Graded Challenge 1 (DA1 and DA2)

Name: Helge Moes

Student Number: 11348801

WG: 02

Date: 22-02-2024

Now that we've seen how Python works and and how to use Pandas, it's time for you to combine and apply this knowledge to analyse Twitter data on your selected organisation or topic.

The warm-up challenge consists of two components:

1. **Programming**
2. **Interpretation**

Some important notes for the challenges:

1. While we of course like when you get all the answers right, the important thing is to exercise and apply the knowledge. So we will give some credit for challenges that may not be complete, as long as we see enough effort. The rubric (see Canvas) reflects this.
2. You will deliver the challenges on Canvas. Make sure to follow the turning-in instructions.

Usage of AI-generated content

In line with the course policies, it is **not allowed** to use AI-generated content to any explanations or text provided in MarkDown. This means that for **Interpretation** students are **not allowed** to use any AI-generated content or AI-assisted tools.

For **Programming**, however, AI-assisted tools (e.g., GitHub Co-Pilot or ChatGPT) may be helpful to understand how to solve some of the questions - as one may also use Google to search for solutions online (e.g., on StackOverflow or other websites). They can, therefore, be used for the programming part of this challenge exceptionally - and only for this part - unless otherwise noted in the question. It is however your responsibility to test and make sure that the solution works - and explain this, in your own words, in the interpretation section.

Facing issues?

We are constantly monitoring the issues on the GitHub to help you out. Don't hesitate to log an issue there, explaining well what the problem is, showing the code you are using, and the error message you may be receiving.

Important: We are only monitoring the repository in weekdays, from 9.30 to 17.00. Issues logged after this time will most likely be answered the next day. This means you should not wait for our response before submitting a challenge :-)

Loading [MathJax]/extensions/Safe.js

Programming

Select one of the Twitter datasets available on OneDrive. Load that file using `pandas` and answer the following questions:

1. How many tweets are in your dataset?
2. What columns does the dataset contain and what is their data type?
3. Are there any missing values?
4. What is the lowest number, highest number, average and the standard deviation of retweets and favorites?
5. What is the most common language of the tweets?
6. Who are the top 5 author_ids with the most tweets on your dataset?
7. Write a function that categorizes the tweet based on information available in one of the columns (e.g., `lang`, `metric_retweet_count`, `possibly_sensitive`, or some other column). It's up to you to define what this function should do (and explain it). One possibility is to create a function that returns 1 if the tweet belongs to a certain category, and 0 if it does not. You can be more creative than this (see Rubric on Canvas). *This week, we ask you to write the function and test if it works (following the tutorial). You will learn how to apply it to the dataframe next week.*

Note on question 7. We do not expect you to use the `text` column for the function. We will learn how to categorize that column in the next tutorials. For question 7 you are **not** allowed to use AI-assisted tools.

Using MarkDown

Make sure to combine code and markdown to answer these questions. Mention specifically the question (and question number) and the answer in markdown, relating to the code and the output of the code. Failing to do will impact the grade, as we will not be able to see whether you answered the question.

Wordcount: There is no maximum wordcount for this section.

Answers

For this excercise the data on artificial intelligence was retrieved from the OneDrive.

Preliminary Steps

In [3]:

```
# First to import the libraries and data
# Import necessary libraries
import pandas as pd
import os
from matplotlib import pyplot as plt

# To ensure from which directory I am working from
print(os.getcwd())

#import data
#read as lines file since its a jsonl
df_jsonl = pd.read_json('artificial_intelligence.jsonl', lines=True)
```

```
/Users/helgegeurtjacobusmoes/Desktop/Digital Analytics
```

The data transformation below is retrieved from DA2 - Loading Different Datasets on GitHub in order to restructure the messy Twitter data.

