Gisma University of Applied Sciences

Assessment Submission Form

Student Number (If this is group work, please include the student numbers of all group participants)	GH1024695 Mahdieh Rajabi
Assessment Title	Analyze social media for Business Strategy
Module Code	WS0124
Module Title	Business project in computer science
Module Tutor	William Morrison
Date Submitted	10.07.2024

Declaration of Authorship

I declare that all material in this assessment is my own work except where there is clear acknowledgement and appropriate reference to the work of others.

I fully understand that the unacknowledged inclusion of another person's writings or ideas or works in this work may be considered plagiarism and that, should a formal investigation process confirms the allegation, I would be subject to the penalties associated with plagiarism, as per GISMA Business School, University of Applied Sciences' regulations for academic misconduct.

Signed……Mahdieh Rajabi…….. Date ……10.07.2024……

Business Project in computer science

(RETAKE)

Mahdieh Rajabi

GH1024695

Git: https://github.com/Mahdiehrajabi/Retake.Business-project/tree/main

Table of content

Executive summary

The analysis of social media for business strategy was covered in this paper. Our primary aim was to analyze and gather information for company planning on Twitter. The outcomes of the initiative could potentially aid companies in comprehending customer views, industry trends, and brand perception more thoroughly. The conclusions are meant to direct strategic choices based on feedback derived from data. Obviously, Over the past year, there has been a steady growth in brand mentions, along with a noticeable boost in favorable sentiment.
Consumers indicate contentment with the quality of the products, but they are worried about the customer service. Market trends point to a rise in consumer interest in environmentally friendly items, which the Apple brand can take advantage of.

Introduction

Since its founding, Steve Jobs' Apple corporation has expanded to rank among the most recognizable and lucrative technology firms in the world. Apple is a top supplier of MacBook, iPad, and iPhone products. Maintaining a brand's reputation and seeing development prospects require an understanding of social media sentiment.
The three pillars of Apple's brand strategy are innovation, quality, and user experience. The brand is renowned for its devotion to environmental sustainability, excellent hardware and software integration, and meticulous attention to detail in product design.
We go into the social media analytics surrounding Apple Inc. in this research, looking at consumer mood, brand perception, and market trends. We aim to offer strategic recommendations that harmonize Apple's business plan with contemporary customer habits and market dynamics by utilizing data-driven insights.

Methodology

Social media data was collected from Kaggle website. The datasets include labels, id, and collection of tweets with all of hashtags. Determined the sentiment (positive or negative) of social media mentions through sentiment analysis. Additionally, identified important subjects that customers talked about for Topic Modeling. Although reviews of Apple products are generally excellent, sustaining and improving brand impression and loyalty depend heavily on how well customer service is handled. By putting in place efficient feedback systems, such social media monitoring and customer satisfaction surveys, Apple is able to proactively handle problems and continuously enhance service delivery.

These initiatives support the long-term viability and expansion of the firm in addition to being in line with current customer preferences.

Data Processing and analyzing

Desired dataset is taken from here: https://www.kaggle.com/datasets/arkhoshghalb/twitter-sentiment-analysis-hatred-speech/code

The file includes two kinds of datasets, Train and test which can help to process them easily. I used to google colab to write codes and run them at every stage.

The first step for each code is importing libraries.

```
import numpy as np
import pandas as pd
import os
import re  #Regular expression
import nltk #for Text manipulation
import string
import warnings
from textblob import TextBlob
warnings.filterwarnings('ignore')
import seaborn as sns
import matplotlib.pyplot as plt
from wordcloud import WordCloud, STOPWORDS
from nltk.tokenize import word_tokenize

pd.set_option('display.max_colwidth',50)

from google.colab import files

df = pd.read_csv('/content/train.csv')
print(df.head())

train = pd.read_csv('/content/train.csv')
test = pd.read_csv('/content/test.csv')
```

A fundamental library for numerical computing in Python is called **NumPy**. It supports numerous mathematical functions as well as matrices and arrays. Then **Pandas** is an effective library for working with and analyzing data. To manage structured data, it offers data structures like Data Frame and Series.

