Ominak Update README.md 4234c59 · 3 minutes ago

Analysis-of-Twitter-sentimen...	Add github.pdf	13 minutes ago
Phase4Project-NLP-TwitterS...	Add NLP project notebook	4 days ago
Phase4Project_NLP_TwitterS...	Broke it down it different sections	5 hours ago
Phase4Project_NLP_TwitterS...	Add notebook as PDF	1 hour ago
README.md	Update README.md	3 minutes ago
crowdflower_twitter_sentime...	Add notebook	last week
github.pdf	Add github.pdf	13 minutes ago
none technical presentation ...	Non-Technical pdf presentation	1 hour ago

README

Project 4: Twitter Sentiment Analysis

Project Title: Twitter Sentiment Analysis — Crowdflower Dataset

1.0 Introduction

This project analyzes Twitter feedback on Apple and Google products using Natural Language Processing (NLP). The aim is to classify tweets as Positive, Neutral, or Negative automatically, providing insights for brand monitoring, customer support, and product development.

2.0 Research Problem & Objectives

Problem Statement:
Apple and Google products generate thousands of tweets daily, reflecting customer experiences, frustrations, and praise. Manually analyzing this volume is slow, inconsistent, and difficult to scale. The goal is to build an automated system for sentiment classification.

Main Objective:
Develop a robust multi-class NLP model that classifies tweet sentiments to support data-driven decisions.

Specific Objectives:

- Prepare and explore the dataset: Clean and preprocess tweets, analyze sentiment distribution.
- Build and compare models: Train Logistic Regression, SVM, Random Forest, and LSTM models.
- Apply TF-IDF vectorization and TruncatedSVD dimensionality reduction.
- Tune a Logistic Regression model using GridSearchCV.
- Interpret model results using evaluation metrics and visualizations.
- Recommend the best-performing model.

Stakeholders:

- Marketing Team: Monitor brand perception.
- Customer Support: Detect negative feedback and prioritize complaints.
- Product Team: Understand user reactions for feature development.
- Data Science Team: Build, validate, and interpret models.
- Business Executives: Use insights for strategic decisions.

Project Scope:

- In-Scope:* Using existing labeled tweets, model development, evaluation, interpretation, and saving models.
- Out-of-Scope: Live data collection, real-time deployment, other brands, or advanced NLP techniques beyond LSTM.

3.0 Data Understanding

3.1 Dataset Overview

The dataset originates from CrowdFlower and contains approximately 9,000 tweets labeled with sentiment (positive, negative or neutral. The data includes text and emotion categories manually annotated by human contributors.

3.2 Loading the Data

The dataset was loaded using 'pandas.read_csv' with appropriate encoding handling to accomodate special characters in tweets.

3.3 Variable Description

- tweet_text: Raw tweet content.
- emotion_in_tweet_is_directed_at: Brand/product mentioned (nullable).
- is_there_an_emotion_directed_at_a_brand_or_product: Sentiment label (Positive, Negative, Neutral, No emotion, I can’t tell).

4.0 Data Cleaning & Pre-processing

4.1 Handling Missing Values

Missing values were removed in non-essential columns to improve model integrity.

4.2 Feature Engineering

Two key feature transformations were applied:

1. TF-IDF Vectorization – Converted cleaned text into numerical format.
2. TruncatedSVD – Reduced dimensionality of TF-IDF vectors for efficiency and improved model generalization

4.3 Text Preprocessing (Tokenization, Lemmatization, Stopwords)

A new column, *clean_text*, was created through the following steps:

- Converting text to lowercase
- Removing punctuation and special characters
- Tokenizing text
- Removing stopwords using NLTK
- Applying stemming or lemmatizati

5.0 Exploratory Data Analysis (EDA)

5.1 Sentiment Distribution

Most tweets are neutral ("No emotion toward brand or product"). Positive sentiment is second, negative is limited, "I can't tell" minimal.

5.2 Text Insights

Common words across the dataset include apple, google, iphone, ipad, google maps, and sxsw, indicating frequent discussion topics.

5.3 Vizualization

Word clouds and sentiment distribution charts were produced to illustrate text patterns and class imbalance.

6.0 Modeling & Evaluation (Classical Machine Learning)

This section implements classical machine learning models for tweet sentiment classification. The goal is to predict whether a tweet expresses 'Positive, Negative, Neutral, or I can't tell sentiment.
Three models were trained using a pipeline that includes TF-IDF vectorization and dimensionality reduction (TruncatedSVD):

1. Logistic Regression
2. Linear SVM
3. Random Forest

6.1 Evaluation Metrics:

- Accuracy
- Precision, Recall, F1-Score
- Confusion Matrix

6.2 Key Findings:

Logistic Regression (~52% Accuracy):

- Performs well on the majority class ("No emotion toward brand or product")
- Poor performance on minority classes ("Negative" and "I can't tell")
- Confusion matrix shows misclassifications biased toward the majority

Linear SVM (~60% Accuracy):

- Slight improvement for Positive emotion detection
- Still weak for minority classes
- Majority class dominates predictions

Random Forest (~64% Accuracy):

- Best among classical models
- Better handles Positive and Negative emotions
- "I can't tell" class remains poorly predicted

7.0 Hyperparameter Tuning with GridSearchCV

A GridSearchCV approach was applied to Logistic Regression to optimize parameters:

- tfidf__max_features: [3000, 5000]
- tfidf__ngram_range: [(1,1), (1,2)]
- svd__n_components: [100, 300]
- model__C: [0.1, 1, 10]

7.1 Results

- Accuracy remained around *51%*
- Majority class still dominates
- Positive sentiment detected moderately
- Negative & I can't tell classes poorly predicted

8.0 Deep Learning Approach – LSTM Model

This section focuses on using a deep learning LSTM model for sentiment analysis of Apple and Google product tweets. The goal is to improve the prediction of minority sentiment classes that are often missed by traditional machine learning models.

