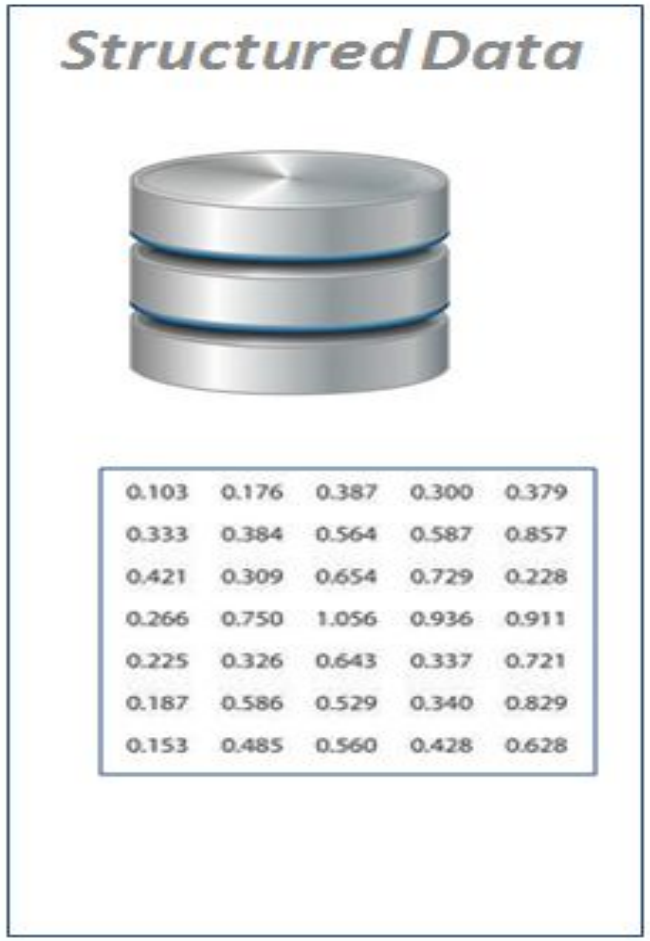
Text Mining and NLP

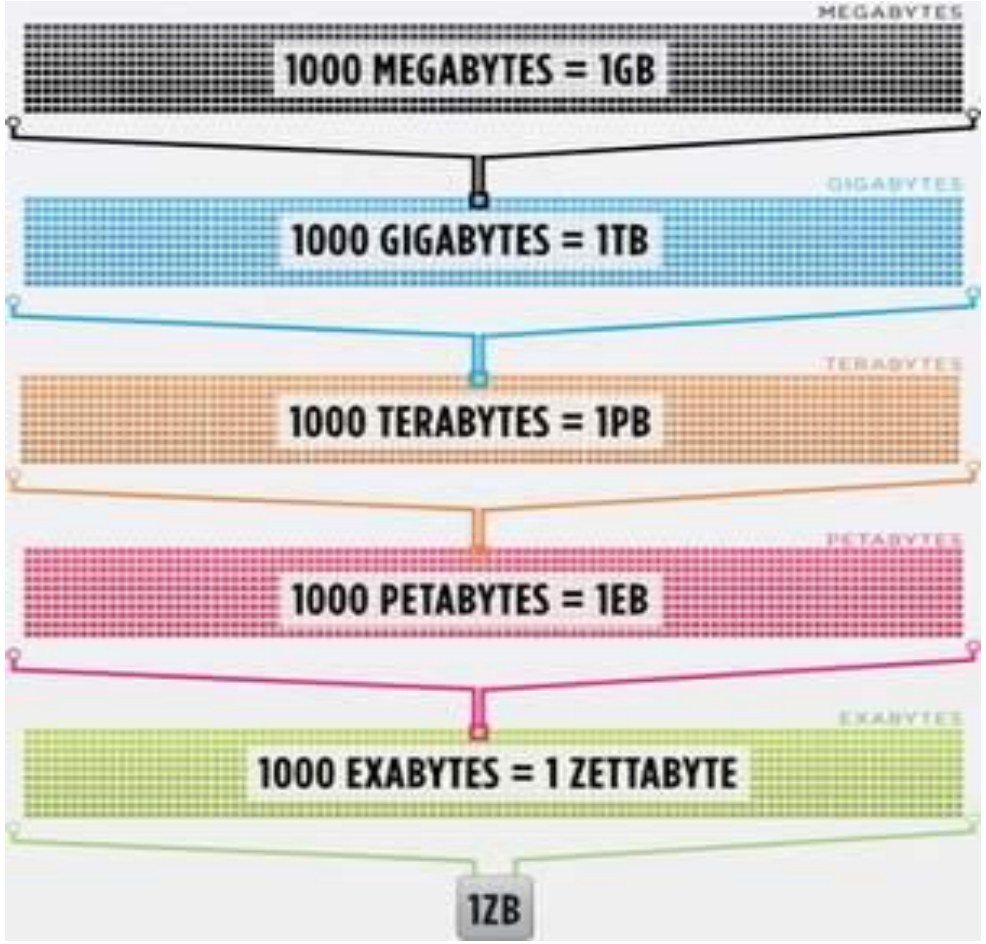
Contents

1. Structured and Unstructured Data
2. Features of Unstructured Data
3. What is Content Analysis ?
4. What is Text Analysis ?
5. Case Study
6. Text Mining in Python
7. Word Cloud in Python
8. Text Mining Using Matplotlib

Structured Vs. Unstructured Data



Unstructured Data Growth



- Research from IDC (International Data Corporation) shows that unstructured content accounts for 95% of all digital information, with estimate of compound annual growth at 65%

- By 2020, IDC predicts the volume of digital data will have reached 40,000 EB or 40 ZB

Features Of Unstructured Data

Does not reside in traditional databases and data warehouses

May have an internal structure, but does not fit a relational data model

Generated by both humans and machines

- Textual and social media content
- Machine-to-machine communication

Examples Of Unstructured Data

Examples of unstructured data include:

- **Personal messaging** – Email, instant messages, tweets, chat
- **Business documents** – Business reports, presentations, survey responses
- **Web content** – Web pages, blogs, wikis, audio files, photos, videos
- **Sensor output** – Satellite imagery, geo-location data, scanner transactions

Value Of Unstructured Data

Unstructured data provides a rich source of information about people, households and economies.

- It may enable more accurate and timely measurement of a range of demographic, social, economic and environmental phenomena
  - When combined with traditional data sources
  - As a replacement for traditional data sources

- As a result, it presents unprecedented opportunities for official statistics to
  - Improve delivery of current statistical outputs
  - Create new information products not possible with traditional data sources

What is Content Analysis ?

- For unstructured data to be useful, **it must be analysed to extract and expose the information it contains.**

- Content analysis is used to quantify and analyze the presence, meanings and relationships of certain words, themes, or concepts.

- Different types of analysis are possible, such as:

  - **Entity analysis** – People, organisations, objects and events, and the relationships between them.

  - **Topic analysis** – Topics or themes, and their relative importance.