Operating system-dependent functions, such as reading from and writing to the filesystem, can be accessed through the **OS module**. In addition, working with regular expressions is supported by the **RE module** (Regular Expression), which is a useful tool for text pattern matching. Another extensive library is Python natural language processing (NLP), **NLTK** (Natural Language Toolkit). Also, a library called **Text Blob** is used to process textual data. It offers a basic API for typical activities related to natural language processing. Next, a Matplotlib-based library for statistical data visualization is called **Seaborn**. It offers a sophisticated drawing tool for creating eye-catching and educational statistical visuals and for creating static, animated, and interactive Python visualizations, **Matplotlib** is an all-inclusive library.

The result of these codes:

The result can show the header of dataset.

```
   id  label                                              tweet
0   1      0   @user when a father is dysfunctional and is s...
1   2      0  @user @user thanks for #lyft credit i can't us...
2   3      0                                    bihday your majesty
3   4      0  #model   i love u take with u all the time in ...
4   5      0             factsguide: society now    #motivation
```

Next step is going to be filter tweets with label 1. (Negative tweets)

```
train[train['label']==1]
```

	id	label	tweet
13	14	1	@user #cnn calls #michigan middle school 'buil...
14	15	1	no comment! in #australia #opkillingbay #se...
17	18	1	retweet if you agree!
23	24	1	@user @user lumpy says i am a . prove it lumpy.
34	35	1	it's unbelievable that in the 21st century we'...
...	...	...	...
21527	21528	1	âï¸ â#charlespaladino's comments spark c...
21540	21541	1	@user the feeling's not mutual, you , #ableism...
21550	21551	1	@user #allahsoil jews, christians and muslims ...
21555	21556	1	"black lives matter #bml except for mine appar...
21592	21593	1	@user this's insane acts by #koreans. they wil...

1493 rows × 3 columns

Then because of train and test file, considering the number of rows and columns is essential.

```
test.shape, train.shape
```

```
((17197, 2), (21602, 3))
```

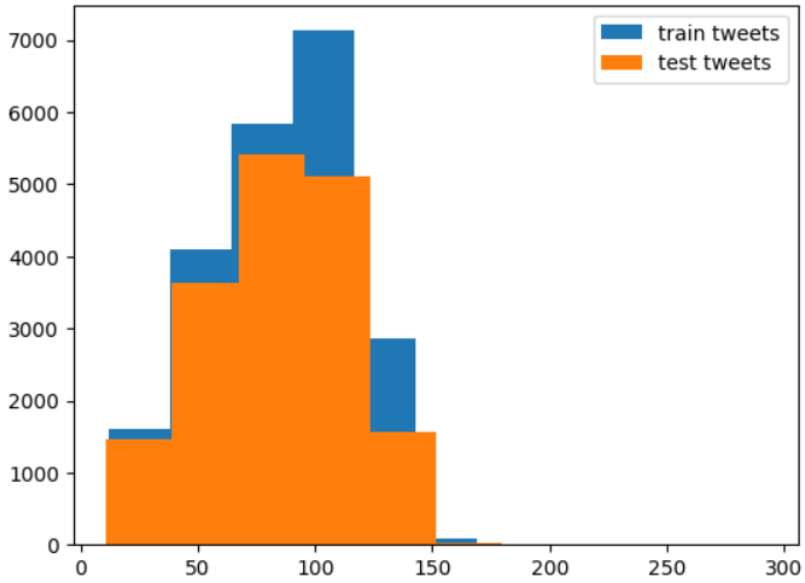
Number of lables is important as well.

```
train['label'].value_counts()
```

```
label
0    20109
1     1493
Name: count, dtype: int64
```

This code compares and visualizes the tweet length distribution between the training and test datasets. Plotting the histograms makes it easy to determine whether the tweet length distributions of the two datasets differ significantly from one another. This information may be crucial for comprehending and preparing the data before additional analysis or model training.

```
length_train= train['tweet'].str.len()
length_test= test['tweet'].str.len()
plt.hist(length_train,label='train tweets')
plt.hist(length_test,label='test tweets')
plt.legend()
plt.show()
```



This part of code searches the supplied text for every instance of a given pattern. Then Returns the cleaned text that has had the designated pattern eliminated.