In [4]:

```
# Data Transformation (retrieved from the Digital Analytics Github)
# Function to extract public metrics from a row dictionary
def get_public_metrics(row):
    # Check if 'public_metrics' key exists in the row dictionary
    if 'public_metrics' in row.keys():
        # Check if the value corresponding to 'public_metrics' key is a dictionary
        if type(row['public_metrics']) == dict:
            # Iterate over key-value pairs in the 'public_metrics' dictionary
            for key, value in row['public_metrics'].items():
                # Add a new key to the row dictionary with 'metric_' prefix and set its
                row['metric_' + str(key)] = value
    # Return the modified row dictionary
    return row

# Function to process DataFrame containing tweet data
def get_tweets(df):
    # Check if 'data' column exists in the DataFrame and store results
    if 'data' not in df.columns:
        return None
    results = pd.DataFrame()
    # Iterate over values in the 'data' column, assuming each value is a list of diction
    for item in df['data'].values.tolist():
        results = pd.concat([results, pd.DataFrame(item)])

    # Apply the get_public_metrics function to each row of the results DataFrame
    results = results.apply(get_public_metrics, axis=1)

    # Reset the index of the results DataFrame and remove the old index column
    results = results.reset_index()
    del results['index']

    # Return the modified results DataFrame
    return results
```

In [5]:

```
# Check the first 5 tweets to see if it worked
tweets = get_tweets(df_jsonl)
tweets.head(5)
```

Out[5]:

	entities	public_metrics	conversation_id	created_at	edit_history_tweet_ids	autho
0	{'mentions': [{'start': 0, 'end': 12, 'usernam...	{'retweet_count': 0, 'reply_count': 0, 'like_c...	1622949079838867458	2023-02-07T13:38:41.000Z	[1622952995548717056]	1428787
1	{'mentions': [{'start': 0, 'end': 13, 'usernam...	{'retweet_count': 0, 'reply_count': 0, 'like_c...	1611857728896630784	2023-02-07T13:38:28.000Z	[1622952941098528771]	153310227184353
2	{'mentions': [{'start': 3, 'end': 14, 'usernam...	{'retweet_count': 1, 'reply_count': 0, 'like_c...	1622952934160883712	2023-02-07T13:38:27.000Z	[1622952934160883712]	87076369475353
3	{'mentions': [{'start': 3, 'end': 12, 'usernam...	{'retweet_count': 819, 'reply_count': 0, 'like...	1622952927165034496	2023-02-07T13:38:25.000Z	[1622952927165034496]	144510128727675
4	{'mentions': [{'start': 0, 'end': 12, 'usernam...	{'retweet_count': 0, 'reply_count': 0, 'like_c...	1622766700323119105	2023-02-07T13:38:24.000Z	[1622952924547624961]	1428787

5 rows × 23 columns

1. How many tweets are in your dataset?

In this case, we count the number of rows to observe the number of tweets there in total are.

In [6]:

```
#Print the number of tweets in the dataset
number_tweets = len(tweets)
number_tweets
```

Out[6]:

```
4999
```

2. What columns does the dataset contain and what is their data type?

The dataset comprises 23 columns, with the majority falling under the 'object' type, which commonly represents string values but may also include other types that defy easy categorization. Additionally, there is one column with a boolean type and several columns that consist of integer types.

In [7]:

```
def print_column_info(df):
    # Print a header indicating that the following output will display information about
    print("Columns and Data Types:")

    # Use the dtypes attribute of the DataFrame 'tweets' to print out the data types of
    print(df.dtypes)

    # Print the number of columns in the DataFrame
    print(f"\nNumber of Columns: {len(df.columns)}")

# Call the function with the DataFrame 'tweets' as argument to print column information
print_column_info(tweets)
```

```
Columns and Data Types:
entities                   object
public_metrics             object
conversation_id            object
created_at                 object
edit_history_tweet_ids     object
author_id                  object
in_reply_to_user_id        object
reply_settings             object
referenced_tweets          object
attachments                object
context_annotations        object
lang                       object
text                       object
edit_controls              object
id                         object
possibly_sensitive           bool
geo                        object
withheld                   object
metric_retweet_count        int64
metric_reply_count          int64
metric_like_count           int64
metric_quote_count          int64
metric_impression_count     int64
dtype: object

Number of Columns: 23
```

3. Are there any missing values?

In this answer, `tweets.isnull().values.any()` is used to print the missing values and associated columns. An 'if' statement is included to retrieve an answer whether there are missing values or not. If there are no missing values in the data, it prints a message that confirms this.

