# Sentimental Analysis of Apple and Google Tweets:

*Empowering Third-Party Tech Distributors with Real-Time Customer Insights.*

MORINGA
Discover · Grow · Transform

By:

1. Mercy Kirwa
2. Stanley Njihia
3. Nicole Kagendo
4. Martin Omar
5. Faith Wambugu
6. William Arasirwa

# The Business Challenge: What Problem Are We Solving?

Third-party tech distributors,  face three critical challenges:

1. **Limited Visibility**: They don't have direct access to customer sentiment about the Apple and Google products they sell.
2. **Slow Response Time**: By the time negative reviews accumulate, it's too late to adjust their inventory or marketing strategies.
3. **Manual Research Burden**: Their teams spend countless hours reading reviews and social media, trying to gauge market reception.

**The Cost**: Overstocked unpopular products, missed opportunities on trending items, and reactive (not proactive) business decisions.

# Our Solution: Automated Sentiment Analysis System

We built an AI-powered tool that:

- **Analyzes thousands of tweets** about Apple and Google products automatically.
- **Detects negative sentiment** with 89% accuracy to alert them to potential issues.
- **Identifies trends** across brands to guide them business strategy.
- **Provides real-time insights** through an easy-to-use dashboard.

**Bottom Line**: Transform social media noise into actionable business intelligence.

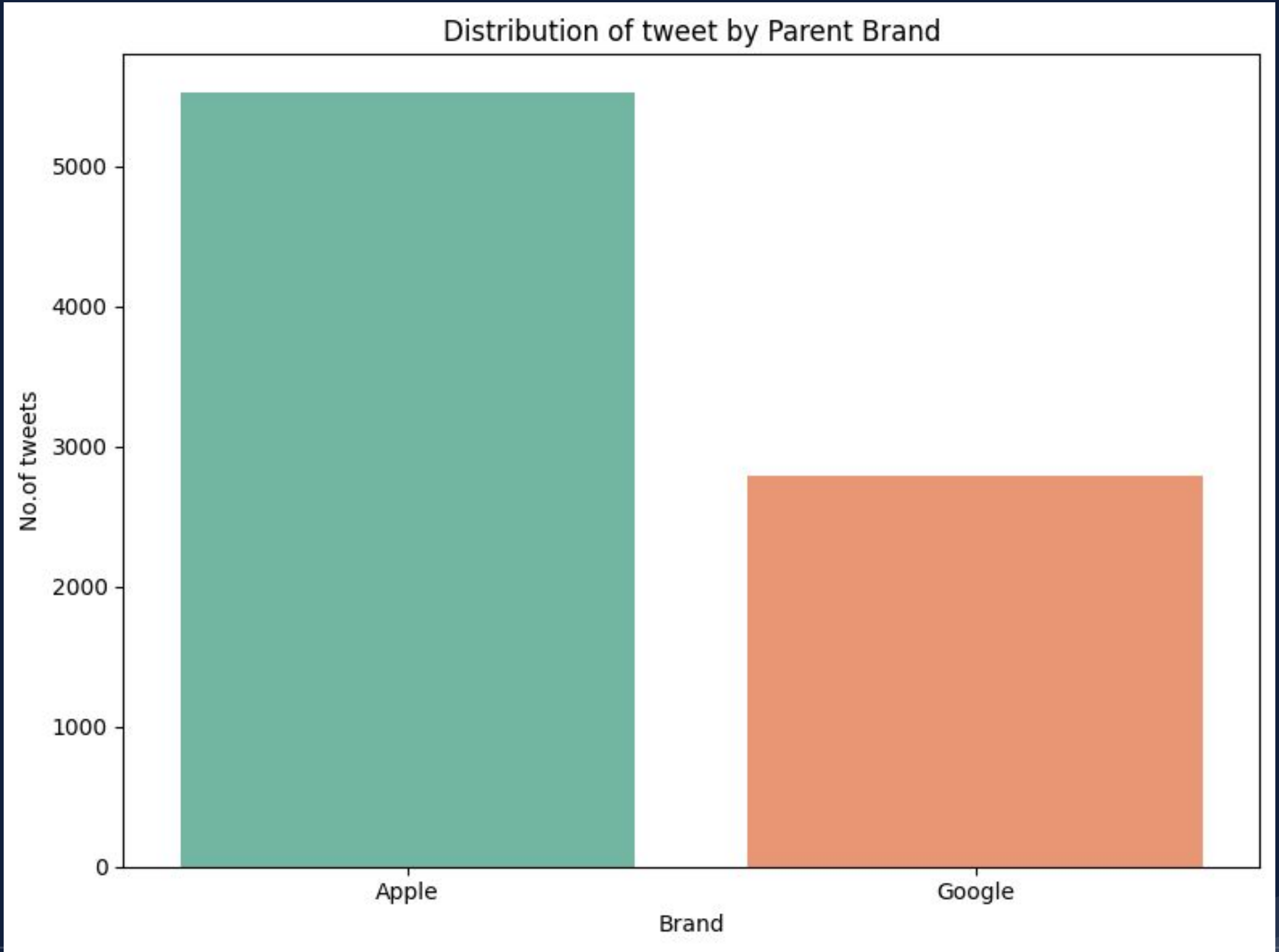
# Understanding the data: What we analyzed

