NLP-BASED SENTIMENT DISSECTION OF APPLE VISION PRO RECEPTION ON MARQUES BROWNLEE'S YOUTUBE CHANNEL.

Contents

Resources

Github repository:

Members

Esther Nyaikuba esther.nyaikuba@student.moringaschool.com
Ashley Simiyu ashley.simiyu@student.moringaschool.com
Evalyne Macharia evalyne.macharia@student.moringaschool.com
Lorrah Ngine lorrah.ngine@student.moringaschool.com
Silah Rugut silah.rugut@student.moringaschool.com
Sydney Mumbo sydney.mumbo@student.moringa.com

Business Understanding

Business Overview

Apple's launch of the Apple Vision Pro, a cutting-edge wearable augmented reality device, has garnered attention from consumers and tech enthusiasts alike. However, understanding the reception of this product among Marques Brownlee's audience, a tech influencer with a vast reach on YouTube, remains crucial for strategic decision-making.

The challenge lies in effectively leveraging Natural Language Processing (NLP) techniques to sift through the vast volume of viewer comments on Marques Brownlee's YouTube channel. The goal is to extract valuable insights regarding consumer sentiment towards the Apple Vision Pro, discerning whether viewers express positive, negative, or neutral opinions. Additionally, identifying indicators of purchase intent or dissatisfaction with specific features is paramount.

The sentiment analysis challenge is heightened by the variety of viewer comments, spanning concise reviews, in-depth critiques, and personal anecdotes. The analysis must be discerning to grasp subtle linguistic elements and contextual cues, enabling precise interpretation of viewer sentiments.

Business Objectives

The **main objective** of this study is to evaluate the perception of the Apple Vision Pro among Marques Brownlee's YouTube audience using Natural Language Processing (NLP) to discern sentiments and gauge purchase intent.

Other objectives are to:

1. Analyze viewer comments to identify positive and negative sentiments towards the Apple Vision Pro.
2. Determine the level of interest and inclination towards purchasing the product expressed by viewers.
3. Provide actionable insights to the marketing team, product development team, and strategic decision-makers within Apple for informed decision-making and strategy formulation.

Success Criteria

This project will be deemed successful when the following is achieved:

1. **Timely Completion**: Successfully complete the project within the designated timeframe, utilizing available resources efficiently.
2. **Functional Requirements Achievement:**
    - Data Science Skills: Demonstrate proficiency in essential data science tasks, including data cleaning and exploratory data analysis (EDA), ensuring the quality and relevance of the dataset.
    - Machine Learning Model Development: Develop robust Machine Learning models utilizing Natural Language Processing (NLP) techniques and sentiment analysis algorithms to accurately analyze viewer comments and sentiment towards the Apple Vision Pro.
    - Model Deployment: Deploy the sentiment analysis model effectively to enable seamless interaction with end-users, ensuring accessibility and ease of use.
3. **Non-functional Requirements Fulfillment:**
    - Customer Satisfaction: Ensure that the sentiment analysis tool contributes positively to customer satisfaction by providing valuable insights into viewer sentiments and preferences.
    - Teamwork: Foster effective collaboration and teamwork among project

stakeholders, promoting communication, mutual support, and shared accountability.

4. **Objective Fulfillment**: Meet all specified project objectives, including:
    - Analyzing viewer comments on Marques Brownlee's YouTube videos to gauge sentiment towards the Apple Vision Pro.
    - Identifying trends and patterns in viewer feedback to inform marketing strategies and product improvements.
    - Assessing purchase intent expressed by viewers to guide sales and marketing initiatives.
    - Providing actionable insights to the marketing team, product development team, and strategic decision-makers at Apple.

Data Understanding Overview

The Marques Brownlee project involves sentiment analysis on comments from his YouTube channel. The goal is to understand the sentiment of viewers' comments towards his videos. By analyzing these sentiments, we aim to gain insights into the audience's reactions and opinions.

Data Description

The dataset used in this analysis contains comments extracted from YouTube videos related to the Apple Vision Pro product. The dataset is stored in a pandas DataFrame named comments_df, and it consists of 59,251 rows (comments) and 4 columns:

1. **comment_text** (object): This column contains the raw text of each comment. It represents the unprocessed comment data as it was scraped from YouTube.
2. **author** (object): This column stores the display name or username of the author who posted each comment.
3. **likes** (int64): This column contains the number of likes received for each comment.
4. **timestamp** (object): This column represents the timestamp or date when each comment was posted.

Verifying Data Quality

To ensure data quality, we conducted several checks:

- **Checked data types:** All columns were appropriately typed.
- **Addressed missing values:** No missing values were found in the dataset.
- **Handled duplicates:** Duplicate comments were identified
- **Language verification:** Comments were checked for language consistency to ensure accurate sentiment analysis.