In the output, the '0' values indicate that there are no missing values in the corresponding columns of the dataset.

In [8]:

```
# Check if there are any missing values in the DataFrame 'tweets'
if tweets.isnull().values.any():
    # If missing values exist, print a message indicating their presence and display the
    print(f"The missing values in the dataset are the following: \n {tweets.isna().sum()
else:
    # If no missing values exist, print a message indicating their absence
    print("There are no missing values in the dataset.")
```

```
The missing values in the dataset are the following:
 entities                     61
public_metrics                 0
conversation_id                0
created_at                     0
edit_history_tweet_ids         0
author_id                      0
in_reply_to_user_id         4425
reply_settings                 0
referenced_tweets           1255
attachments                 4348
context_annotations          468
lang                           0
text                           0
edit_controls                  0
id                             0
possibly_sensitive             0
geo                         4985
withheld                    4998
metric_retweet_count           0
metric_reply_count             0
metric_like_count              0
metric_quote_count             0
metric_impression_count        0
dtype: int64
```

4. What is the lowest number, highest number, average and the standard deviation of retweets and favorites?

In this answer, the descriptive statistics are calculated with the `re_tweets_stat = tweets[''].describe()` for the 'metric_retweet_count' and 'metric_like_count' columns of the DataFrame 'tweets', because they are integers.

In [9]:

```
# Calculate the descriptive statistics for the 'metric_retweet_count' column and store t
re_tweets_stat = tweets['metric_retweet_count'].describe()

# Calculate the descriptive statistics for the 'metric_like_count' column and store them
fav_tweet_stat = tweets['metric_like_count'].describe()

# Print the descriptive statistics for retweets
print(f"Retweets descriptives: min={re_tweets_stat['min']}, max={re_tweets_stat['max']},

# Print the descriptive statistics for favorites / likes
print(f"Favorites descriptives: min={fav_tweet_stat['min']}, max={fav_tweet_stat['max']}
```

```
Retweets descriptives: min=0.0, max=27991.0, mean=202.60692138427686, std=827.1790595746
106
Favorites descriptives: min=0.0, max=1598.0, mean=0.9761952390478096, std=24.30318274060
599
```

5. What is the most common language of the tweets?

In this answer, the code takes the 'lang' (language) column from the DataFrame and counts how many observations there are. It stores the observations as a new value and indexes from '0' onwards.

In order to see if there are any other languages, a visualization is made with a library. Inspiration for this code was retrieved from: https://www.tutorialspoint.com/matplotlib/index.htm