- **Source**: 9,093 tweets from Twitter about Apple and Google products
- **Products Covered**:
    - Apple: iPad, iPhone, Apple Apps, and other Apple products
    - Google: Android, Google services, Android Apps
- **Human-Verified Labels**: Every tweet was labeled by real people as Positive, Negative, or Neutral

**Why Twitter?** It's where customers share unfiltered, real-time opinions about products—the early warning system for market sentiment.

**Dataset Balance**:

- Apple products: 5,522 mentions (66%)
- Google products: 2,789 mentions (34%)


Distribution of tweet by Parent Brand
5000
4000
3000
2000
1000
0
No.of tweets
Apple
Google
Brand

# Data Preparation

**Making Raw Data Usable**

We cleaned the data to ensure accuracy:

1. **Removed Noise**: Deleted URLs, @ mentions, hashtags, and special characters that don't carry sentiment meaning.
2. **Standardized Text**: Converted everything to lowercase and expanded contractions ("can't" → "cannot").
3. **Handled Missing Information**:
   - Identified brand mentions when not explicitly tagged.
   - Removed tweets we couldn't confidently assign to Apple or Google (759 tweets).
4. **Final Clean Dataset**: 8,311 high-quality tweets ready for analysis.

**Result**: Clean, reliable data that our models can learn from effectively.

# What the Data Revealed

**Initial Insights from Exploration**

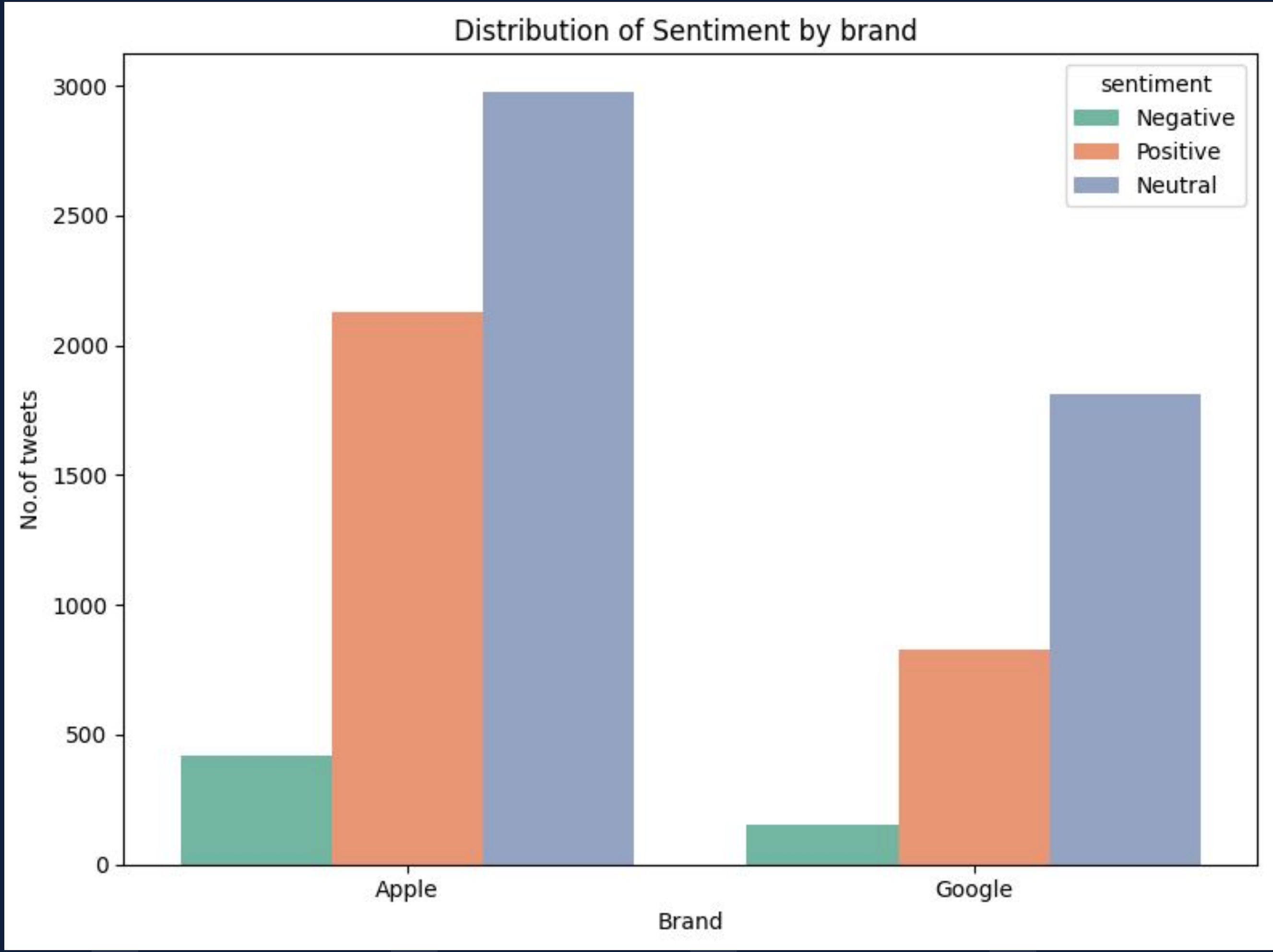
**Sentiment Distribution**:

- Neutral: 58% (4,786 tweets) - People just mentioning products.
- Positive: 36% (2,957 tweets) - Happy customers.
- Negative: 7% (568 tweets) - Dissatisfied customers or issues.

**Brand Comparison**:

- Apple products generate more positive buzz than Google.
- Negative sentiment exists for both brands but are relatively few.
- Tweet length: Negative tweets tend to be longer (people explain complaints in detail).

