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	11.04.2024

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 ……………11.04.2024……………………

Business Project in computer science

Mahdieh Rajabi

GH1024695

Link: https://github.com/Mahdiehrajabi/Business-project.git

Introduction

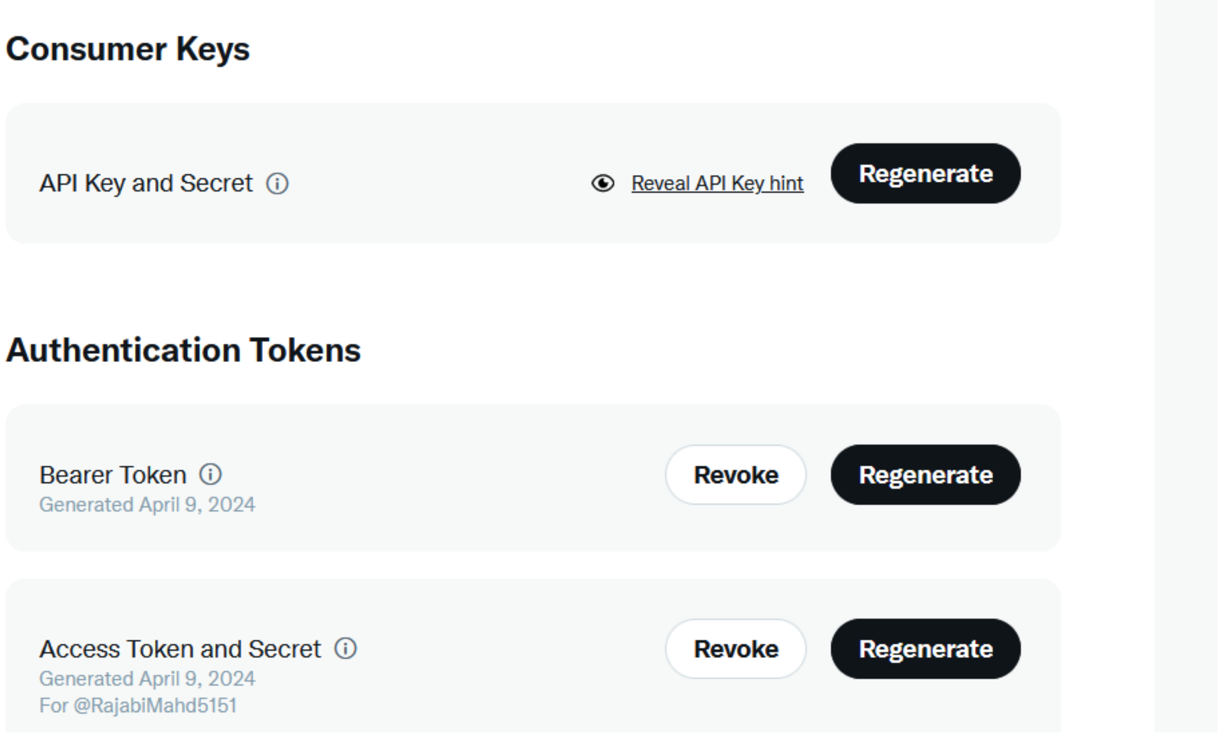
In this report, mentioned how we can Analyze social media for Business Strategy**.** We focused on twitter to understand and obtain knowledge for business strategy. The project's output may help firms better understand consumer attitudes, industry trends, and brand perception. The codes are written with python and twitter API. There are four steps such as data collection, data preprocessing, sentiment analysis and visualization.

For accessing the twitter data, need to use twitter API which enables programmatic interaction between developers and Twitter's platform.

For the first step, apply for a twitter developer account and sign up for developer account. The second step is creating an app and project. Then Generate API Keys and Tokens is third step. In order to authenticate queries to the Twitter API, these are necessary.