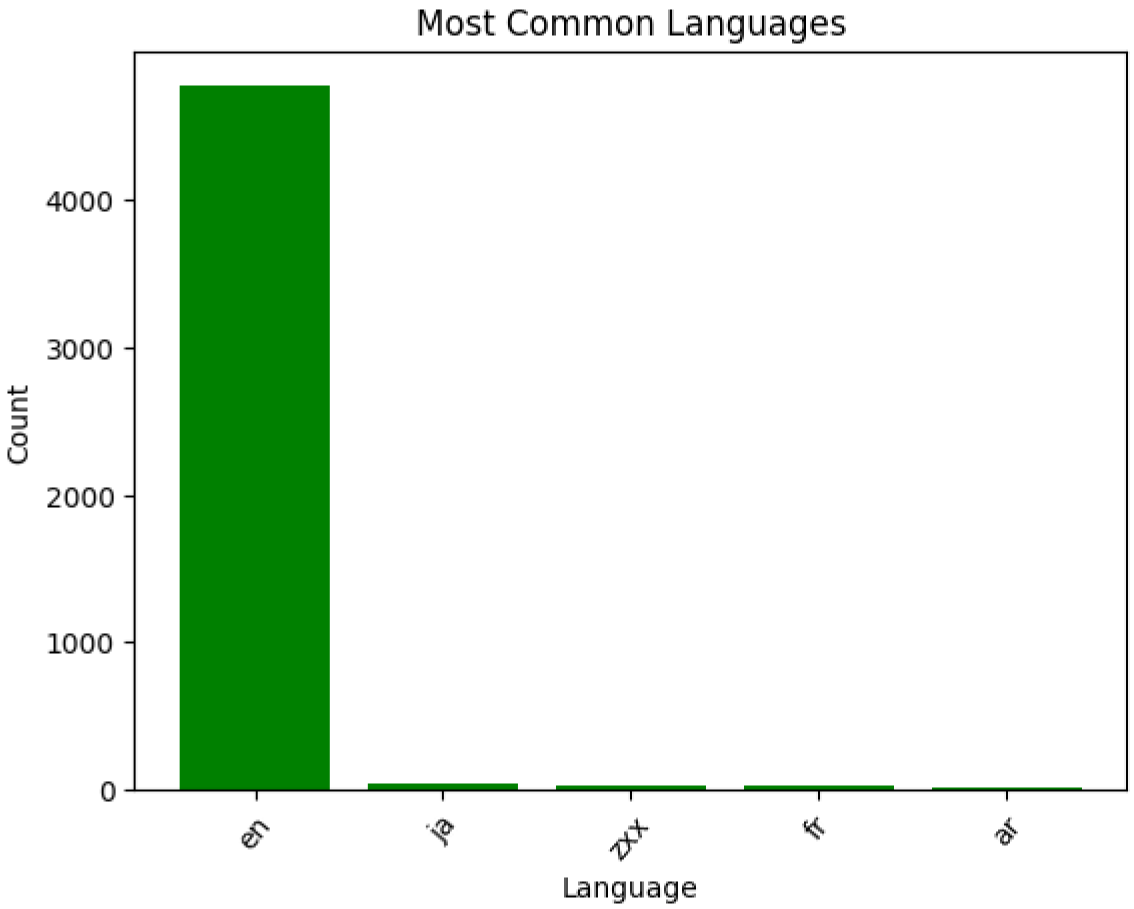
In [10]:

```
# Calculate the counts of each unique language in the 'lang' column of the DataFrame 'tw
                         = tweets['lang'].value_counts()
```

Loading [MathJax]/extensions/Safe.js

```
# Extract the index (which represents the language code) of the most frequent language
most_common_lang = lang_counts.index[0]

# Print the most common language along with its count
print(f"Most common language: {most_common_lang}")

# To make a visualization of the most popular languages, the counts of the top 5 most co
top_languages = lang_counts[:5]

# Create a new figure and axis
fig, ax = plt.subplots()

# Create a vertical bar chart and adding a green color to the graph
ax.bar(top_languages.index, top_languages.values, color='green')

# Set labels for the x and y axes
ax.set_xlabel('Language')
ax.set_ylabel('Count')

# Set the title of the plot
ax.set_title('Most Common Languages')

# Rotate the x-axis labels for better readability
plt.xticks(rotation=50)

# Display the plot
plt.show()
```

```
Most common language: en
```



6. Who are the top 5 author_ids with the most tweets on your dataset?

In this answer, the number of tweets are counted for each user and saved in a variable. Afterwards, the 5 with the most counts from the list are saved to another variable. This is also repeated for the authors.

In [11]:

```
# Calculate the counts of each unique author_id in the 'author_id' column of the DataFra
author_counts = tweets['author_id'].value_counts()

# Select the top 5 authors with the most tweets
top_5_authors = author_counts.head(5)

# Print the top 5 author_ids along with the count of tweets for each author
print("Top 5 authors with the most tweets:")
for author_id, count in top_5_authors.items():
    print(f"{author_id}: {count}")
```

```
Top 5 authors with the most tweets:
1517099246478147585: 52
1396125269925285888: 49
1421511710259744779: 43
317868471: 40
1622701737516695560: 35
```

7. Write a function that categorizes the tweet based on information available in one of the columns

For this section, the objective is to conduct a basic sentiment analysis in order to categorize each post by 'negative', 'neutral' or 'positive.'

The code snippet utilizes TextBlob, a sentiment analysis tool, to evaluate the sentiment of text data. It defines a function called `get_sentiment` that calculates the polarity score of the text, categorizing it as positive, negative, or neutral based on the score. The function is then applied to the 'text' column of a DataFrame, storing the sentiment of each tweet in a new column named 'sentiment', and finally, it counts the occurrences of each sentiment category in the dataset.The code snippet utilizes TextBlob, a sentiment analysis tool, to evaluate the sentiment of text data. It defines a function called `get_sentiment` that calculates the polarity score of the text, categorizing it as positive, negative, or neutral based on the score. The function is then applied to the 'text' column of a DataFrame, storing the sentiment of each tweet in a new column named 'sentiment', and finally, it counts the occurrences of each sentiment category in the dataset.