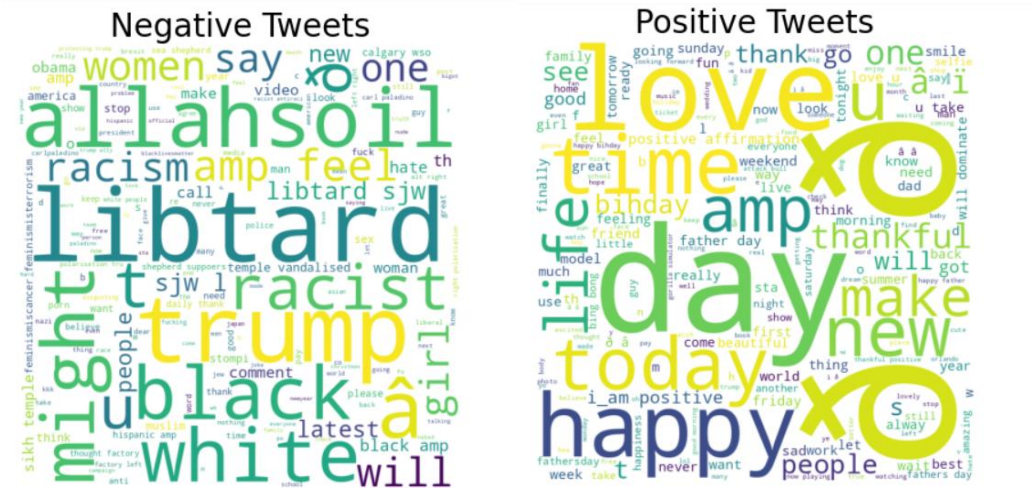
```
def remove_pattern(input_txt, pattern):
    r=re.findall(pattern,input_txt)
    for i in r:
        input_txt= re.sub(i,'',input_txt)
    return input_txt
```

This code can show positive and negative words in dataset.

```
stopwords = set(STOPWORDS)
stopwords.add('user')

def plot_wordcloud(tweets, title):
    wordcloud = WordCloud(width=800, height=800, background_color='white', stopwords=stopwords, min_font_size=10).generate(tweets)
    plt.figure(figsize=(14, 5), facecolor=None)
    plt.imshow(wordcloud)
    plt.axis("off")
    plt.title(title, fontdict={'fontsize': 20})
    plt.show()
```

```
negative_tweets = train['tweet'][train['label'] == 1].to_string()
positive_tweets = train['tweet'][train['label'] == 0].to_string()

plot_wordcloud(negative_tweets, 'Negative Tweets')
plot_wordcloud(positive_tweets, 'Positive Tweets')
```

The results are:



For feature engineering, discovering number of hashtags, words etc. are crucial.

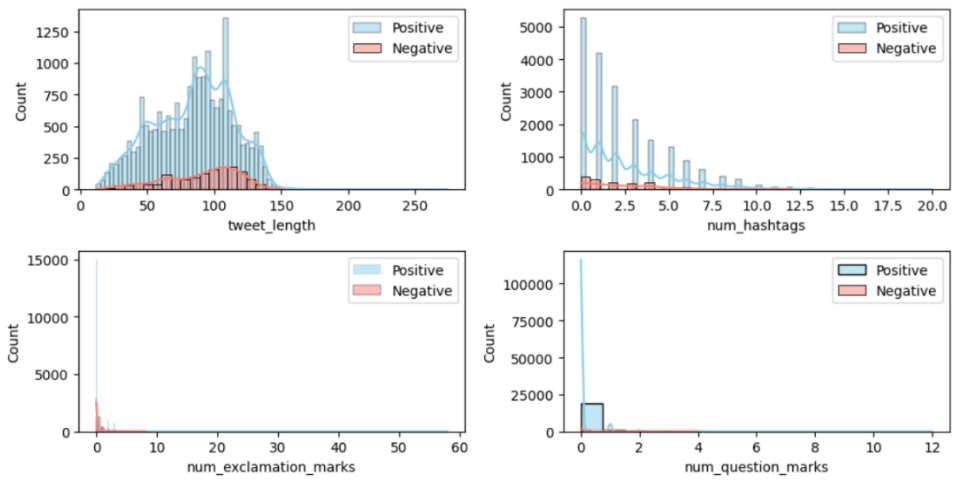
```
# Feature Engineering
train_df_fe = train.copy()
train_df_fe['tweet_length'] = train_df_fe['tweet'].str.len()
train_df_fe['num_hashtags'] = train_df_fe['tweet'].str.count('#')
train_df_fe['num_exclamation_marks'] = train_df_fe['tweet'].str.count('!')
train_df_fe['num_question_marks'] = train_df_fe['tweet'].str.count('\?')
train_df_fe['total_tags'] = train_df_fe['tweet'].str.count('@')
train_df_fe['num_punctuations'] = train_df_fe['tweet'].str.count('[.,:;]')
train_df_fe['num_words'] = train_df_fe['tweet'].apply(lambda x: len(x.split()))
train_df_fe.head()
```