8.1 Overview

- The LSTM (Long Short-Term Memory)* network is designed to capture sequential patterns in text, making it suitable for analyzing tweet sentiments.
- Preprocessing included tokenization and padding of text data to standardize input sequences.
- Sentiment labels were encoded and class weights applied to address data imbalance.

8.2 Key Steps

1. Model Architecture

- Embedding layer to convert words into dense vectors.
- LSTM layer to capture contextual dependencies in tweets.
- Dense layers with ReLU and softmax activation for multi-class classification.

2. Training
   - Model trained with early stopping to avoid overfitting.
   - Class weights ensured minority sentiment classes were appropriately considered.
3. Evaluation
   - Test accuracy achieved: ~59%.
   - Improved prediction for minority classes compared to traditional ML approaches.

8.3 Insights

- LSTM effectively handles sequential text data and improves sentiment classification, especially for underrepresented classes.
- Performance is limited by dataset size; larger datasets or pre-trained embeddings (e.g., BERT) could enhance accuracy.
- The model demonstrates the potential of deep learning for nuanced sentiment analysis beyond conventional techniques.

8.4 Summary

The LSTM-based approach complements classical ML models by better capturing tweet context and providing more balanced predictions across sentiment classes. It serves as a foundational step before advanced techniques like transformer-based models.

9.0 Model Interpretability – LIME

This section focuses on understanding why the sentiment model makes specific predictions* using LIME (Local Interpretable Model-agnostic Explanations).

9.1 Overview

- LIME provides insights into which words influence a model's sentiment prediction.
- The method is model-agnostic, meaning it works with any trained classifier, including Random Forest or Logistic Regression.
- Helps validate the model and build trust for business stakeholders.

9.2 Key Insights from LIME Analysis

- Example Tweet:
  "congrats rt mention yes gowalla wins best android app at the team android choice awards thanks all sx..."
- True Sentiment: Positive emotion
- Predicted Sentiment: Positive emotion (90% confidence)
- Words contributing to the positive prediction:
  "android", "team", "app", "congrats", "wins", "awards"

- LIME highlights contextual words that indicate sentiment, e.g., achievement-related language strengthens positive predictions.

9.3 Business Value

- Provides transparent explanations for model predictions.
- Helps marketing and product teams trust automated sentiment analysis.
- Enables detection of positive and negative brand mentions with clear reasoning.
- Supports actionable insights for monitoring brand perception and guiding strategic decisions.

9.4 Summary

LIME ensures that sentiment analysis models are interpretable and explainable, allowing stakeholders to understand model decisions and verify that predictions are based on meaningful textual features

10.0 Conclusion and Recommendations

10.1 Conclusion

- The project successfully implemented automated sentiment analysis on tweets related to Apple and Google products.
- Classical ML models (Logistic Regression, Linear SVM, Random Forest) achieved moderate accuracy, with Random Forest performing best (~64% accuracy).
- Deep learning (LSTM) provided comparable performance, demonstrating the dataset's class imbalance and limited size as challenges.
- Model interpretability with LIME confirmed that predictions are based on relevant textual features, ensuring explainable insights for stakeholders.

10.2 Key Observations

- Class imbalance heavily influenced minority class prediction (Negative emotion, I can’t tell).
- Most tweets were neutral, highlighting the need for careful metric selection and possibly resampling or weighted approaches.
- Positive sentiment is detected more reliably than negative, indicating potential improvement areas for critical feedback monitoring

10.3 Recommendations

1. Data Enhancement
    - Increase the size of labeled negative and ambiguous tweets to improve model performance on minority classes.
    - Include tweets from recent product releases and events to maintain relevance.
2. Model Improvement
    - Explore ensemble methods combining classical ML and deep learning for better accuracy.
    - Experiment with pre-trained language models (e.g., BERT, RoBERTa) for improved contextual understanding.
3. Business Application
    - Use model predictions to monitor brand perception in real-time campaigns.

- Prioritize negative sentiment detection to address customer complaints quickly.
- Track positive sentiment trends to identify successful marketing messages and product features.

4. Interpretability

- Continue using tools like LIME or SHAP to *explain individual predictions*, ensuring transparency and trust for stakeholders.

11.0 References

- CrowdFlower Twitter Sentiment Dataset, data.world
- NLTK Documentation
- scikit-learn Documentation
- TensorFlow/Keras Documentation
- LIME: Local Interpretable Model-Agnostic Explanations

Group Members

- Catherine Gachiri
- Tanveer Mbitiru
- Kelvin Omina
- Cindy Achieng
- James Ouma

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Contributors 5

Languages

Jupyter Notebook 100.0%