This analysis was inspired by the following references:

TextBlob: Simplified Text Processing—TextBlob 0.16.0 documentation. (n.d.). Retrieved February 21, 2024, from https://textblob.readthedocs.io/en/dev/

How to build a Twitter sentiment analyzer in Python using TextBlob. (2018, October 24). freeCodeCamp.Org. https://www.freecodecamp.org/news/how-to-build-a-twitter-sentiments-analyzer-in-python-using-textblob-948e1e8aae14/How to build

Firstly, libraries are imported for the sentiment analysis and TextBlob to classify the tweets.

In [12]:

```
# Import libraries for sentiment analysis
import nltk
import re
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer
```

```
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')

# Import the classifier
from textblob import TextBlob

# If libraries are succesfully imported, it responds with a 'true'
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     /Users/helgegeurtjacobusmoes/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data]     /Users/helgegeurtjacobusmoes/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package wordnet to
[nltk_data]     /Users/helgegeurtjacobusmoes/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
```

Secondly, the data needs to be cleaned in order to get rid of the noise. By removing special characters and stopwords, the classifier becomes more accurate in identifying sentiment.

In [13]:

```
# Define a function to clean text data
def clean_text(text):
    # Remove special characters and punctuation using regular expressions
    text = re.sub(r'[^\w\s]', '', text)

    # Remove URLs using regular expressions
    text = re.sub(r'https?:\/\/\S+', '', text)

    # Remove mentions (@username) using regular expressions
    text = re.sub(r'@[A-Za-z0-9]+', '', text)

    # Remove hashtags (#hashtag) using regular expressions
    text = re.sub(r'#[A-Za-z0-9]+', '', text)

    # Tokenize the text
    tokens = word_tokenize(text)

    # Convert tokens to lowercase
    tokens = [token.lower() for token in tokens]

    # Remove stopwords using NLTK's stopwords list
    stop_words = set(stopwords.words('english'))
    tokens = [token for token in tokens if not token in stop_words]

    # Lemmatize tokens to their base form
    lemmatizer = WordNetLemmatizer()
    tokens = [lemmatizer.lemmatize(token) for token in tokens]

    # Join the tokens back into a string
    text = ' '.join(tokens)

    # Return the cleaned text
    return text

# View clean text
print(tweets['text'].head(10))
```

