

WambuiMunene / Phase-4-Sentiment-Analysis-NLP-Project

<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Setting

A Natural Language Processing (NLP) model that rates the sentiment of tweets about Apple and Google products as positive, negative or neutral.

☆ 0 stars

🔗 0 forks

👁 1 watching

🌿 Branches

📈 Activity

🏷 Tags

🌐 Public repository

🔗 1 Branch

🏷 0 Tags

🔍 Go to file












Go to file

+

Add file

Code

...

 WambuiMunene	notebook.pdf created	301ffdb · 1 minute ago	
	.ipynb_checkpoints	pptx converted to pdf and more edi...	1 hour ago
	Presentation.pdf	pptx converted to pdf and more edi...	1 hour ago
	Presentation.pptx	pptx converted to pdf and more edi...	1 hour ago
	README.md	edited table formating in README file	1 hour ago
	desktop.ini	this is the intial commit	last week
	index.ipynb	pptx converted to pdf and more edi...	1 hour ago
	notebook.pdf	notebook.pdf created	1 minute ago
	sentiments_by_product.png	almost final edits on the README file	17 hours ago
	tweet_product_company.csv	this is the intial commit	last week

📖 README

Developing a NLP Model to Analyze and Classify the Sentiment of Tweets About Apple and Google Products as Positive, Negative, or Neutral.

Project Summary

Social media is a dynamic platform where customers express their thoughts about products, services, and brands. Analyzing sentiments from social media platforms like X (formerly Twitter) provides businesses with real-time insights into customer opinions and experiences.

Data Understanding

The objective of this project is to build a Natural Language Processing (NLP) model that classifies the sentiment of tweets about Apple and Google products as positive, negative or neutral, and in particular be able to pick out negative tweets with a high level of recall. The dataset used to build the model is sourced from CrowdFlower via data.world <https://data.world/crowdflower/brands-and-product-emotions>. This dataset consists of slightly over 9,000 human-rated tweets.

Features: prior to the preprocessing steps every row in the dataset only contains two feature columns; a string containing the full text of an individual tweet, and another string on the product being referred to in the tweet. During preprocessing the string of tweet text will be converted into individual words creating more features.

Target: the target consists of labels (emotions) for each tweets - positive, negative, neutral and 'can't tell'. By looking at the value counts for each sentiment, a decision was made to drop the 'can't tell' rows.

Problem Statement

Sentiment Analysis is a powerful tool that provides businesses with deep insights into public perception of their products and services. By leveraging sentiment analysis, companies can effectively gauge customer sentiment and understand the emotional tone behind customer interactions. This enables businesses to identify areas of concern in real-time, allowing them to proactively address customer needs and improve their offerings.

By analyzing these sentiments from the tweets about their products and that of their competitor, Apple can tap into a wealth of authentic feedback that traditional surveys or feedback forms might miss. This immediate access to customer sentiment will allow them to swiftly identify trends, preferences, and potential issues, allowing for proactive engagement and timely adjustments to strategies.

Business Objectives

1. **Goal:** Train classification models to identify sentiments (Positive, Neutral, Negative) about Apple and Google Products.
2. **Specific Objectives:**
 - Identify the distribution of negative and positive tweets by company.
 - Train, tune, and evaluate at least 3 classification models for sentiment analysis.
 - Provide the optimal model to Apple for identifying **negative** sentiments in future data.

Requirements to Meet Objectives

1. Load the Data

- Used `pandas` to load the dataset and inspect the data.

2. Perform Data Cleaning with `nltk` and `re` libraries

- Regular Expressions (REGEX) to remove irrelevant information such as URLs, mentions, and hashtags from the library `re`
- Convert all text to lowercase to ensure uniformity.
- Apply lemmatization to reduce words to their base forms using `WordNetLemmatizer` from `nltk`
- Remove stop words to focus on meaningful words using `stopwords` from `nltk.corpus`
- Tokenize the cleaned text.

3. Perform Exploratory Data Analysis

- Analyze positive and negative sentiments by company.
- Visualize the distribution of sentiment labels using bar charts and value counts.
- Visualize the top 10 most common words using `matplotlib` and `seaborn`
- Create word clouds for positive, negative, and neutral tweets using `wordcloud`

4. Vectorize the Text Data with `TfidfVectorizer`

- Use `TfidfVectorizer` from `sklearn.feature_extraction.text` to convert the text data into numeric form.

