Sentiment Analysis of Tweets Using NLP

An End-to-End NLP Project (Group 5)

Problem Statement

- This project aims to build a sentiment classification model using a dataset of tweets relating to Apple and Google Products applying Natural Language Processing (NLP) techniques
  - Dataset: ~9000 tweets

- Value Proposition for this model:
  - Understanding Customer Sentiment
  - Monitoring Brand Reputation
  - Getting Product Specific Insights

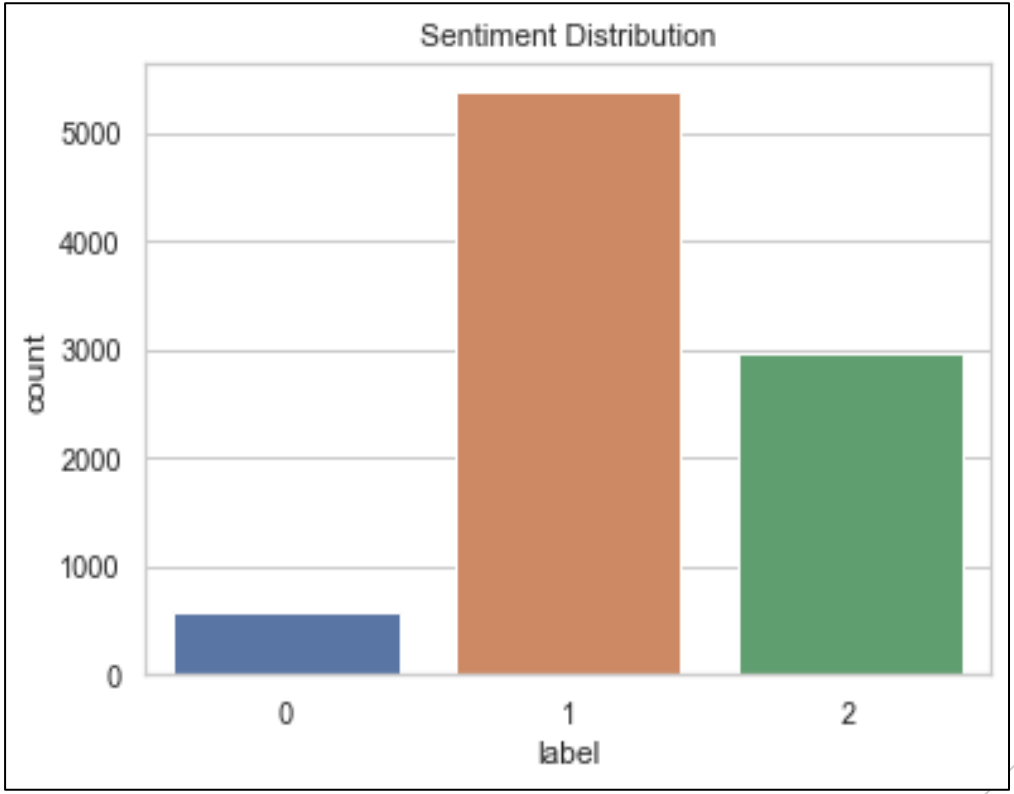
Project Objectives

- **Convert:** unstructured tweet text into structured input for modeling by applying text preprocessing and feature engineering.
- **Develop:** an NLP-based classification model that categorizes tweet sentiments into **Positive**, **Neutral** or **Negative** using machine learning techniques.
- **Evaluate:** and compare multiple classification algorithms such as Naïve Bayes, Random Forest and XGBand optimize performance using techniques like SMOTE and GridSearchCV.

Data Methodology

- **Data Cleaning and Text Preprocessing:**
  - Removed null values, duplicates and columns
  - Lowercasing and removing special characters
  - Stopword removal
  - **Tokenization** using TweetTokenizer
  - **Lemmatization** with WordNetLemmatizer
- **Feature engineering**: Product Category, Chars, Words
- **Exploratory Data Analysis:** class distribution
- **Model Training & Evaluation:**
  - Baseline Model: RandomForest Classifier
  - Tuning: using TF-IDF, SMOTE, GridSearchCV
  - Benchmarking: XGBClassifier, Naïve Bayes
  - Evaluation metrics: Accuracy, F1 Score, ROC-AUC Score
- **Reporting:** Conclusions & Recommendations

EDA: A Case of Class Imbalance



Model Training and Evaluation

Models Trained:

- Random Forest Classifier (baseline)
- Ensemble methods: XGB Classifier
- Naïve Bayes Classifiers: Multinomial + Complement Naïve Bayes

Techniques used:

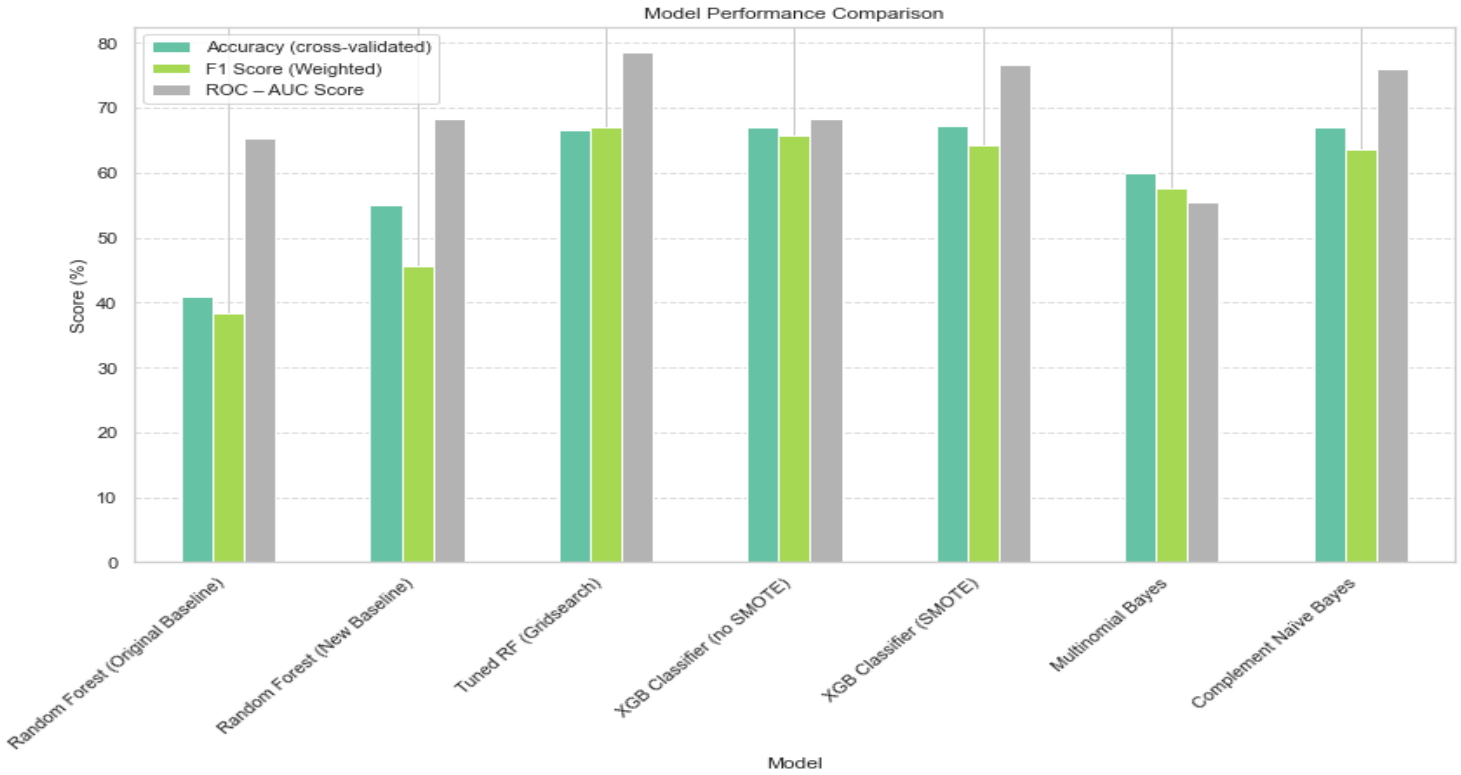
- Pipelines – for consistent preprocessing and modeling
- Vectorization techniques: CountVectorizer and TF-IDF
- SMOTE – to balance class distribution
- GridSearchCV – for hyperparameter tuning

Evaluation metrics: *(Cross-validated) Accuracy, Weighted F1 Score, ROC-AUC Score*

Results Comparison

Model	Accuracy (cross-validated)	F1 Score (Weighted)	ROC – AUC Score
Random Forest (Original Baseline)	40.88%	38.42%	65.38%
Random Forest (New Baseline)	55.09%	45.53%	68.19%
Tuned RF (Gridsearch)	66.66%	66.94%	78.53%
XGB Classifier (no SMOTE)	67.06%	65.76%	68.19%
XGB Classifier (SMOTE)	67.25%	64.28%	76.64%
Multinomial Bayes	60.03%	57.56%	55.53%
Complement Naïve Bayes	67.06%	63.51%	76.06%

Model Comparison



Conclusion

Key Takeaways:

- XGBClassifier is the best overall model by Accuracy
- Tuned RF is the best by F1 Score and ROC-AUC Score: shows proof of good performance gains from hyperparameter tuning and feature engineering
- SMOTE Benefit: increase ROC-AUC Score
- Complement Naïve Bayes' performance is second only to XGBClassifier and Tuned RF

Challenges Faced:

- Handling noise and short text
- Highly Imbalanced classes

Business Recommendations (Next Steps)

- Use insights from positive tweets to identify what customers love and amplify those product features.
- Implement a live dashboard or API port that uses the trained model to monitor twitter sentiment continuously.
- Engage promptly with users who post negative sentiment tweets to resolve issues and protect brand reputation.

THANK YOU!

James wachira
Tim Musungu
Vivian Kwamboka
Calvin Mutua
Hashim Ibrahim