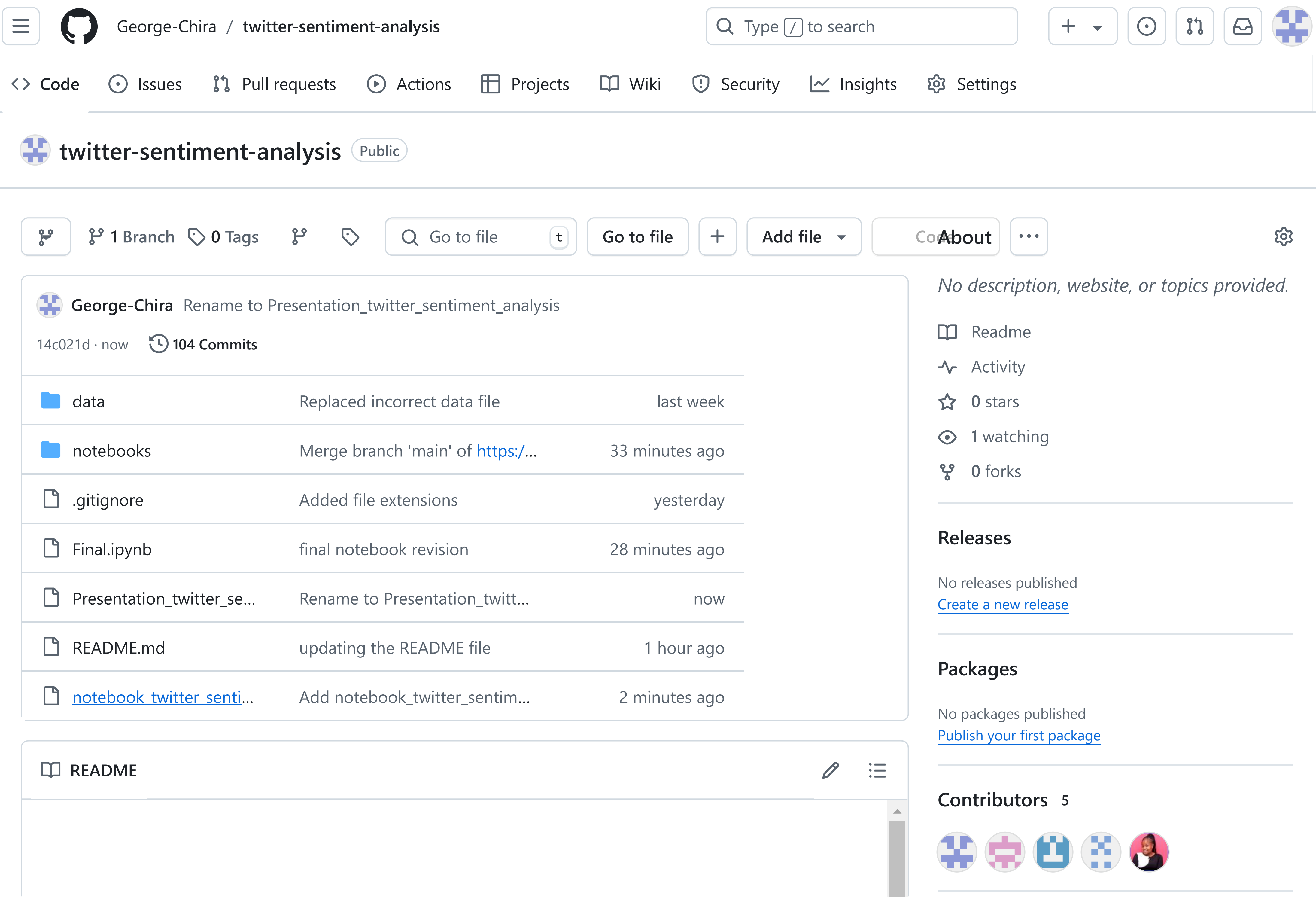


George-Chira Rename to Presentation_twitter_sentiment_analysis

14c021d · now 104 Commits

data	Replaced incorrect data file	last week
notebooks	Merge branch 'main' of https:/...	33 minutes ago
.gitignore	Added file extensions	yesterday
Final.ipynb	final notebook revision	28 minutes ago
Presentation_twitter_se...	Rename to Presentation_twitt...	now
README.md	updating the README file	1 hour ago
notebook_twitter_senti...	Add notebook_twitter_sentim...	2 minutes ago

README

No description, website, or topics provided.

Readme

Activity

0 stars

1 watching

0 forks

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

Contributors 5

SentimentFlow: Sentiment Analysis of Twitter Data for Apple and Google Products

Overview

SentimentFlow aims to solve the problem of understanding public sentiment towards Apple and Google products on Twitter. By analyzing tweets, the project provides valuable insights for companies, marketing teams, and decision-makers who want to gauge public opinion and make informed strategic decisions.