Data Preparation

Data preparation involves several steps:

1. Scraping Comments: Comments were scraped from Marques Brownlee's YouTube videos using the YouTube Data API.
2. Cleaning and Preprocessing: Comments underwent cleaning to remove HTML tags, special characters, and non-English text. They were then tokenized, lemmatized, and stopwords were removed to prepare them for analysis.
3. Feature Engineering: Relevant features such as comment length were derived to provide additional insights into the data.
4. Splitting the Data: The dataset was split into training, validation, and test sets for model training and evaluation.

Preprocessing

Text preprocessing is a crucial step in natural language processing tasks, as it helps to clean, normalize, and transform the raw text data into a format suitable for analysis and modeling. In this analysis, we have applied several preprocessing techniques to the comments dataset to prepare it for sentiment analysis.

1. **Text Cleaning**:
    - We defined a custom function clean_text to remove HTML tags and special characters from the text data. This step is essential because HTML tags and special characters can introduce noise and irrelevant information that may negatively impact the analysis.
    - The function converts the cleaned text to lowercase to ensure consistency and uniformity throughout the dataset.
2. **Handling Non-English Comments**:
    - Since the analysis focuses on English comments, we need to identify and potentially remove non-English comments from the dataset.

- To achieve this, we installed the langdetect library, which can detect the language of a given text.
- We then filtered out comments that are not in English based on the language detection results.

3. **Removing Duplicates and Empty Rows**:
    - Duplicate comments can introduce bias and redundancy in the analysis, so we identified however we did not drop them since they were from different authors.
    - Additionally, we checked for and removed any rows where the cleaned_text column was empty or contained only whitespace characters, as these rows do not contribute any valuable information to the analysis.
4. **Tokenization and Stop Word Removal**:
    - We used the word_tokenize function from the nltk.tokenize module to tokenize the text data.
    - After tokenization, we removed stop words (common words like "the," "and," "is," .) from the tokenized text, as they do not carry much meaning and can introduce noise in the analysis.
    - The stop words were obtained from the nltk.corpus module, and we used a list comprehension to filter out stop words from the tokenized text.
5. **Lemmatization**:
    - We utilized the WordNetLemmatizer from the nltk.stem module to perform lemmatization on the tokenized text.
    - Additionally, we defined a helper function get_wordnet_pos to determine the part-of-speech tag for each token, as lemmatization requires this information.
    - The lemmatized text was stored in a new column lemmatized_text for further analysis.

Modeling

We employed TextBlob, VADER, Support Vector Machines (SVM), and Long Short-Term Memory (LSTM) neural networks for sentiment analysis. TextBlob and VADER served as lexicon-based models, while SVM and LSTM represented machine learning and deep learning approaches, respectively. Each model was trained and evaluated using appropriate metrics.

- **TextBlob**:

- TextBlob was employed as a baseline model to obtain initial sentiment polarity scores for the preprocessed comments.
- The model classified comments into positive, negative, or neutral categories based on the polarity scores.
- The sentiment distribution predicted by TextBlob was visualized using a pie chart, providing an overview of the model's performance.

- **VADER (Valence Aware Dictionary and sEntiment Reasoner)**:

- The NLTK (Natural Language Toolkit) library was used to implement the VADER model.
- The preprocessed and lemmatized text data were first joined back into a single string for each comment. The SentimentIntensityAnalyzer from the VADER library was then used to calculate sentiment scores for each comment, including positive, negative, neutral, and compound scores.
- The sentiment was categorized based on the compound score, where a score greater than or equal to 0.05 was considered 'Positive', a score less than or equal to -0.05 was considered 'Negative', and scores in between were labeled as 'Neutral'.

- The sentiment distribution predicted by VADER was also visualized using a pie chart for comparison with TextBlob.

- **Support Vector Machines (SVM)**:

- In addition to the lexicon-based models, a machine learning approach using Support Vector Machines (SVM) was employed for sentiment analysis. The text data were first converted into a numerical feature representation using the Term Frequency-Inverse Document Frequency (TF-IDF) vectorizer.
- The TF-IDF vectors were split into training and testing sets, and an SVM classifier with a linear kernel was trained on the training data. The trained model was then used to predict sentiment labels for the testing data.
- The performance of the SVM model was evaluated using various metrics, including accuracy, precision, recall, F1-score, and a confusion matrix. The model achieved an accuracy of 0.92 and a weighted average F1-score of 0.92 on the testing data.