  - **Sentiment analysis** – Subjective view of a person to a particular topic.

What Is Text Analysis ?

- Text Mining is also known as **Text Data Mining (TDM)** and **Knowledge Discovery in Textual Database (KDT)**

- It is a process of identifying novel information from a collection of texts (Also known as a ‘Corpus’)

- **Corpus is a collection of ‘documents’ containing natural language text.** Here, documents, generally, are sentences. Each document is represented as a separate line.

Search Vs. Discover

	Search (Goal-oriented)	Discover (Opportunistic)
Structured Data	Data Retrieval	Data Mining
Unstructured Data (Text)	Information Retrieval	Text Mining

Case Study – HR Appraisal Process Feedback

Background

- The company XYZ carried out Annual Performance Appraisal process which is a routine HR process.
- The employees were asked to give feedback about the overall process and questions used for assessing their performance level.

Objective

- To understand the employee sentiments and incorporate recommendations in the current performance appraisal process.

Available Information

- Feedback and comments from the employees were stored in a text document.

Data Snapshot

HR Appraisal process

Text Observations

The process was transparent.
There is a lot of scope to improve the process, as most questions were subjective.
Happy with the process, but salary increment in 2019 is very low as compared to previous years.
Many questions were very subjective. Very difficult to measure the performance.
Questions could have been specific to function. Very general questions.
More research is required to come out with better process next time.
Very happy with the process adopted. Fair and transparent.

* These are the comments received from employees.
Note that, data is not in structured format.

Text Mining In Python

```
#Install NLTK library in Anaconda Prompt
```

```
pip install nltk
```

```
#Import NLTK library
#Import data and convert into ‘Corpus’
```

```
import nltk
nltk.download()
from nltk.book import *
text = [line.rstrip() for line in open("HR Appraisal
process.txt")]
text[0:5]
```

- Install and load NLTK(Natural Language Toolkit) library.
- rstrip() reads all text lines from a file or connection.
- rstrip() interprets each element of the vector as a document. It converts and saves data as a corpus.