Methodology

Found these API keys and tokens here:



Consumer Keys

API Key and Secret

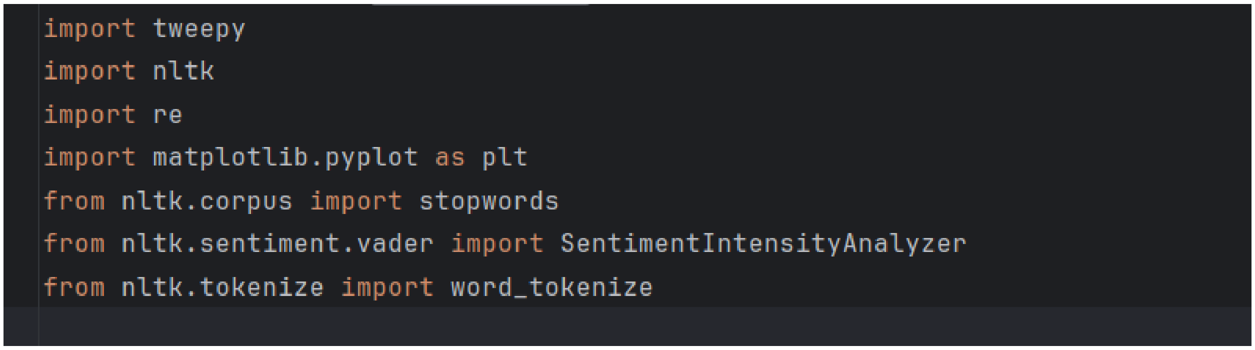
Authentication Tokens

Bearer Token
Generated April 9, 2024

Access Token and Secret
Generated April 9, 2024
For @RajabiMahd5151

Knowing these are significant for using in related code. API providers may efficiently authenticate, grant access, and oversee the use of their services by guaranteeing safe and regulated access to APIs.

At the beginning of the code, we should install and import various libraries, which is necessary.



```
import tweepy
import nltk
import re
import matplotlib.pyplot as plt
from nltk.corpus import stopwords
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from nltk.tokenize import word_tokenize
```

At first, mentioned “import tweepy”, A Python package that makes it easier to communicate with the Twitter API. Upon importing Tweepy into the script, users can utilize its classes and functions to gain access to a range of Twitter API functionalities, including the ability to retrieve, post, search for, and more.

“import nltk” is a statement that imports Natural Language Toolkit (NLTK) library. A variety of NLP tasks can be carried out using its different modules and capabilities. Regarding handling text data, NLTK offers an extensive collection of tools and resources.

“Import re” stands for “regular expressions” that is useful for searching texts.

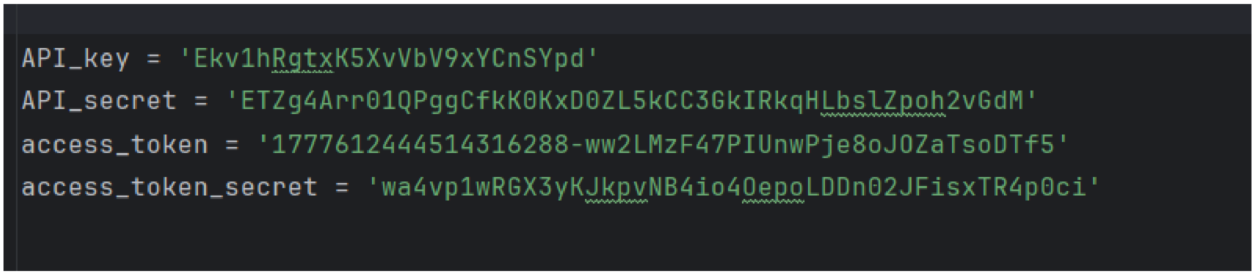
“From nltk. corpus import stopwords” provides a collection of stopwords that during text preprocessing, were filtered out.

“Import matplotlib. pyplot as plt” used for creating static, line plots etc.

The VADER (Valence Aware Dictionary and sEntiment Reasoner) module within NLTK is intended especially for sentiment analysis.

