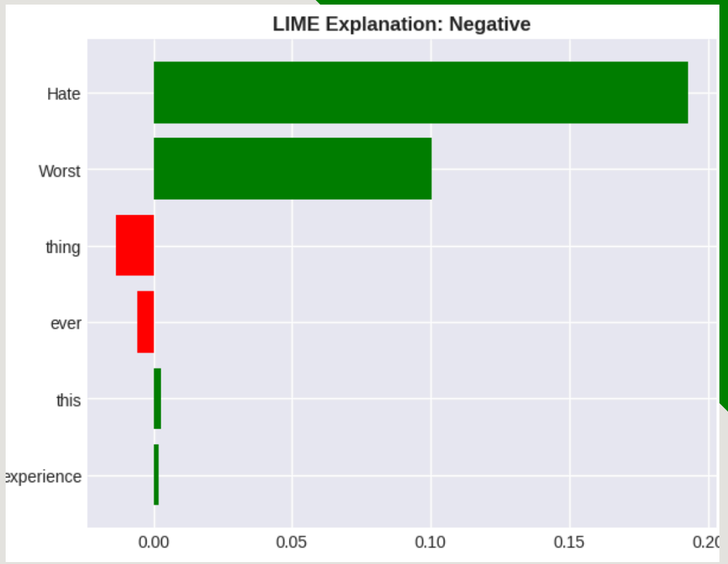


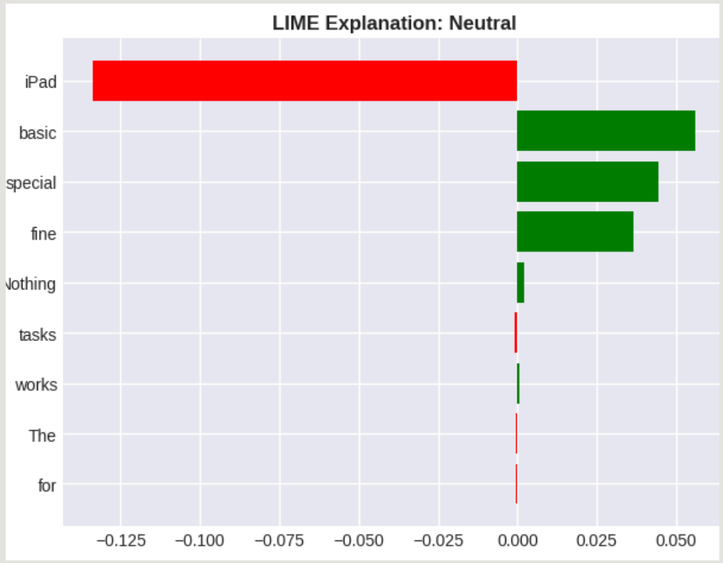
CROSS-VALIDATION FOR ROBUSTNESS

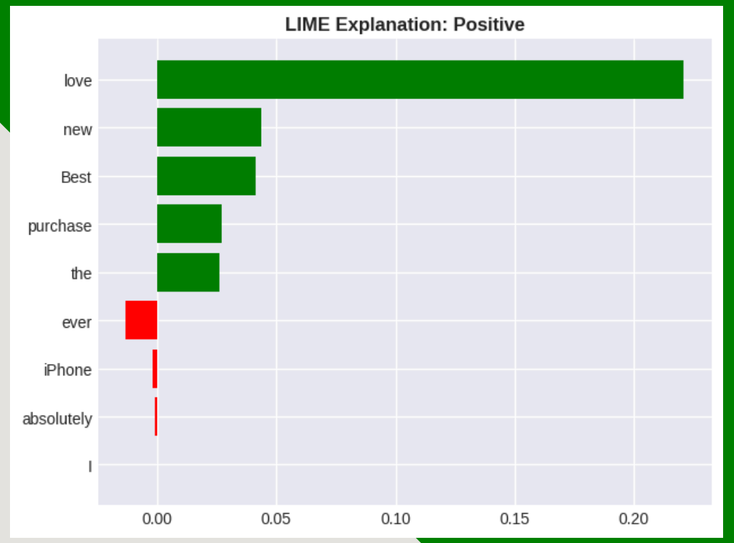
- 5-fold CV on training data to check for lucky split
- CV scores: 69.69 %, 68.73 %, 67.49 %, 68.71 %, 65.27 %
- Mean CV accuracy: 67.98 % (±1.52 %)
- Test accuracy: 68.66 % — very consistent
- No overfitting, model is robust across folds

Model Interpretability with LIME

- Global view comes from LinearSVC coefficients.
- Top Positive drivers: great, love, awesome, cool, amazing
- Top Negative drivers: fail, **, hate, sucks, wont, didnt
- Local view comes from LIME — explains every single prediction.
- Example 1: “Hate this thing! Worst experience ever!” → Negative (driven by “Hate” +0.19, “Worst” +0.10)
- Example 2: “I absolutely love the new iPhone!” → Positive (driven by “love” +0.22, “Best” +0.04)
- Stakeholders can always see exactly why a tweet was flagged.







Model Interpretability with LIME

01

Key Takeaways in our lime model:

- LIME shows which words in EACH tweet drove the prediction
- Positive weights push toward predicted class
- Negative weights push away from predicted class
- This builds trust - stakeholders can see model reasoning

02

Business Applications:

- Customer Support: "Why was this flagged?" → LIME shows reason
- Model Validation: Check if model relies on right features
- Bias Detection: Identify if model uses inappropriate words
- User Communication: Explain automated decisions transparently

Conclusion & Final Recommendation

Project Achievement

- Delivered a sentiment classifier with 70.75 % accuracy
- +21.6 % improvement over baseline (58.2 %)
- Negative recall jumped from 10 % → 63 % — catches most real complaints

What Made the Difference

- Smart preprocessing + TF-IDF with bigrams
- Full GridSearchCV hyperparameter tuning
- Class weighting beat SMOTE — no synthetic data needed
- Focus on Negative recall as the business-critical metric

Current Limitations

- Cannot detect sarcasm or irony
- English-only
- Still misses ~37 % of negative tweets (acceptable for POC)

Production Readiness

- Model is fast, stable, and 100 % explainable
- Ready for immediate proof-of-concept deployment
- Use 70 % confidence threshold for automated flagging
- All other predictions → human review loop
- Retrain quarterly with fresh data

Final Recommendation: Deploy the tuned LinearSVC now as a real-time brand monitoring tool. It will catch emerging issues faster, amplify positive buzz, and give every decision a clear, transparent reason.

Thank you!