* Note : When imported nltk, nltk.download() will download the required libraries from NLTK for text mining. Run nltk.download() only for the first time

Text Mining In Python

```
# Output:
['The process was transparent.',
 'There is a lot of scope to improve the process, as most questions were
subjective.',
 'Happy with the process, but salary increment in 2019 is very low as
compared to previous years.',
 'Many questions were very subjective. Very difficult to measure the
performance.',
 'Questions could have been specific to function. Very general questions.']
```

Interpretation:

- **text[0:5]** prints first 5 text lines from the data with each line as one set of strings.

```
# Display a particular document from corpus.
```

```
text[2]
```

```
'Happy with the process, but salary increment in 2019 is very low as
compared to previous years.'
```

- **text[2]** prints text line of specified number in []. Here it is printing 3rd line.
- Python indexing starts from 0, thus 2 represents 3rd data point (sentence).

Text Mining In Python

```
# Clean the Corpus for further analysis
corp = [item.lower() for item in text]
corp [2]
```

```
'happy with the process, but salary increment in 2019 is very low as
compared to previous years.'
```

```
from string import punctuation
remove_punc = str.maketrans('','', punctuation)
corp = [item.translate(remove_punc) for item in corp]
corp[2]
```

```
'happy with the process but salary increment in 2019 is very low as
compared to previous years'
```

```
from string import digits
remove_digits = str.maketrans('', '', digits)
corp = [item.translate(remove_digits) for item in corp]
corp[2]
```

```
'happy with the process but salary increment in is very low as
compared to previous years'
```

- **lower()** converts text to lowercase.
- **maketrans(",", punctuation)** removes punctuation.
- **maketrans(", ", digits)** removes numbers.

Text Mining In Python

```
# Clean the Corpus for further analysis
```

```
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
stop_words = nltk.corpus.stopwords.words('english')
fs=[]
for item in corp:
    word_tokens = word_tokenize(item)
    filtered_sentence = [w for w in word_tokens if not w in
stop_words]
    fs.append(filtered_sentence)
fs[2]
```

```
['happy', 'process', 'salary', 'increment', 'low', 'compared’,
'previous', 'years']
```

- **stopwords("english")** remove stop words like: i, me, our, and, the, is, etc. There are more than 100 in-built English Stopwords in Python NLTK. Use stopwords("english") to view the list of these stopwords.

Text Mining In Python

```
# Clean the Corpus for further analysis
newStopWords = ['process']
stop_words.extend(newStopWords)
fs=[]
for item in corp:
    word_tokens = word_tokenize(item)
    filtered_sentence = [w for w in word_tokens if not w in
stop_words]
    fs.append(filtered_sentence)
fs[2]
```

```
['happy', 'salary', 'increment', 'low', 'compared', 'previous',
'years']
```

- If you wish to remove specific words from the corpus use **.extend("word")** to add the word in list of stopwords. Here "**process**" word is removed.

Text Mining In Python

```
# Convert to term-document matrix format
```

```
import itertools
filtered_text = list(itertools.chain.from_iterable(fs))
fdist = nltk.FreqDist(filtered_text)
```

- **FreqDist()** gives frequency of each word in the list

```
# Find most common words i.e. words having highest frequency
```

```
fdist.most_common(10)
```

```
[('questions', 13),
 ('hr', 12),
 ('happy', 10),
 ('subjective', 8),
 ('fair', 7),
 ('performance', 6),
 ('work', 6),
 ('difficult', 5),
 ('measure', 5),
 ('salary', 4)]
```

Interpretation:

- “questions”, “hr”, “happy”, “subjective”, “fair”, “performance”, “work”, “difficult”, “measure”, ”salary” are the top 10 words by frequency.
- The frequencies of the words are listed besides them.