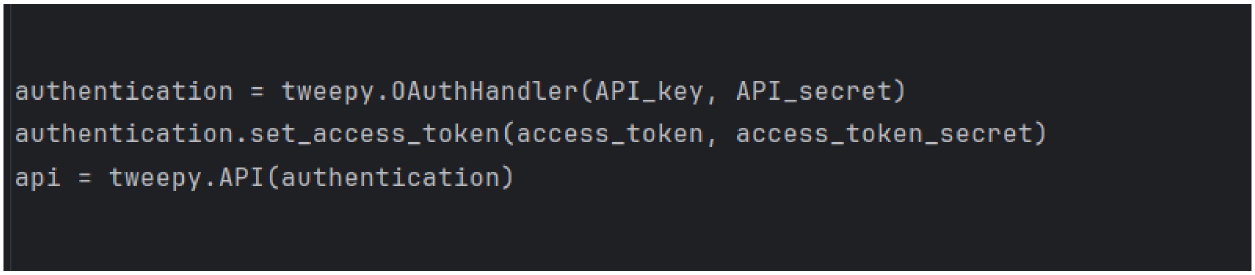
“From nltk. Tokenize import word_tokenize” For many NLP applications, such as text analysis, sentiment analysis, machine translation, and information retrieval, tokenization is a necessary preprocessing step.

The next step would be to use API keys and tokens for authentication.



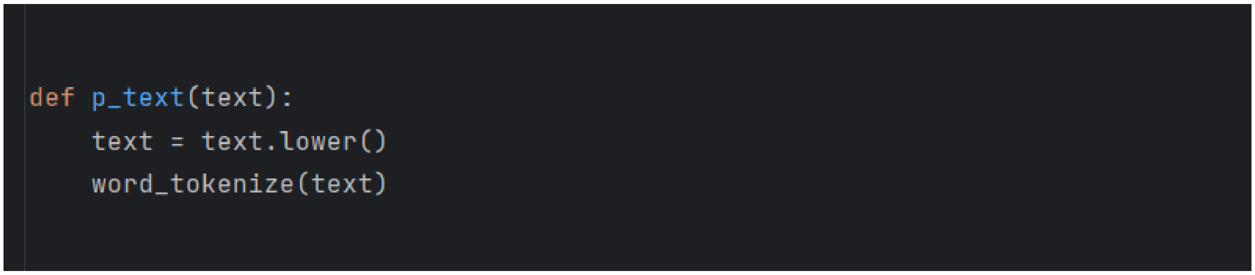
```
API_key = 'Ekv1hRgtxK5XvVbV9xYCnSYpd'
API_secret = 'ETZg4Arr01QPggCfkK0KxD0ZL5kCC3GkIRkqHLbslZpoh2vGdM'
access_token = '1777612444514316288-ww2LMzF47PIUnwPje8oJ0ZaTsoDTf5'
access_token_secret = 'wa4vp1wRGX3yKJkpvNB4io40epoLDDn02JFisxTR4p0ci'
```

Then



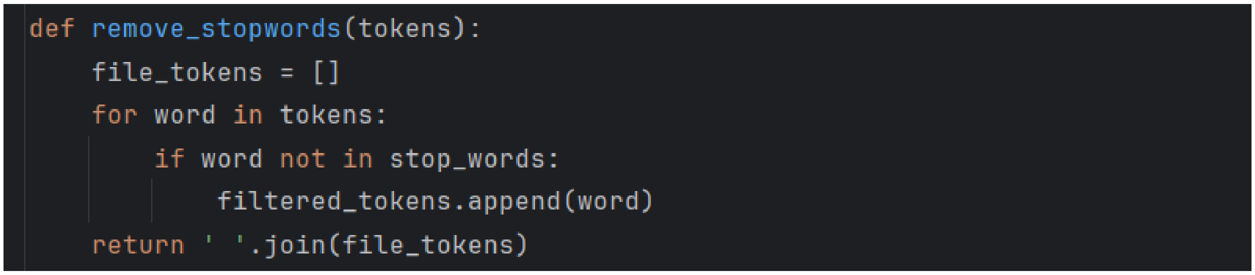
```
authentication = tweepy.OAuthHandler(API_key, API_secret)
authentication.set_access_token(access_token, access_token_secret)
api = tweepy.API(authentication)
```

After that adding this code is vital for convert text to lower case and tokenize text to word.