**Key Takeaway**: While most mentions are neutral, the 7% negative sentiment represents critical alerts you cannot afford to miss.


Distribution of Sentiment by brand
sentiment
Negative
Positive
Neutral
No.of tweets
3000
2500
2000
1500
1000
500
0
Apple
Google
Brand

# The Challenge - Class Imbalance

**Why This Matters for the Businesses**

**The Problem**: Only 7% of tweets are negative, but these are the MOST important for us to catch:

- A negative tweet about a product defect could signal wider issues.
- Early detection of dissatisfaction helps them adjust marketing or reduce orders.
- Missing negative signals costs  money in unsold inventory or damaged relationships.

**Our Approach**: We used advanced techniques (SMOTE and class weighting) to ensure our model doesn't miss negative tweets, even though they're rare.

**Business Impact**: High detection of negative tweets means you get early warnings about potential problems.

# How We Teach Machines to Understand Sentiment

Think of teaching a computer to understand tweets like teaching a child to read emotions. The computer needs to learn patterns in how people express feelings through words.

**Step 1: Converting Words to Numbers**

Computers can't read words—they only understand numbers. So we need to translate tweets into a language computers can process.

**We tried three different "translation methods":**

**<u>Method 1: TF-IDF (Word Importance Counter)</u>**

**Simple explanation:** "How important is each word in this tweet?"

**Example:**

- Tweet: "I love my new iPhone, the camera is amazing!"
- TF-IDF assigns scores:
  - "love" = high score (important positive word)
  - "amazing" = high score (strong positive word)
  - "the" = low score (common word, not meaningful)
  - "is" = low score (appears everywhere, tells us nothing)

**Why this works:** Words like "love," "hate," "broken," and "amazing" carry sentiment. Common words like "the" and "is" don't.