5. Iteratively Build and Evaluate Baseline and Ensemble Models

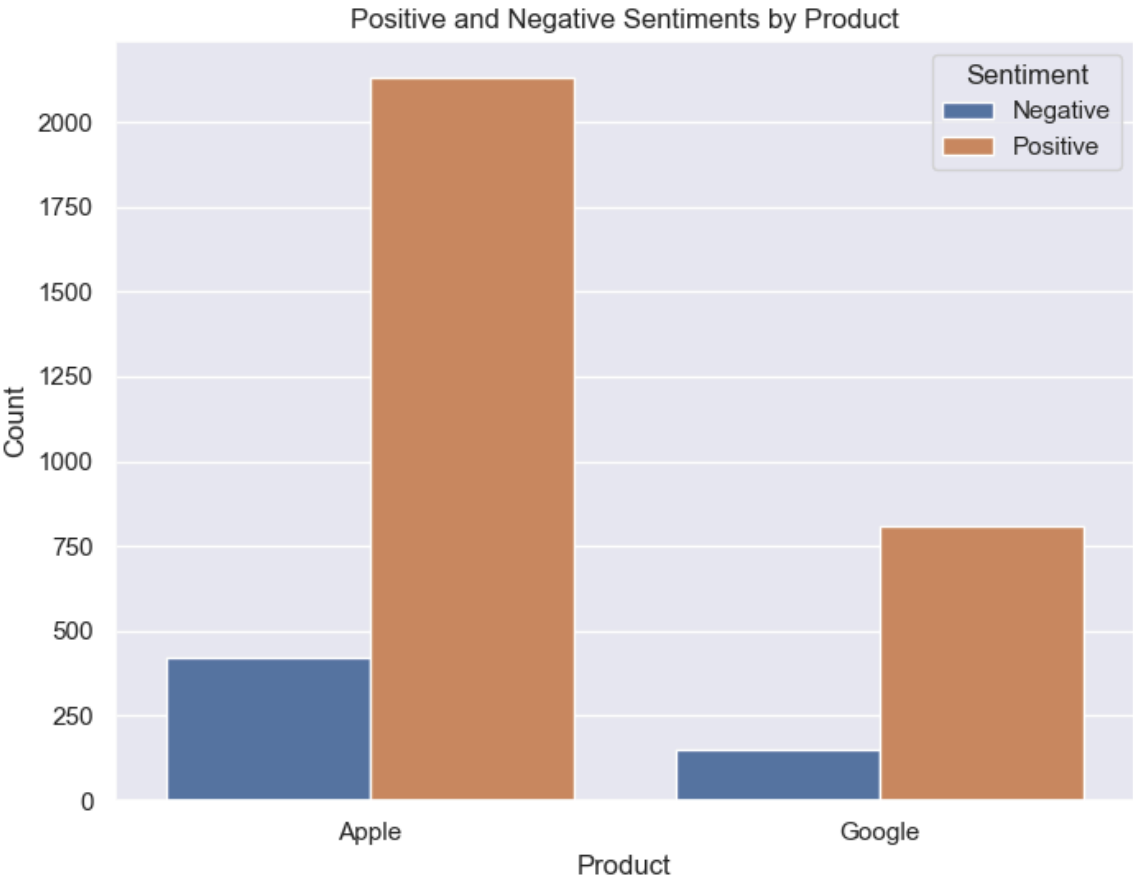
- Used `Pipelines` from `sklearn.pipeline` and `imblearn.pipeline` to build and tune `LogisticRegression` and `MultinomialNB`
- Build and train `RandomForestClassifier`, `AdaBoostClassifier` and `xgboost` ensemble models and compare results with the baseline models.
- Used `SMOTE` to oversample the minority class and `GridSearchCV` to identify the best parameters

6. Evaluation The models were evaluated using:

- `classification_report` and `confusion_matrix` from `sklearn.metrics` for model evaluation.

Objective # 1.

Sentiment Analysis and Competition Landscape



- **Popularity and Sentiment Balance:** Apple products are more popular but also have higher negative and positive sentiments. This is indicative of strong feelings about Apple products on both ends of the spectrum. It also implies that Apple has a strong and dedicated fan base and they should tap into this to get honest feedback about their products.
- **Strategy for Negative Sentiments:** Apple should monitor and address negative sentiments to maintain its market position and proactively address customer complaints by enhancing customer service, improving product quality, and engaging with users on social media.

Objective # 2

Evaluation of the Top 3 Best Performing Classification Models to Identify Positive, Neutral, and Negative Classes

Classification Report (Baseline Logistic Regression):

	precision	recall	f1-score	support
Negative	0.37	0.48	0.41	189.00
Neutral	0.74	0.72	0.73	1612.00
Positive	0.56	0.56	0.56	880.00
accuracy	0.65	0.65	0.65	0.65
macro avg	0.56	0.58	0.57	2681.00
weighted avg	0.66	0.65	0.65	2681.00



Classification Report (Tuned Logistic Regression):

	precision	recall	f1-score	support
Negative	0.42	0.39	0.40	189.00
Neutral	0.74	0.74	0.74	1612.00
Positive	0.57	0.58	0.57	880.00
accuracy	0.66	0.66	0.66	0.66
macro avg	0.57	0.57	0.57	2681.00
weighted avg	0.66	0.66	0.66	2681.00

classification Report (Random Forest):

	precision	recall	f1-score	support
Negative	0.66	0.21	0.32	189.00
Neutral	0.70	0.84	0.76	1612.00
Positive	0.60	0.48	0.53	880.00
accuracy	0.67	0.67	0.67	0.67
macro avg	0.65	0.51	0.54	2681.00
weighted avg	0.67	0.67	0.66	2681.00

Interpretation

- Baseline Logistic Regression Model showed an overall accuracy of 65%, while the tuned Logistic Regression and Random Forest Model had an accuracy of 66% and 67% respectively. Comparing the metrics for the negative class:
- Precision: Random Forest is better at avoiding false positives for negative tweets.
- Recall: Baseline Logistic Regression captures a higher percentage of actual negative tweets.
- F1-Score: Baseline Logistic Regression offers a balanced approach with better recall.