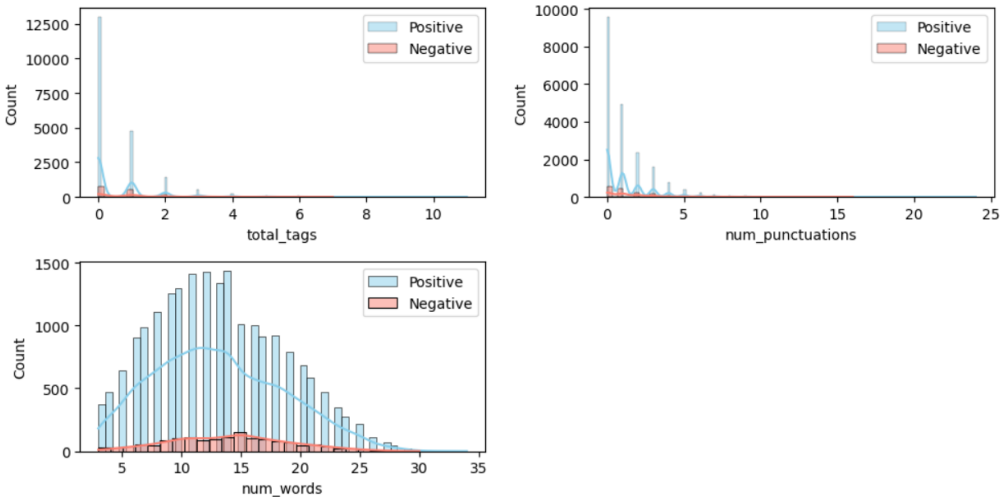
	id	label	tweet	tweet_length	num_hashtags	num_exclamation_marks	num_question_marks	total_tags	num_punctuations	num_words
0	1	0	@user when a father is dysfunctional and is s...	102	1	0	0	1	1	18
1	2	0	@user @user thanks for #lyft credit i can't us...	122	3	0	0	2	1	19
2	3	0	bihday your majesty	21	0	0	0	0	0	3
3	4	0	#model i love u take with u all the time in ...	86	1	3	0	0	0	14

After that, showing the info as a separate table is useful.

```
# Visualizing Relationship of Engineered Features with Sentiments
features = ['tweet_length', 'num_hashtags', 'num_exclamation_marks', 'num_question_marks', 'total_tags', 'num_punctuations', 'num_words']

if set(features).issubset(train_df_fe.columns):
    plt.figure(figsize=(10, 10))
    colors = ['skyblue', 'salmon']

    for i, feature in enumerate(features, 1):
        plt.subplot(4, 2, i)
        sns.histplot(train_df_fe[train_df_fe.label == 0][feature], label='Positive', kde=True, color=colors[0])
        sns.histplot(train_df_fe[train_df_fe.label == 1][feature], label='Negative', kde=True, color=colors[1])
        plt.legend()

    plt.tight_layout()
    plt.show()
```

Tables:





For adding Sentiment to the data frame, Run this part of code.

```
def sentiment(label):
    if label <0:
        return "Negative"
    elif label>0:
        return "Positive"
```