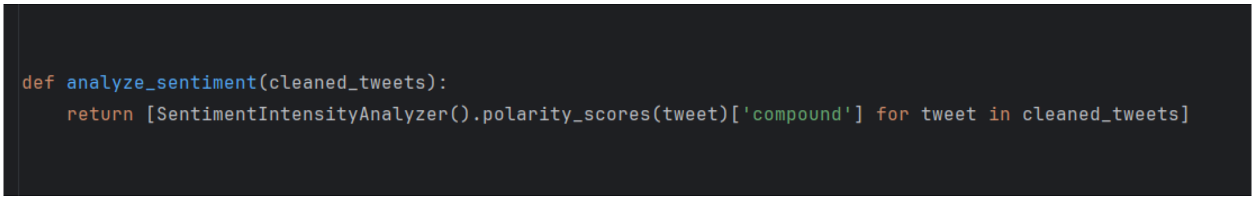
```
def p_text(text):
    text = text.lower()
    word_tokenize(text)
```

Then with this part of code the function will return the remaining tokens as a string after appropriately removing stopwords from the provided list of tokens.



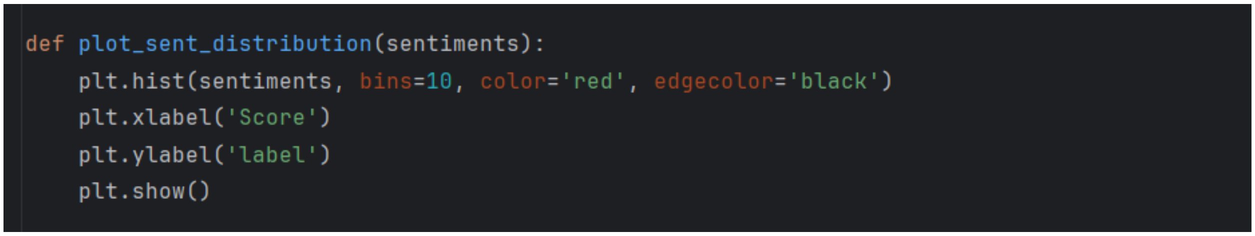
```
def remove_stopwords(tokens):
    file_tokens = []
    for word in tokens:
        if word not in stop_words:
            filtered_tokens.append(word)
    return ' '.join(file_tokens)
```

Next part of code is related to function receives as input a list of cleaned tweets, applies VADER sentiment analysis to determine each tweet's sentiment score, and outputs a list of compound sentiment scores that reflect each tweet's overall sentiment.



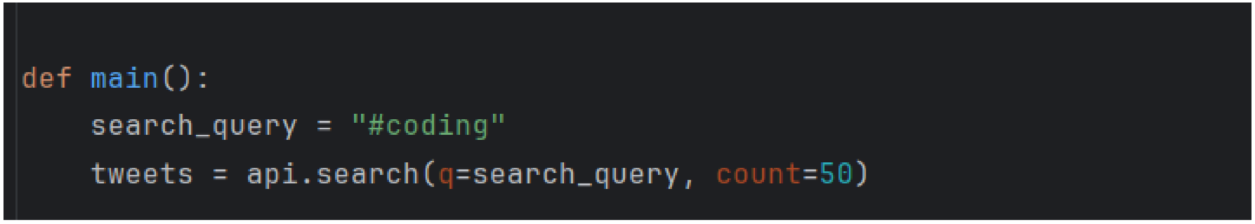
```
def analyze_sentiment(cleaned_tweets):
    return [SentimentIntensityAnalyzer().polarity_scores(tweet)['compound'] for tweet in cleaned_tweets]
```

The following codes are for function shows the plot of the histogram created from an input list of sentiment scores, allowing the user to see how the sentiment scores are distributed.