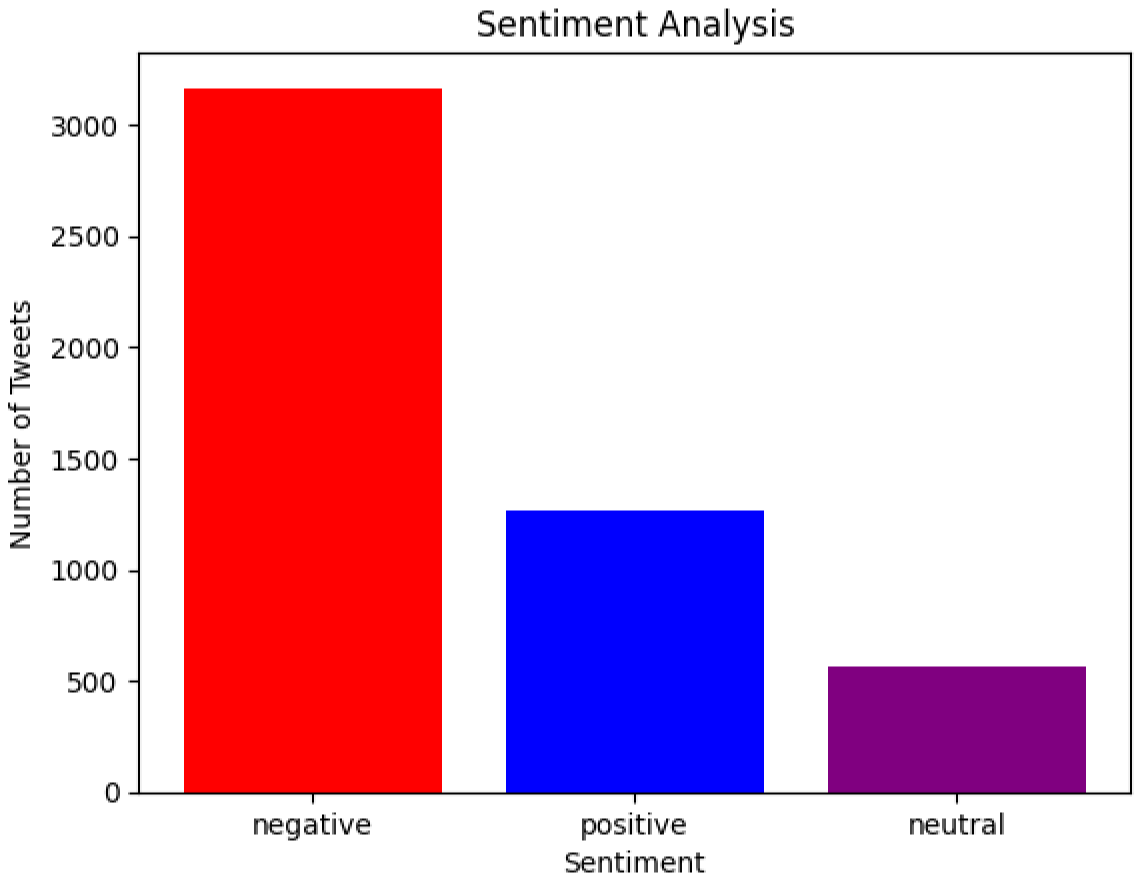
Loading [MathJax]/extensions/Safe.js

```
0    @bscgemspump Check out this artificial intelli...
1    @IncomeSharks Are you looking to make the most...
2    RT @MemekingAi: @EndTheGlobe @b4sicb1tchSPC @M...
3    RT @GPTChain: GPT Chain aims to establish the ...
4    @NikolaBench Check out this artificial intelli...
5    RT @BusInsiderSSA: #MARKETS | ChatGPT rival Ba...
6    RT @Uncle_Gober22: Teknologi AI nantinya akan ...
7    RT @robotdogeaibsc: Welcome to RobotDogeAi\n\n...
8    #SingularityNET $AGIX is a blockchain-powered ...
9    RT @KeAiPublishing: Artificial Intelligence in...
Name: text, dtype: object
```

In [14]:

```python
# Import the classifier and matplotlib
from textblob import TextBlob
import matplotlib.pyplot as plt

# Define the sentiment analysis function
def get_sentiment(text):
    # Create a TextBlob object from the input text
    blob = TextBlob(text)

    # Get the polarity score of the text
    sentiment = blob.sentiment.polarity

    # Classify the sentiment based on the polarity score
    if sentiment > 0:
        return 'positive'
    elif sentiment < 0:
        return 'negative'
    else:
        return 'neutral'

# Apply the sentiment analysis function to the 'text' column of the DataFrame
tweets['sentiment'] = tweets['text'].apply(get_sentiment)

# Count the number of tweets in each sentiment category
sentiment_counts = tweets['sentiment'].value_counts()

# Define colors for each sentiment category
colors = {'positive': 'blue', 'negative': 'red', 'neutral': 'purple'}

# Create a bar chart of the sentiment counts with specified colors
plt.bar(sentiment_counts.index, sentiment_counts.values, color=[colors[sentiment] for se

# Set the chart title and axis labels
plt.title("Sentiment Analysis")
plt.xlabel("Sentiment")
plt.ylabel("Number of Tweets")

# Show the chart
plt.show()
```

Loading [MathJax]/extensions/Safe.js



Based on the visualization of the sentiment analysis, which categorizes tweets into positive, negative, or neutral sentiments, it appears that Twitter users predominantly express negative sentiments towards the topic of artificial intelligence. To gain a deeper understanding of the reason behind this sentiment, a further analysis is conducted to examine examples to observe how the tweets are categorized by sentiment. Moreover, this is achieved by showcasing the count of tweets within each sentiment category and a sample of 10 tweets from each category, offering a more detailed and comprehensive depiction of the sentiment distribution.

