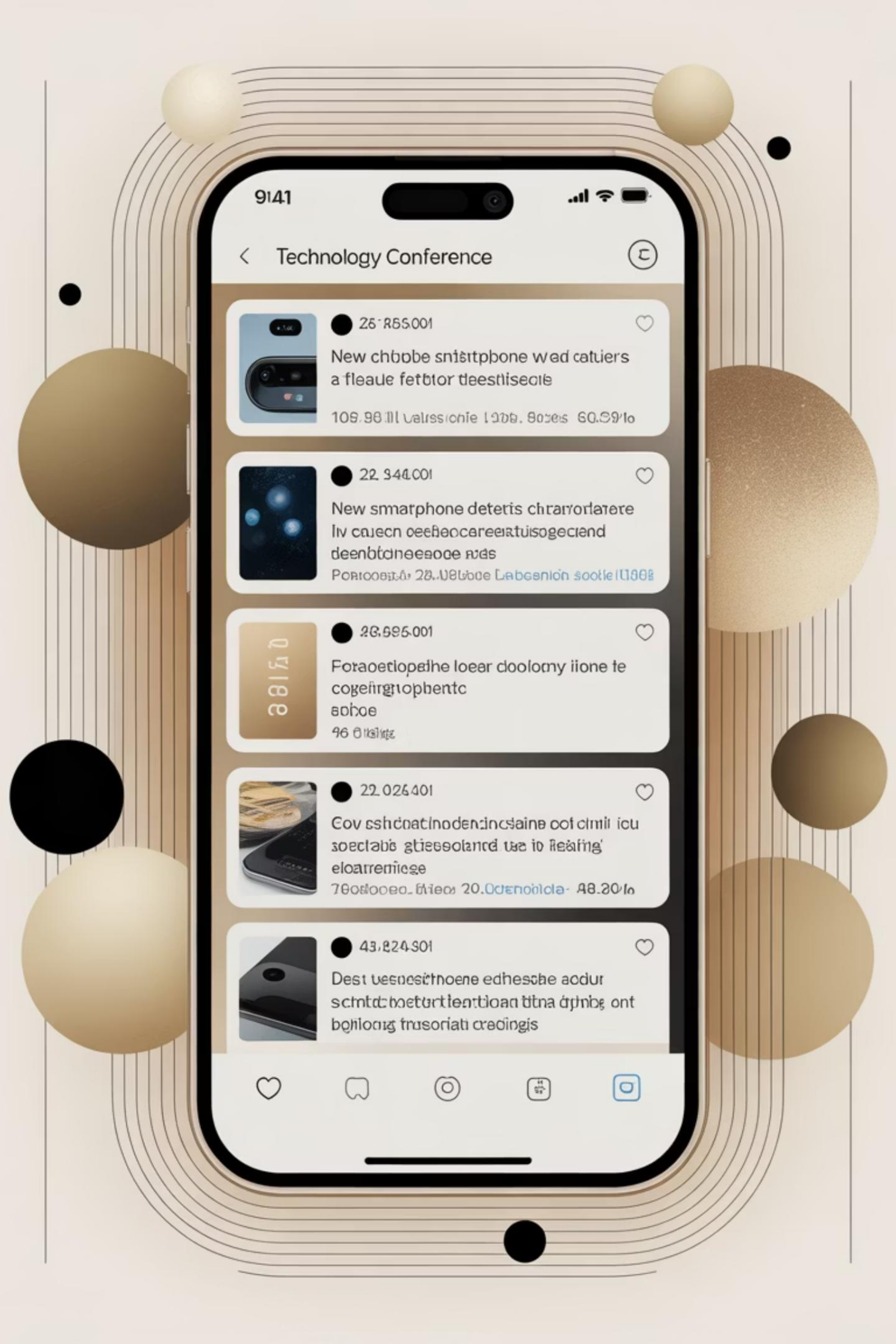


Sentiment Analysis of SXSW Tech Tweets

Natural Language Processing & Machine Learning Project

By: Lilibeth Langat
Jeniffer Mbugua
Lydia Khisa
Vicky Gakuo
Kefa Mwai

Introduction: The Power of Sentiment Analysis

Unstructured Text Challenge

Social media platforms generate massive volumes of unstructured text daily, creating opportunities for automated analysis