Business Understanding

Problem Statement

The main objective is to accurately classify the sentiment of tweets related to Apple and Google products into positive, negative, or neutral categories. This classification can help companies understand customer satisfaction, identify potential issues, and tailor their responses accordingly.

Stakeholders

1. **Companies (Apple and Google)**: Directly impacted by public sentiment, they want to monitor their product's perception and identify areas for improvement.
2. **Marketing Teams**: Use sentiment analysis to adjust campaigns, respond to negative feedback, and highlight product strengths.
3. **Decision-Makers**: Require insights into public sentiment for informed decisions on product development, customer support, and brand management.

Value Proposition

Languages

● **Jupyter Notebook** 100.0%

By accurately classifying tweets, the NLP model provides actionable insights, such as:

- Identifying negative sentiment to address issues promptly.
- Recognizing positive sentiment to reinforce successful strategies.
- Understanding neutral sentiment for balanced context.

Objectives

Main Objective

- Develop a Natural Language Processing (NLP) multiclass classification model for sentiment analysis, targeting an accuracy and recall score of 80% or higher.

Specific Objectives

1. Identify the most common words in the dataset using word clouds.
2. Confirm frequently used words with positive and negative sentiment tags.
3. Recognize products mentioned by users.
4. Examine the sentiment distribution across the dataset.

Data Understanding

Data Sources

The dataset originates from CrowdFlower via data.world, containing over 9,000 labeled tweets. Each tweet is labeled as expressing positive, negative, or no emotion toward a brand or product. The data is available in the `data` folder of this repository.

Suitability of Data

- **Relevance**: Aligns with the goal of understanding Twitter sentiment for Apple and Google products.
- **Real-World Context**: Represents actual user opinions, making it highly applicable.
- **Multiclass Labels**: Enables building both binary and multiclass classifiers.

Data Exploration

- **Key Features**: The dataset includes `tweet_text` , `is_there_an_emotion_directed_at_a_brand_or_product` , and `emotion_in_tweet_is_directed_at` .
- **Challenges**: Addressing label noise, class imbalance, contextual limitations, and missing data.

Data Cleaning

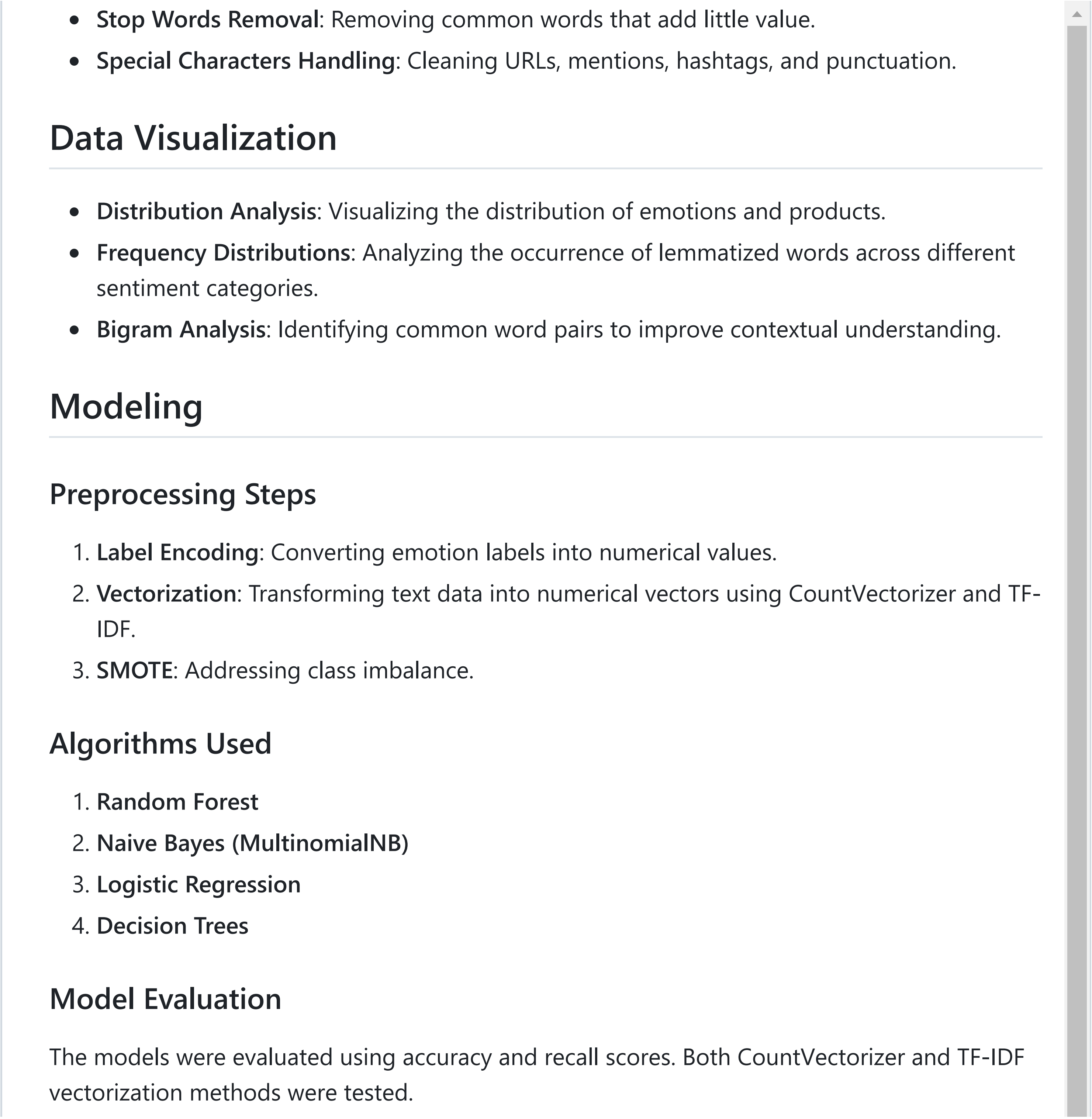
The data cleaning process included:

1. **Corrupted Records Removal**: Detecting and removing corrupted records.
2. **Handling Missing Values**: Dropping or filling missing values using relevant techniques.
3. **Class Imbalance Resolution**: Applying SMOTE (Synthetic Minority Oversampling Technique) to address class imbalance.
4. **Column Renaming and Consistency Checks**: Ensuring column names are uniform and data types are correct.

Data Preprocessing

Text preprocessing included:

- **Tokenization and Lemmatization**: Splitting text into words and reducing them to base forms.



- **Stop Words Removal**: Removing common words that add little value.
- **Special Characters Handling**: Cleaning URLs, mentions, hashtags, and punctuation.

Data Visualization

- **Distribution Analysis**: Visualizing the distribution of emotions and products.
- **Frequency Distributions**: Analyzing the occurrence of lemmatized words across different sentiment categories.
- **Bigram Analysis**: Identifying common word pairs to improve contextual understanding.

Modeling

Preprocessing Steps

1. **Label Encoding**: Converting emotion labels into numerical values.
2. **Vectorization**: Transforming text data into numerical vectors using CountVectorizer and TF-IDF.
3. **SMOTE**: Addressing class imbalance.

Algorithms Used

1. **Random Forest**
2. **Naive Bayes (MultinomialNB)**
3. **Logistic Regression**
4. **Decision Trees**

Model Evaluation

The models were evaluated using accuracy and recall scores. Both CountVectorizer and TF-IDF vectorization methods were tested.

Key Findings

1. **TF-IDF Vectorization** consistently outperformed CountVectorizer, providing superior feature representation.
2. **Random Forest and Logistic Regression** achieved the highest accuracy (83.7%) and recall (83.6%) scores.
3. **Hyperparameter Tuning** significantly improved model performance, increasing accuracy and recall by more than 10% in some cases.
4. **Class Imbalance Handling with SMOTE** ensured balanced model performance across all sentiment categories.

Recommendations

1. **Monitor Negative Sentiments**: Implement real-time alerts for prompt issue resolution.
2. **Scalability**: Optimize models for handling large-scale data in production environments.
3. **Real-Time Processing**: Explore real-time sentiment analysis for timely decision-making.
4. **Continuous Model Monitoring**: Regularly retrain models with new data to maintain accuracy.
5. **Integration with Social Media APIs**: Enable continuous monitoring and real-time insights.

Conclusion

Summary of Findings

- The project successfully evaluated various machine learning models for sentiment classification.
- **TF-IDF Vectorization** consistently provided the best feature representation.
- **Tuned Logistic Regression and Random Forest models** achieved high accuracy and recall scores.

- **Hyperparameter tuning and SMOTE** were effective in improving model performance and addressing class imbalance.

Future Work

1. **Enhance Real-Time Capabilities**: Implement streaming data analysis for live sentiment tracking.
2. **Expand Product Categories**: Include more brands and products for a broader sentiment analysis scope.
3. **Explore Deep Learning Models**: Test models like LSTM and BERT for potentially better performance.

Installation and Setup

1. Clone the repository: git clone https://github.com/George-Chira/twitter-sentiment-analysis
2. Install the required packages
3. The dataset is available in the data folder.
4. Run the Jupyter notebook for data analysis and modeling:

How to Contribute

Contributions are welcome! Follow these steps to contribute to the project:

1. Fork the repository: Click the "Fork" button at the top right corner of the repository page.
2. Clone your fork: git clone
3. Create a new branch for your feature
4. Make your changes in the codebase.