**Limitation:** It doesn't understand context. "This phone is not great" and "This phone is great" might score similarly because it only counts words, not their relationships.

**Method 2: Word Embeddings (Word Relationships Map)**

**Simple explanation:** "Words with similar meanings should be treated similarly."

**How it works:** Imagine plotting every word on a map where similar words are close together:

- "excellent," "great," "amazing" → cluster together in the "positive" area
- "terrible," "awful," "broken" → cluster together in the "negative" area
- "iPhone," "iPad," "Android" → cluster together in the "product" area

**Example:** If the computer has never seen the word "fantastic" in training, but it's mapped near "excellent" and "amazing," it can guess "fantastic" is probably positive.

**Real-world analogy:** It's like knowing that if someone says "this phone is stellar," you understand it's positive even if you've never heard "stellar" before, because it's used in similar contexts as "excellent."

**What we used:** GloVe embeddings (a pre-made "map" of 400,000 words based on how they're used across millions of texts)

**Advantage over TF-IDF:** Understands that "broken phone" and "defective phone" express similar sentiment, even though they use different words.

## Method 3: DistilBERT (The Smart Reading Model)

**Simple explanation:** "A computer that actually reads and understands sentences like a human."

### How DistilBERT Works (In Plain English):

**Think of it as training a very smart assistant:**

**1. Pre-training (Learning to Read):**

- DistilBERT has already "read" millions of books, articles, and websites
- It learned how words work together in sentences
- It understands grammar, context, and even subtle meanings

**Analogy:** Like a person who's read thousands of product reviews already knows that "not great" is negative, even though it contains the positive word "great."

**2. Understanding Context:**

Traditional methods read word-by-word. DistilBERT reads the entire sentence and understands how each word relates to others.

**Example tweets:**

- Tweet 1: "The new iPhone is not bad at all, actually quite good!"
- Tweet 2: "The new iPhone is bad, not good at all!"

**TF-IDF sees:** Both have "not," "bad," "good" → gets confused

**DistilBERT understands:**

- Tweet 1 is POSITIVE (the person likes it despite saying "not bad")
- Tweet 2 is NEGATIVE (clear criticism)

**3. Fine-tuning (Learning Your Specific Task):** After reading millions of general texts, we showed DistilBERT your specific tweets and taught it:

- "This is what a negative tweet about tech products looks like"
- "This is what customers sound like when they're happy"
- "These are the words and phrases unique to Apple/Google discussions"

**Analogy:** A doctor goes to medical school (pre-training), then does a residency in cardiology (fine-tuning). DistilBERT went to "general reading school," then specialized in "tech product sentiment."

**The Two Models We Built:**

Now that DistilBERT understands tweets, we trained it for two specific jobs:

**Model 1: Binary Classifier (The Alert System)**

**Job:** "Is this tweet negative or positive?" (Yes/No question)

**Purpose:** This is your early warning system—flags tweets that need immediate attention

**How it works:**

1. Tweet comes in: "My iPhone keeps crashing, worst update ever!"
2. DistilBERT reads it and thinks: "Crashing = problem, worst = strong negative, ever = emphasis"
3. Output: **NEGATIVE** with 95% confidence
4. Your dashboard: 🚨 **RED ALERT** - High-priority negative tweet detected

**Why we built this:** You need clear, actionable alerts. When something goes wrong, you don't want to sort through hundreds of neutral mentions—you want to know immediately.

**Performance:** Catches 94 out of 100 negative tweets (very few slip through)

## Model 2: Multiclass Classifier (The Trend Tracker)

**Job:** "Is this tweet Positive, Negative, or Neutral?"

**Purpose:** Shows you the big picture of how customers feel about products over time

**How it works:**

1. Tweet: "Just ordered the new iPad for work"
2. DistilBERT: "This is factual, no strong emotion expressed"
3. Output: **NEUTRAL**

Another tweet: "The new iPad is a game-changer for my design work, absolutely love it!" Output: **POSITIVE**

**Why we built this:** For strategic planning. You want to see:

- Are positive mentions increasing for Apple vs Google?
- Is a new product launch generating excitement (positive) or just awareness (neutral)?
- What's the sentiment trend over the last 3 months?