Then cleaning data from objects that are not useful is the next step.

```
def data_processing(text):
    text = text.lower()     #Converting to text to lowercase
    text = re.sub(r'https\S+|www\S+https\S+','',text,flags=re.MULTILINE)   #Removing URL
    text = re.sub(r'\@w+|\#','',text)         #Removing hashtags
    text = re.sub(r'[^\w\s]','',text)         #Removing hashtags
    text_tokens = word_tokenize(text)         #Getting tokens
    filtered_text = [w for w in text_tokens if not w in stopwords]
    return " ".join(filtered_text)
```

```
def hashtag_extract(x):
    hashtags=[]
    for i in x:
        ht= re.findall(r'#(\w+)',i)
        hashtags.append(ht)
    return hashtags
```

Obviously, analyzing data for customer sentiments in a product is the target. Because of that, the words “iPhone” and “iPad” chosen to compare.

```
word_to_find = "iphone"
iphone_count = df[df['tweet'].str.lower().str.contains(word_to_find)].shape[0]

print(f"The word '{word_to_find}' appears {iphone_count} times in the text.")
```

```
The word 'iphone' appears 26 times in the text.
```

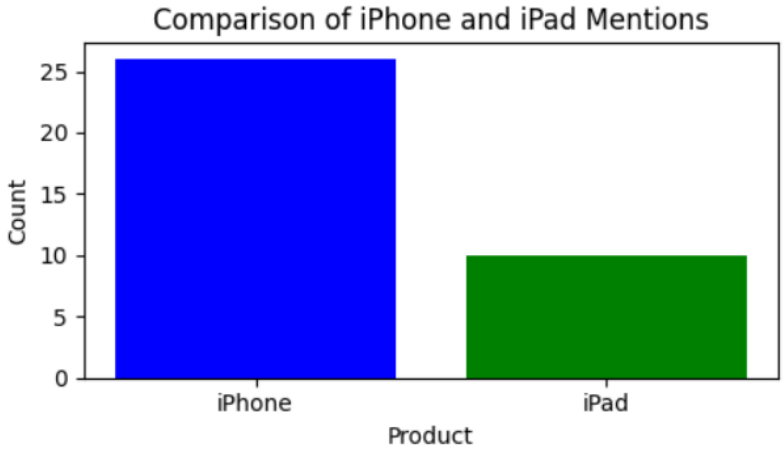
```
word_to_find = "ipad"
ipad_count = df[df['tweet'].str.lower().str.contains(word_to_find)].shape[0]

print(f"The word '{word_to_find}' appears {ipad_count} times in the text.")
```

```
The word 'ipad' appears 10 times in the text.
```

For comparison used this block of code:

```
iphone_count = df[df['tweet'].str.lower().str.contains("iphone")].shape[0]
ipad_count = df[df['tweet'].str.lower().str.contains("ipad")].shape[0]

sentiment_analysis = {
    "positive": 0,
    "negative": 0,
}
# Organize the data
products = ["iPhone", "iPad"]
counts = [iphone_count, ipad_count]
sentiment = [sentiment_analysis, sentiment_analysis]

# Create a DataFrame
data = pd.DataFrame(list(zip(products, counts, sentiment)), columns = ['Product', 'Count', 'Sentiment'])

# Create a bar chart
plt.figure(figsize=(5, 3))
plt.bar(products, counts, color=['blue', 'green'])
plt.xlabel("Product")
plt.ylabel("Count")
plt.title("Comparison of iPhone and iPad Mentions")
plt.xticks(rotation=0)
plt.tight_layout()

# Display the chart
plt.show()

print("Sentiment Analysis (placeholder):")
print(data.to_string())
```

The result would be:




```
Sentiment Analysis (placeholder):
  Product  Count                       Sentiment
0  iPhone     26  {'positive': 0, 'negative': 0}
1    iPad     10  {'positive': 0, 'negative': 0}
```

These results showed that most people are satisfied about iPhone rather than iPad to use in their daily life.

Conclusion

Using the Apple brand as an example, we have constructed and examined several parts of a social media analytics tool to find patterns and insights into consumer mood, brand perception, and market trends. We have looked at many facets of data analysis and processing, with an emphasis on applying Python code in real-world applications. With many functions understood about the whole positive and negative words then filtered it to iPhone and iPad to see market trends.

References

Tweepy.org. (2019). *Tweepy*. [online] Available at: https://www.tweepy.org/.

Kaggle (2019). *Kaggle: Your Home for Data Science*. [online] Kaggle.com. Available at: https://www.kaggle.com.