The Baseline Logistic Model, despite a slightly lower accuracy is the better model for identifying the three classes; it has the highest recall of the Negative Class which is hugely important to identify.

Sub-optimal performance on all the models trained can be attributed to class imbalance. Although SMOTE was used to oversample the minority class, the synthetic data did not significantly enhance model performance.

In our quest to develop a model with a higher recall for the negative class, we undertook the following steps:

- **Class Consolidation:** Neutral and Positive classes combined into a new class labeled 'Other'.
- **Resampling:** Built a model with a resampled subset of the new class.
- **Model Training:** Trained both baseline and tuned LogisticRegression models, along with three Ensemble models- RandomForestClassifier , AdaBoostClassifier and xgboost

Objective # 3**Evaluation of the Top 3 Best Performing Classification Models to Identify the Negative Class****Classification Report Baseline Logistic Regression:**

	precision	recall	f1-score	support
Negative	0.69	0.75	0.72	183.00
Other	0.78	0.73	0.76	228.00



accuracy	0.74	0.74	0.74	0.74
macro avg	0.74	0.74	0.74	411.00
weighted avg	0.74	0.74	0.74	411.00

Classification Report Tuned Logistic Regression:

	precision	recall	f1-score	support
Negative	0.71	0.73	0.72	183.00
Other	0.78	0.76	0.77	228.00
accuracy	0.74	0.74	0.74	0.74
macro avg	0.74	0.74	0.74	411.00
weighted avg	0.75	0.74	0.74	411.00

Classification Report Random Forest:

	precision	recall	f1-score	support
Negative	0.76	0.58	0.66	183.00
Other	0.72	0.86	0.78	228.00
accuracy	0.73	0.73	0.73	0.73
macro avg	0.74	0.72	0.72	411.00
weighted avg	0.74	0.73	0.73	411.00

After evaluating the performance of the different models, it is evident that the Baseline `LogisticRegression` model provides the highest recall score for the negative class at 75%, which is crucial for identifying negative sentiments accurately. The Tuned `LogisticRegression` model, while having slightly higher precision, has a recall of 73% for the negative class. The `RandomForestClassifier` model, although having higher precision for the negative class, has a significantly lower recall compared to both `LogisticRegression` models.

Given the focus of the business is to identify negative sentiments accurately, the Baseline `LogisticRegression` model with balanced classes is recommended. It achieves the highest recall for the negative class, ensuring a higher number of negative sentiments are accurately identified, and also performs quite all on the other classes ,providing the most balanced scores.

Next Steps

Deploy the Selected Model:

- Deploy the Baseline `LogisticRegression` model with balanced classes into a production environment.
- Use a pipeline for real-time or batch processing of new tweets to classify the sentiment efficiently.
- This involves setting up the infrastructure, such as cloud services or servers, and integrating the model into existing systems.

Continuous Model Monitoring and Improvement:

- Establish a system to monitor the performance of the deployed model regularly.
- Collect feedback, track key metrics like precision, recall, and F1-score, and analyze any drifts in data or model accuracy.
- Schedule periodic retraining of the model with new, better labeled data to improve its performance and relevance.

Develop a Sentiment Response Strategy:

- Based on the insights derived from the sentiment analysis, create a comprehensive strategy for responding to negative sentiments identified in the tweets.
- This could include setting up automated alert systems for negative sentiment spikes, and defining customer service protocols for addressing issues
- Develop content strategies to engage with customers and improve brand perception.

These steps will ensure that the sentiment analysis model continues to deliver valuable insights and helps Apple proactively address customer concerns.

Appendix

Libraries Used

1. Pandas, Numpy

- Purpose: Data manipulation and analysis.
- Usage: Loading datasets, cleaning data, and transforming data for analysis.

2. Matplotlib and Seaborn

- Purpose: Data visualization.
- Usage: Creating bar charts, value counts, and other visualizations to understand class balance.

3. nltk (Natural Language Toolkit) and REGEX

- Purpose: Text preprocessing.
- Usage: Tokenization, lemmatization, removing stop words, and text cleaning.

4. scikit-learn (sklearn),imblearn and xgboost

- Purpose: Machine learning and model evaluation.
- Usage: Building and evaluating models, including Logistic Regression, Naive Bayes, and ensemble models. Metrics such as classification_report and confusion_matrix.

5. WordCloud and Counter

- Purpose: Text visualization.
- Usage: Creating word clouds to visualize the most common words in each sentiment class.

```
import pandas as pd
import numpy as np
```

```
import matplotlib.pyplot as plt
import seaborn as sns
```

```
import re
import nltk
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
```



```
from imblearn.pipeline import Pipeline
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import LabelEncoder
from sklearn.linear_model import LogisticRegression
```



Releases

No releases published

[Create a new release](#)

Packages

No packages published

[Publish your first package](#)

Languages

● Jupyter Notebook 100.0%