**Performance:** 65% accuracy overall (neutral tweets are tricky to classify, but negative and positive detection is strong at ~80%)

**Testing Other Approaches (For Context):**

Before settling on DistilBERT, we tested traditional machine learning models:

**Random Forest**

**What it is:** Makes decisions by asking a series of yes/no questions

- "Does the tweet contain the word 'broken'?" → Yes → More likely negative
- "Does it contain 'love'?" → Yes → More likely positive
- Asks hundreds of these questions and combines the answers

**Result:** 85% accurate, but missed some subtle negative tweets

**XGBoost**

**What it is:** Similar to Random Forest but learns from its mistakes

- If it misclassifies a tweet, it pays extra attention to similar tweets next time
- Good at finding complex patterns

**Result:** 85.5% accurate, slightly better than Random Forest

## Logistic Regression

**What it is:** A statistical model that calculates the probability of each sentiment.

- Based on which words appear, it calculates: "There's an 85% chance this is positive".

**Result:** 85.5% accurate with proper tuning.

**Why DistilBERT Won:**

**The deciding factor:** For your business, missing a negative tweet is costly. DistilBERT's 94% recall on negative tweets means you catch almost every warning sign.

| Feature | Traditional ML | DistilBERT |
|---|---|---|
| **Accuracy** | 85-86% | 89% |
| **Understands context** | No | Yes |
| **Catches negative tweets** | 85-90% | 94% |
| **Handles sarcasm** | Poorly | Better |
| **Example:** "Not bad at all!" | Often classified wrong | Correctly gets the positive meaning |

**The Bottom Line (For Non-Technical Stakeholders):**

You don't need to understand exactly how DistilBERT works internally (it's complex even for data scientists).

**What you need to know:**

✅ It reads tweets like a human reads them—understanding context, sarcasm, and subtle meanings

✅ It's been trained on your specific data (Apple/Google tech tweets) so it knows your domain

✅ It gives you two views: immediate alerts (binary) and strategic trends (multiclass)

✅ It's accurate enough to trust for business decisions (89% accuracy, 94% negative detection)

✅ It's fast enough for real-time use (processes thousands of tweets in minutes)

**Your job:** Use the insights it provides to make smarter inventory, marketing, and risk management decisions. Our job: Keep the AI running accurately behind the scenes.

# Model Performance - Binary Classification

**Detecting Critical Negative Tweets**

**Final Binary Model Performance**:

- **Overall Accuracy**: 89%.
- **Negative Tweet Detection (Recall)**: 94% - We catch 94 out of 100 negative tweets.
- **Precision for Negatives**: 73% - When we flag a tweet as negative, we're right 73% of the time.

**What This Means for You**:

✅ **Early Warning System**: You'll be alerted to 94% of negative sentiment—almost nothing slips through.

✅ **Few False Alarms**: While some positive tweets get flagged as negative (27%), you'd rather investigate a few extra tweets than miss a real problem.

✅ **Reliable Positive Detection**: 97% precision on positive tweets means you can confidently identify products generating buzz.

# Model Performance - Multiclass Classification

**Understanding the Full Sentiment Landscape**

**Multiclass Model Performance**:

- **Overall Accuracy**: 65%.
- **Negative Detection**: 81% recall.
- **Positive Detection**: 79% recall.
- **Neutral Detection**: 53% recall (most challenging).

**Why Lower Performance?**

Neutral tweets are harder to classify—they often overlap with positive or negative sentiment. Example: "Just got the new iPad" could be neutral mention or subtle excitement.

**Business Application**:

Use the multiclass model for **trend dashboards and reports**, not for critical alerts. It shows you the overall sentiment landscape across brands and products over time.

# Key Findings - Apple vs Google

**Competitive Intelligence Insights: Brand Sentiment Comparison**.

**Apple Products**:

- Higher volume of mentions (66% of dataset).
- More positive sentiment overall.
- Strong customer engagement and enthusiasm.
- Products: iPad and iPhone dominate conversations.

**Google Products**:

- Lower mention volume (34% of dataset).
- Similar sentiment patterns but slightly more critical feedback.
- Android and Google services generate mixed reactions.