In [15]:

```
# Group the tweets by their sentiment category
grouped_tweets = tweets.groupby('sentiment')

# Print the number of tweets in each sentiment category
print("Sentiment count:")
print(grouped_tweets.size())

# Print a sample of tweets from each sentiment category
for sentiment, group in grouped_tweets:
    print(f"Sentiment: {sentiment}")
    # Print a sample of 10 tweets from each sentiment category
    print(group['text'].sample(10))
    print('\n')
```

```
Sentiment count:
sentiment
negative    3166
neutral      563
positive    1270
dtype: int64
Sentiment: negative
1300    RT @spectatorindex: Google is set to launch Ch...
4844    @cryptojack #AIShiba - The first Artificial In...
3002    RT @Jackheppika: @iambroots $AIS\nA project th...
1105    Drishti Marine, the state-appointed lifeguard ...
1927    RT @Yasmin2186: Along with his digital data, a...
3343    "Meta, Long an A.I. Leader, Tries Not to Be Le...
3727    RT @Cryptic_Maestra: @CryptoEmdarks #AIShiba -...
2415    RT @AFP: AI in 2023: what's working now?\n\n#A...
2024    RT @spectatorindex: BREAKING: Google set to la...
3027    artificial intelligence integration partnershi...
Name: text, dtype: object


Sentiment: neutral
3226    RT @HannahOctavio1: #الذكاء الاصطناعي\n#مجالات...
2980    RT @real_alethea: ✨Medical #AI is advancing in...
312     RT @jc_stubbs: Working with @satariano and @pa...
878     RT @b4sicb1tchSPC: @MrBigWhaleREAL #XRDC XRPL ...
2685                                 https://t.co/g6QPgONV4r
2703                                 https://t.co/88X3P1wSFf
825                                  https://t.co/teHUv8jO5J
4408    @altcryptocom #XRDC XRPL DIGITAL CURRENCY\nhtt...
4476    OpenAI CTO Mira Murati says regulators must go...
3654    RT @coingecko: What are #AI tokens and what ar...
Name: text, dtype: object


Sentiment: positive
1006    RT @robotdogeaibsc: Welcome to RobotDogeAi\n\n...
932     Google is coming up with its own version of #C...
1598    RT @robotdogeaibsc: Welcome to RobotDogeAi\n\n...
4546    RT @LindaGrass0: Adopting MLOps means putting ...
4028    #Google launches #Osmo, an #AI to predict new ...
254     RT @ElianPeltier: The usage of deepfakes for p...
713     RT @satariano: For years, we've heard warning ...
1690    Google is rolling out a new conversational art...
1001    RT @robotdogeaibsc: Welcome to RobotDogeAi\n\n...
4578    @ianheinischmma #AIShiba - The first Artificia...
Name: text, dtype: object
```

In conclusion, the classifier succesfully portrays the overarching sentiment on artificial intelligence on Twitter, which is mostly negative.

However, this result may not be representative and can be considered to emphasize common words and labels, since it is based on a preprocessed dataset and utilizes a pretrained classfier.

For future research, it would be interesting to explore if there are any other languages present in the data and to clean the data further in order to get rid of more noise, such as 'RT' and '@'.

Intepretation

Explaining your process

Write a brief description explaining how you created or assembled the code (i.e., sources used, and how did you adjust the code). In this description, include a brief reflection of the main challenges you had during the process, and how you handled these challenges.

Wordcount: maximum of 200 words. Please inform the wordcount for this answer

Answer

I was able to code this challenge due to my previous experience in coding during my bachelors and my masters.