Public Opinion Insights

Sentiment analysis enables organizations to understand public opinion, track brand perception, and identify emerging trends in real-time

Automated Classification

Natural Language Processing and Machine Learning techniques automate sentiment classification, replacing manual annotation

Problem Statement

1 Manual Analysis Inefficiency

Manual sentiment analysis becomes impractical with thousands of tweets, requiring excessive time and resources

2 Noisy Data Challenge

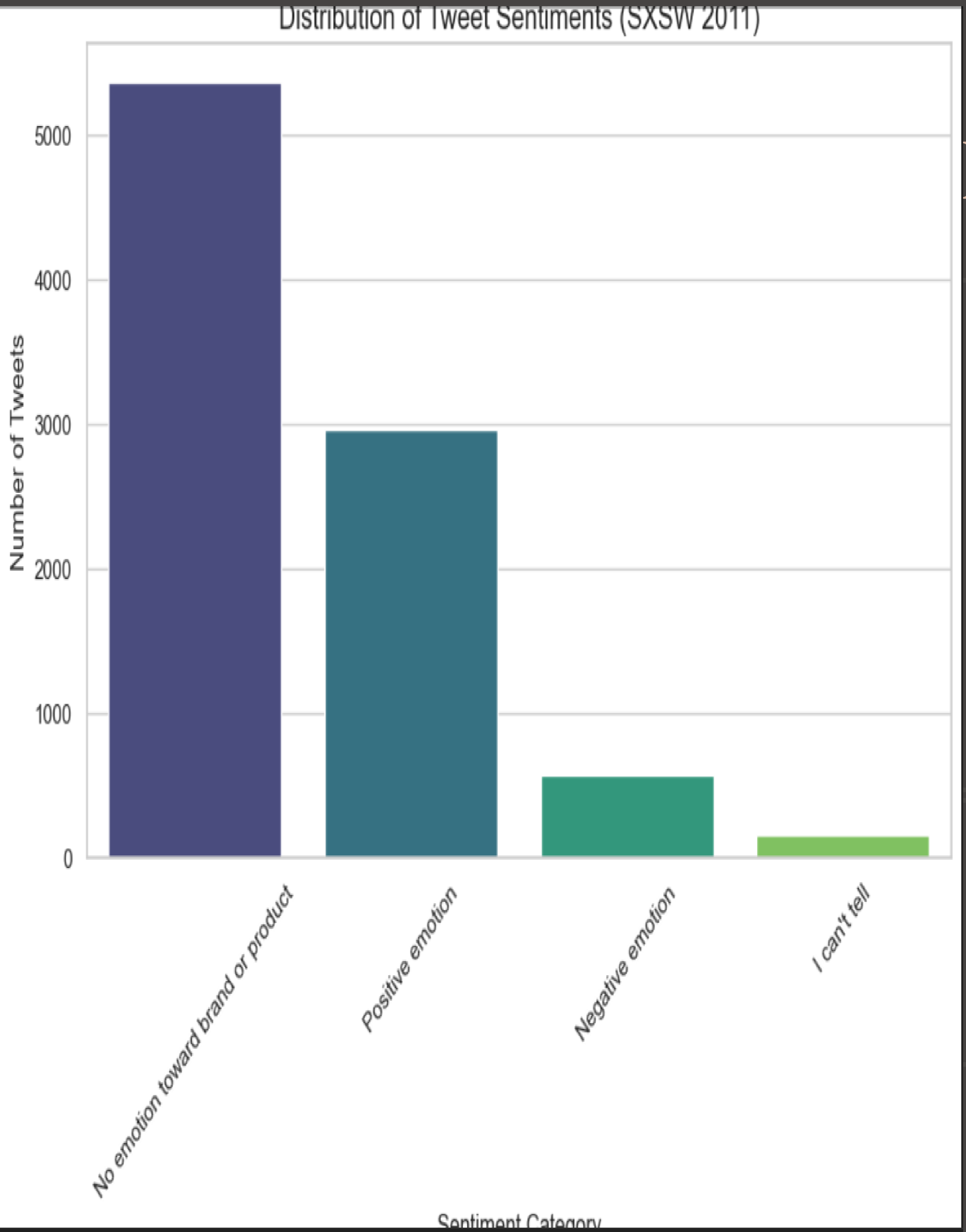
Social media data contains emojis, hashtags, slang, and typos that complicate automated processing