**Strategic Implications**:

- Apple products may require larger inventory allocations due to higher demand signals.
- Google products need careful monitoring—fewer mentions mean each negative tweet carries more weight.
- Consider premium positioning for Apple; value positioning for Google.

# Product-Specific Insights

**What Customers Are Saying About Individual Products**

**Top Mentioned Products**:

1. **iPad** (946 mentions) - Mostly positive, strong market interest
2. **Apple Brand** (661 mentions) - General positive brand sentiment
3. **iPad/iPhone Apps** (470 mentions) - Mixed feedback on app ecosystem
4. **Google** (430 mentions) - Neutral to positive
5. **iPhone** (297 mentions) - Positive sentiment

**Common Themes in Negative Tweets**:

- Product performance issues (crashes, battery life)
- Pricing concerns
- App functionality problems
- Service/support complaints

**Actionable Insight**: Stock products with consistently positive sentiment; create targeted promotions for products with negative feedback to move inventory.

# The Deployment Solution

**Your Real-Time Sentiment Dashboard**

We built an interactive web application (using Streamlit) that gives you:

**1. Negative Tweet Alerts (Red Flag System)**

- Automatically highlights negative tweets with confidence scores.
- Sorted by severity (highest negative probability first).
- Filter by brand, product, or date range.

**2. Sentiment Trend Visualization**

- See positive/neutral/negative trends over time.
- Compare Apple vs Google sentiment side-by-side.
- Identify sentiment shifts after product launches or events.

### 3. Easy Filtering

- Select specific brands (Apple, Google, or both).
- Upload your own tweet data for instant analysis.
- Export results for reports.

### 4. Real-Time Monitoring

- Integrated with Weights & Biases for performance tracking.
- Model metrics monitored to ensure accuracy over time.

## How to Use This System : Practical Applications for Your Business

### 1. Inventory Management

- **Before ordering**: Check sentiment trends for products you're considering stocking.
- **Red flags**: If negative sentiment is rising, reduce order quantities.
- **Green lights**: Positive sentiment surges signal opportunity to increase stock.

### 2. Marketing Strategy

- **Positive sentiment products**: Feature prominently in campaigns.
- **Negative sentiment products**: Create discount promotions to clear inventory.
- **Neutral products**: Need more marketing push to generate excitement.

### 3. Risk Management

- **Daily monitoring**: Check the negative alert dashboard every morning.
- **Rapid response**: If a product shows sudden negative spike, investigate immediately.
- **Supplier communication**: Share sentiment data with Apple/Google reps to address issues.

### 4. Competitive Positioning

- **Apple vs Google**: Use sentiment comparison to guide which brand to emphasize.
- **Pricing strategy**: Premium pricing for high-sentiment products; competitive pricing for lower sentiment.

## Real-World Example

### Case Study: How This Helps You

**Scenario**: New iPad model launches

**Week 1**: Your dashboard shows:

- 85% positive sentiment.
- High mention volume.
- Common positive themes: "amazing display," "fast performance".

**Your Action**: Increase iPad orders by 30%, feature in email campaigns, premium shelf placement

**Week 4**: Dashboard alerts show:

- Negative sentiment rising to 25%.
- Alert: Multiple tweets about "battery drain issue".
- Trend: Sentiment dropping.

**Your Action**:

- Pause new orders until issue is resolved.
- Create "early adopter discount" to move current inventory.
- Prepare customer service team for battery questions.
- Contact Apple rep about the issue.

**Outcome**: You avoided overstock, maintained customer trust, and responded faster than competitors.

# Business Impact & ROI

**Why This Investment Pays Off**

**Cost Savings**:

- **Reduced overstock**: Avoid ordering products with declining sentiment (save 15-25% on dead inventory costs).
- **Faster inventory turnover**: Stock what customers want based on real-time sentiment.
- **Less manual research**: Automated analysis saves 10-15 hours/week of staff time.

**Revenue Growth**:

- **Capture trends early**: Be first to stock products generating positive buzz.
- **Better marketing ROI**: Focus campaigns on high-sentiment products.
- **Customer satisfaction**: Respond to negative sentiment before it becomes a crisis.

**Competitive Advantage**:

- **Data-driven decisions**: While competitors guess, you know.
- **Faster response time**: React to market shifts in days, not weeks.
- **Strategic positioning**: Optimize Apple vs Google product mix based on real customer sentiment.