- **LSTM (Long Short-Term Memory)**:

- The sentiment analysis task was approached via a deep learning methodology utilizing LSTM (Long Short-Term Memory) neural networks, specialized recurrent neural network (RNN) architectures adept at capturing long-term dependencies in sequential data, particularly suited for text analysis.
- The LSTM model comprised three main layers: an Embedding Layer responsible for converting input text data into fixed-size dense vectors essential for neural network

processing, with an output dimension of 128, representing each word in the vocabulary by a 128-dimensional vector; an LSTM Layer used to learn temporal dependencies within the text data, comprising 128 LSTM units to capture complex patterns and relationships within text sequences, with dropout and recurrent dropout regularization techniques applied at a dropout rate of 0.2 to prevent overfitting; and an Output Dense Layer featuring three units corresponding to the three sentiment classes (positive, neutral, and negative), employing a softmax activation function to facilitate multi-class classification.

- Model training ensued utilizing the Adam optimizer alongside a sparse categorical cross-entropy loss function, conducive to multi-class classification tasks, iterating over the training data for 10 epochs with a batch size of 32 to optimize convergence and computational efficiency.

Evaluation

TextBlob Model Evaluation

The primary evaluation method used for the TextBlob model was a pie chart visualization. The pie chart displayed the distribution of sentiment labels ('Positive', 'Negative', and 'Neutral') predicted by the TextBlob model on the cleaned text data.

While the pie chart provides a visual representation of the sentiment distribution, it does not give quantitative metrics or scores to evaluate the model's performance objectively. The lack of numerical evaluation metrics makes it challenging to assess the TextBlob model's accuracy, precision, recall, or F1-score.

VADER Model Evaluation

The confusion matrix and the calculated TP, FP, FN, and TN values provide valuable insights into the VADER model's performance:

- True Positives (TP): 3491
- False Positives (FP): 250
- False Negatives (FN): 157
- True Negatives (TN): 7889

From these values, we can observe that the VADER model performs relatively well in identifying neutral and positive sentiments, with a high number of true positives for the neutral class (3491) and a relatively low number of false negatives (157) for the positive class. However, the model struggles more with identifying negative sentiments, as indicated by the higher number of false positives (250) for the negative class.

SVM Model Evaluation

The SVM model achieved an impressive overall accuracy of 0.92, indicating that it correctly classified 92% of the instances in the testing data.

The precision, recall, and F1-scores for each class are:

- Negative:
    - Precision: 0.89
    - Recall: 0.78
    - F1-score: 0.83
- Neutral:
    - Precision: 0.93
    - Recall: 0.96
    - F1-score: 0.94
- Positive:
    - Precision: 0.92
    - Recall: 0.95
    - F1-score: 0.93

These metrics suggest that the SVM model performs well across all classes, with particularly high precision and recall for the neutral and positive classes. The lower recall for the negative class (0.78) indicates that the model may struggle more in identifying negative sentiments correctly.

The confusion matrix further reinforces these observations, showing a higher number of false negatives (387) for the negative class compared to the other classes.

LSTM Model Evaluation

The trained LSTM model was evaluated on a separate test dataset to assess its performance. The following performance metrics were computed:

1. Test Loss:
    - The test loss, calculated using log loss, measures the model's accuracy in predicting the class probabilities for each instance in the test dataset.
    - The obtained test loss was approximately 0.436.
2. Test Accuracy:
    - Test accuracy quantifies the proportion of correctly classified instances in the test dataset.
    - The LSTM model achieved an impressive test accuracy of approximately 92.37%.

Deployment

We deployed a sentiment analysis application designed to interpret and analyze YouTube comments using our csv file that has been preprocessed comments.

The deployment was achieved using Streamlit, an open-source app framework specifically suited for machine learning and data science projects. Streamlit allows for the rapid creation of interactive, web-based applications that can present data insights and accept user inputs without extensive web development experience.

Deployment Process

1. **Development Environment:**
    - The sentiment analysis model was developed using Python, utilizing libraries such as Pandas for data manipulation, NLTK for natural language processing, and Sklearn for model building and evaluation.
2. **Streamlit Integration:**
    - The model was integrated into a Streamlit application, which acts as the interface for user interaction. The integration involved scripting in Python, with Streamlit commands to control the layout and interactivity of the application.
3. **Application Features:**
    - **Interactive Dashboard:** Users can view overall sentiment analysis results through texts or uploading their csv that contained mined comments.
    - **Visualization:** We also included word cloud to visual positive, negative, and neutral sentiments.
    - **Comment Search:** Users can search for specific comments or filter comments based on sentiment categories (positive, negative, neutral).

Here is our link https://s3ntiment-analyzer.streamlit.app/

Conclusion

This report demonstrates the successful application of NLP techniques for sentiment analysis on viewer comments regarding the Apple Vision Pro. The insights derived from this analysis will inform strategic decision-making processes and contribute to enhancing the product's reception among Marques Brownlee's audience.