3 Need for Automation

Automated sentiment classification systems are essential for real-time analysis at scale

Research Focus

Developing ML models to classify SXSW technology tweet sentiment accurately despite class imbalance



Dataset Overview

Source Data

Tweets mentioning technology products during SXSW conference events

Key Features

- tweet_text: Full tweet content
- product_target: Mentioned product
- sentiment: Labeled sentiment class

Sentiment Classes

Multiple sentiment categories: positive, negative, and neutral classifications present

Exploratory Data Analysis

Key Findings

Sentiment Distribution

Analyzed frequency of each sentiment class across the dataset

Class Imbalance

Identified significant imbalance among sentiment categories

Positive Dominance

Positive sentiment substantially outnumbers negative and neutral classes

```
Unique Sentiment Labels:
sentiment
No emotion toward brand or product    5373
Positive emotion                      2968
Negative emotion                       569
I can't tell                           156
Name: count, dtype: int64
```

Text Preprocessing Pipeline

Normalization

Convert all text to lowercase for consistency

Cleaning

Remove punctuation, numbers, and special characters

Stopword Removal

Filter out common words with minimal semantic value

Tokenization

Split text into individual words or tokens

Lemmatization

Reduce words to their base dictionary form

Feature Engineering: TF-IDF Vectorization

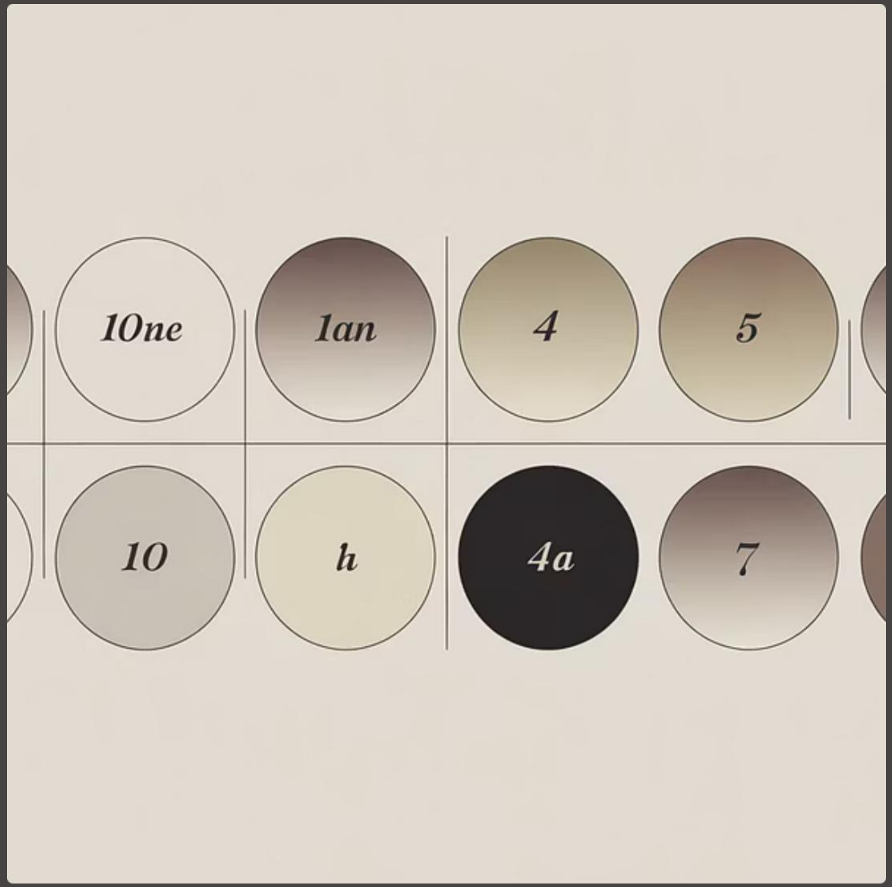
Methodology

- Term Frequency-Inverse Document Frequency technique
- Converts text documents into numerical feature vectors
- Weights terms by importance in document vs. corpus
- Higher scores for terms frequent in document but rare overall
- Preserves semantic meaning through weighted representation