In general, I tend to refer to documentation in packages / library publishers and then change the code to fit my variables. Examples of where these codes or libraries are from are:

The Python Standard Library. (n.d.). Python Documentation. Retrieved February 21, 2024, from https://docs.python.org/3/library/index.html

Arcila, W. van A., Damian Trilling &. Carlos. (2022, March 11). Computational Analysis of Communication. https://cssbook.net/

How to build a Twitter sentiment analyzer in Python using TextBlob. (2018, October 24). freeCodeCamp.Org. https://www.freecodecamp.org/news/how-to-build-a-twitter-sentiments-analyzer-in-python-using-textblob-948e1e8aae14/

TextBlob: Simplified Text Processing—TextBlob 0.16.0 documentation. (n.d.). Retrieved February 21, 2024, from https://textblob.readthedocs.io/en/dev/

Read JSON file using Python. (2019, December 17). GeeksforGeeks. https://www.geeksforgeeks.org/read-json-file-using-python/

Matplotlib Plot. (n.d.). Retrieved February 20, 2024, from https://www.tutorialspoint.com/matplotlib/index.htm

In the case I alter the code, I tend to have bugs or other errors. Therefore, I spend a considerable amount of time debugging, so the variable names correspond to the dataset and ensure that the data is properly processed.

For this assignment I found it particularly challenging to work with a JSON file, since I did not have experience with this before. In this case, the process of translating it into tabular form is especially difficult. By resorting to YouTube tutorials or search for an online course on data structures, similar to the Google Analytics Training we had to complete for this course, I was able to familiarize myself with this issue.

Although I am able to follow the steps and code on github to a certain degree, it is difficult to visualize what the data does. Therefore, I run multiple examples until I understand how the data is structured and how I can retrieve relevant information for an analysis.

Wordcount: 188 (excluding the references)

Explaining your code

You used a lot of code above to answer the seven questions. Select what you consider to be the three most important steps in the code, justify why they are important, and explain the commands being used.

Note: there are different correct answers to what the three most important steps of the code are. We want to understand your reasoning/justification for this, and will accept different answers depending on the logic and clarity of your argumentation.

Wordcount: maximum of 200 words. Please inform the wordcount for this answer

Answer

In this case, the answer depends on the objective of the analysis. For this challenge, it was to engage in the basic practices of exploratory data analysis of twitter data and a basic analysis by using a classifier for a sentiment analysis.

Firstly, I consider the preliminary steps as fundamental for running the right packages/libraries/data. For instance, a standard utility module such as `os` allows to find the directory of the file that is being analyzed. This is integral for digital analytics, since we need to know about where we are drawing our conclusions from and through what means. Without understanding these components, you as a coder have no idea what the possibilities and limitations are and this may affect the exploration of the data.

Secondly, preparing the data for analysis and making it comprehensible is a fundamental step in coding. In this case, it is also important to properly annotate the code with # and keep explaining what a line of code is contributing to throughout the notebook.

Lastly, regarding the conclusion of the sentiment analysis, it is important that the pre-processing of the data is thoroughly executed in order to get accurate results. The performance of a model is directly tied to the quality of the data it utilizes, using the code `print()` is essential to understand and get insight in this data. If there is a lot of noise in the data, the NLP model cannot process the text properly and the data needs to be more thoroughly cleaned.

Wordcount: 196

Quality assurance

You are assembling code from different sources - some of it you learned in the tutorial, you may have reused code we provided, and in some cases you may have used different online sources. But how do you know that the code actually worked and delivered the expected results? Please explain how someone else can verify that the code worked (e.g., what should they look at) and which steps you built into the code for yourself to know that it worked (explicitly indicate at least three steps)

Wordcount: maximum of 200 words. Please inform the wordcount for this answer

Answer

Firstly, if another reader operates the code on a different page, they need to be aware of the necessary libraries. Usually I start each project by making sure that all used libraries are imported first when used in the code, for example `pd.read_json` can only run if the `pandas` library is imported as `pd`.

Secondly, similar to writing documents, it is crucial to often save the notebook and select "Restart & Run all" in the Kernel drop-down menu in order to prevent possible errors.

Lastly, if there are errors, I would ensure that the code runs in a correct sequence. After writing multiple cells of code, I would check whether no objects are referred to before they have been created.

With regard to sentiment analysis, the code that is used can be portrayed as a 'black box.' This means that we are never truly sure what happens to the data, but we do know the input and output. By going back to the dataset, going through the steps in this challenge and coming to the same outcomes that seem to do what is expected, we can safely assume that the code works properly.

Wordcount: 193