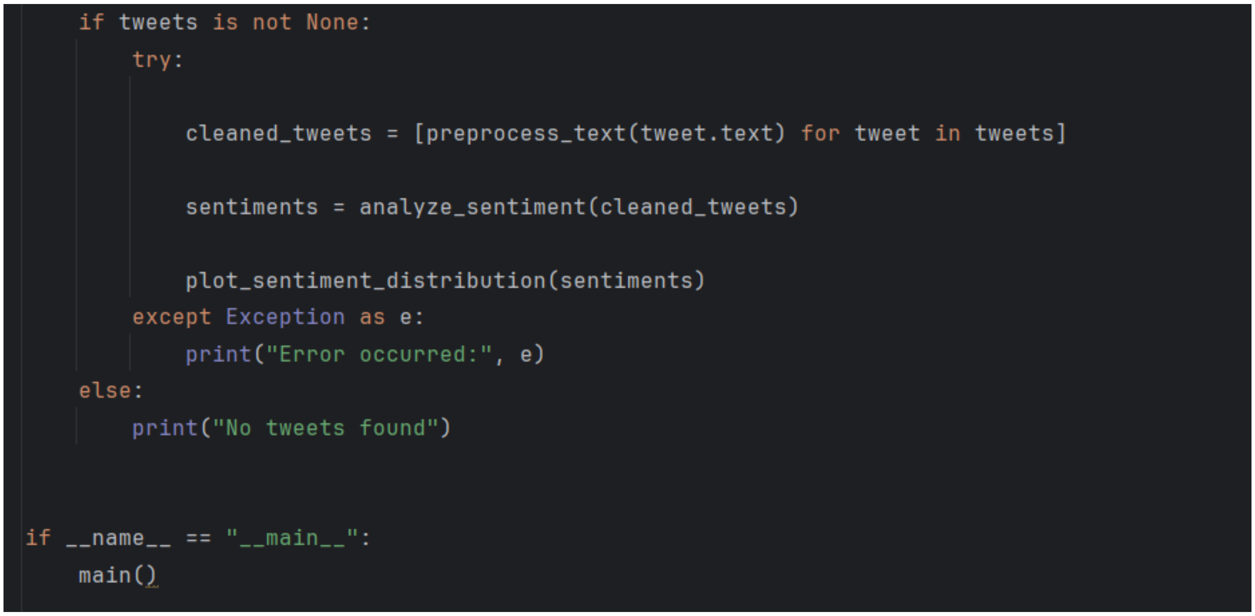
```python
def plot_sent_distribution(sentiments):
    plt.hist(sentiments, bins=10, color='red', edgecolor='black')
    plt.xlabel('Score')
    plt.ylabel('label')
    plt.show()
```

In order to facilitate additional processing and analysis of the obtained tweets, this section of the code configures the first search for tweets based on the given search criteria.



```python
def main():
    search_query = "#coding"
    tweets = api.search(q=search_query, count=50)
```

The rest of the code uses Python and the Twitter API to show a basic workflow for tweet retrieval, sentiment analysis, and sentiment distribution visualization.



```python
    if tweets is not None:
        try:

            cleaned_tweets = [preprocess_text(tweet.text) for tweet in tweets]

            sentiments = analyze_sentiment(cleaned_tweets)

            plot_sentiment_distribution(sentiments)
        except Exception as e:
            print("Error occurred:", e)
    else:
        print("No tweets found")


if __name__ == "__main__":
    main()
```

References

developer.twitter.com. (n.d.). *API Key and Secret*. [online] Available at: https://developer.twitter.com/en/docs/authentication/oauth-1-0a/api-key-and-secret.

Twitter (2023). *Twitter API Documentation*. [online] developer.twitter.com. Available at: https://developer.twitter.com/en/docs/twitter-api.

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