Implementation

Text data transformed into high-dimensional numerical vectors suitable for ML algorithms while maintaining word importance signals

Machine Learning Models

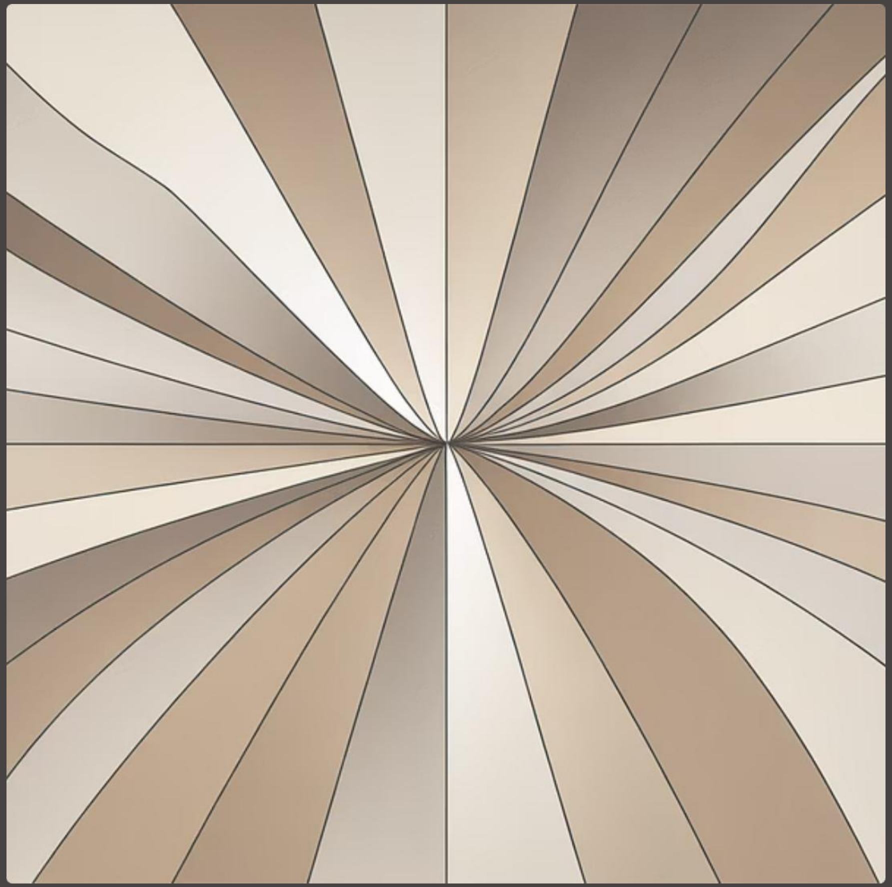


Multinomial Naive Bayes

Baseline probabilistic classifier using Bayes' theorem with frequency-based features

Random Forest

Ensemble method combining multiple decision trees for improved accuracy and robustness



XGBoost

Gradient boosting framework with optimized tree-based models for enhanced performance

Model Training and Evaluation

01

Data Splitting

Divided dataset into training and testing sets for model validation

02

Hyperparameter Tuning

Applied GridSearchCV to optimize model parameters systematically

03

Performance Metrics

Evaluated using precision, recall, and F1-score across all sentiment classes

04

Confusion Matrix

Analyzed classification errors and misclassification patterns

05

Class Imbalance

Accounted for uneven class distribution during evaluation

Results and Future Directions

Key Findings

Best Model

Random Forest achieved highest overall performance across evaluation metrics

Ensemble Success

Tree-based ensemble methods outperformed baseline Naive Bayes classifier

Challenge Area

Negative sentiment prediction proved most difficult due to class imbalance

Future Work

Data Balancing

Apply oversampling or undersampling techniques to address class imbalance

Deep Learning

Explore neural network architectures like LSTM or BERT for enhanced performance