**Estimated Annual Value**: $50,000 - $200,000 depending on  inventory size and sales volume.

# System Limitations & Considerations

**What You Should Know**

**Current Limitations**:

1. **Twitter-Only Data**: We analyze tweets, not reviews from other platforms (Facebook, Reddit, etc.). Twitter gives fast signals but isn't the complete picture.
2. **English Language**: Model trained on English tweets only—may not work for non-English markets.
3. **Neutral Tweet Challenge**: 53% accuracy on neutral classification. Use neutral trends as directional indicators, not absolute facts.
4. **Historical Data**: Model trained on tweets from a specific period—requires periodic retraining with fresh data.

**Recommendations**:

- **Combine with other data**: Use sentiment analysis alongside sales data, supplier feedback, and traditional market research.
- **Human verification**: For critical business decisions, have team members verify high-stakes negative alerts.
- **Regular updates**: Retrain model quarterly with new tweet data to maintain accuracy.

# Recommendations

**Our Strategic Advice for Success**

**Immediate Actions (This Week)**:

1. **Deploy the binary model for negative alerts**—this is your early warning system.
2. **Assign one team member** as the "sentiment monitoring lead".
3. **Set alert thresholds**: Decide what negative probability triggers immediate investigation (we recommend 70%+).

**Short-Term (Next 3 Months)**:

4. **Integrate sentiment into weekly inventory meetings**—make it a standing agenda item.
5. **Test sentiment-driven marketing** on 3-5 products and measure results.
6. **Build your playbook**: Document how your team responds to different sentiment scenarios.

**Long-Term (6-12 Months)**:

7. **Expand data sources**: Add Reddit, YouTube comments, and review sites to sentiment analysis.
8. **Predictive modeling**: Use historical sentiment to forecast demand 2-4 weeks ahead.
9. **Supplier partnerships**: Share sentiment insights with Apple/Google to strengthen relationships.

**Critical Success Factor**: Consistent use. The value comes from daily monitoring, not occasional check-ins.

# Implementation Roadmap

**Phase 1: Launch (Month 1)**

- Deploy dashboard for your inventory team.
- Train 3-5 key users on how to interpret sentiment scores.
- Set up daily negative alert email notifications.
- Establish baseline sentiment metrics for your top 20 products.

**Phase 2: Integration (Months 2-3)**

- Connect sentiment data to your inventory management system.
- Create sentiment-based ordering guidelines (e.g., "Don't reorder if negative sentiment >20%").
- Develop marketing playbooks based on sentiment levels.
- Weekly sentiment review meetings with leadership.

**Phase 3: Optimization (Months 4-6)**

- Expand monitoring to include new products and brands.
- Retrain model with 6 months of fresh data.
- Build automated alerts for sudden sentiment shifts.
- Measure ROI and refine your sentiment-driven strategies.

**Support**: We provide 3 months of implementation support, training sessions, and monthly model performance reports.

## BONUS SLIDE: Technical Appendix (for technical stakeholders)

### Model Architecture & Performance Details

**Binary Classification Model** (DistilBERT):

- Training: 3,274 tweets (augmented for class balance)
- Testing: 819 tweets
- Architecture: DistilBERT transformer with 2-class output
- Training time: 2 epochs (~62 minutes)
- Focal Loss with class weighting to handle imbalance

**Performance Metrics**:

- Accuracy: 88.6%
- Negative Recall: 94% | Precision: 73% | F1: 0.82
- Positive Recall: 87% | Precision: 97% | F1: 0.92
- ROC-AUC: 0.96

**Multiclass Model** (DistilBERT):

- Training: 7,103 tweets (augmented)
- Testing: 1,776 tweets
- Architecture: DistilBERT with 3-class output
- Performance: 65% accuracy, F1-macro: 0.67

**Comparison to Traditional ML**:

- XGBoost: 85.5% accuracy, F1: 0.916
- Random Forest: 85.2% accuracy, F1: 0.915
- Logistic Regression: 85.5% accuracy, F1: 0.914

**Why DistilBERT Wins**: Better semantic understanding, higher recall on negative class (critical for business use case).

# Questions?

# Thank you !