- **fdist.most_common(n)** gives the list of top n words sorted highest to lowest by frequency

Word Cloud In Python

Word cloud, as the name suggests, is an **image showing compilation of words**, in which, **size of words indicates its frequency or importance**.

```
# Install the library “wordcloud” in Anaconda Prompt
```

```
pip install wordcloud
```

```
# Get Word Cloud
```

```
from wordcloud import WordCloud
import matplotlib.pyplot as plt
wordcloud =
WordCloud(background_color="white").generate(str(filtered_text))
plt.figure(figsize = (8, 8))
plt.imshow(wordcloud); plt.axis("off")
plt.tight_layout(pad = 0); plt.show()
```

- **background.color** allows you to select the color of the background.
- **fig.size** allows you to adjust the size/dimensions of the wordcloud.
- **plt.imshow()** is used to display data as an image.
- **plt.axis("off")** means axis lines and labels are turned off.
- **plt.tight_layout()** automatically adjusts subplot parameters to give specified padding (**here 0**).

Word Cloud In Python

```
# Output :
```

Interpretation:

- Word ‘questions’ has largest size, indicating most frequent word followed by ‘happy’ and ‘hr’ and so on..

Text Mining Using Matplotlib

```
# Plotting frequent terms as a bar plot
```

```
a = fdist.most_common(10)
```

```
# Transform as a dataframe
```

```
import pandas as pd
b = pd.DataFrame(a)
b = b.rename(columns={0:'Words',1:'Freq'})
```

```
# Horizontal bar plot
```

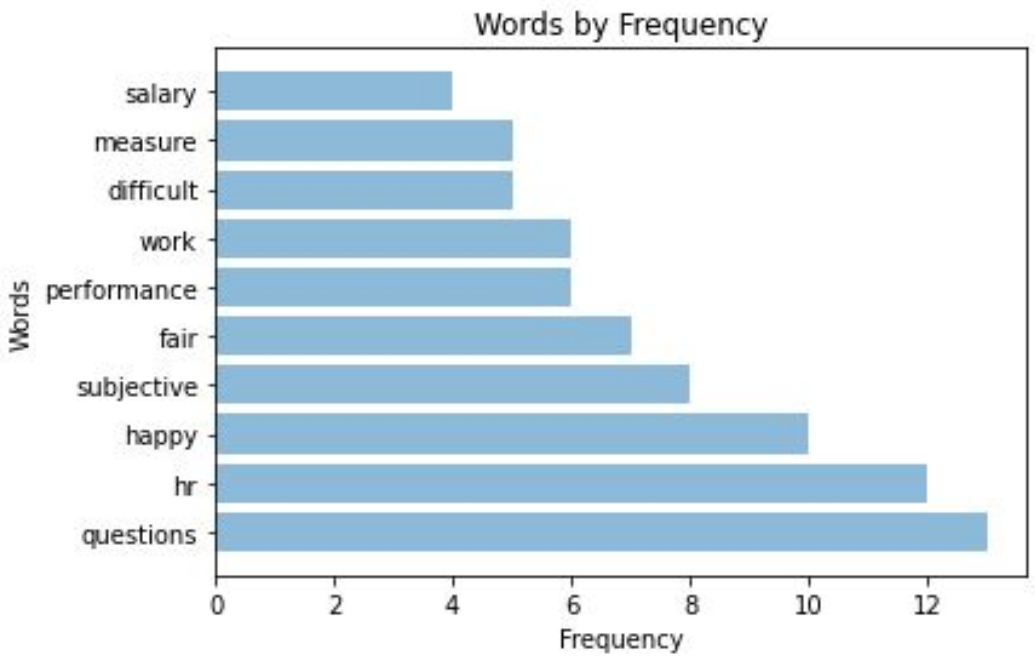
```
import numpy as np
c=b.Words
y=np.arange(len(c))
x=b.Freq

plt.barh(y, x, align='center', alpha=0.5)
plt.yticks(y, c);plt.ylabel('Words')
plt.xlabel('Frequency');plt.title('Words by Frequency')

plt.show()
```

Text Mining Using Matplotlib

```
# Output :
```



Interpretation:

- Graph shows the frequency of the top 10 words by frequency on a horizontal bar graph. “Very” is the most frequent word with frequency 14.

Quick Recap

In this session, we learnt **Text Mining in Python** :

Unstructured Data	• Does not reside in traditional databases and data warehouses. • Example: emails, tweets, feedback, blogs, webpages, etc.
Text Analysis	• Process of identifying novel information from a collection of texts. (Also known as a 'Corpus')
Text mining in Python	• Install '**nltk**' library. Convert data into corpus. • Clean the corpus: convert all words to lowercase/uppercase, remove punctuation, numbers, stopwords, words.
Word Cloud in Python	• An image showing compilation of words, in which, size of words indicates its frequency or importance. • Install '**wordcloud**' library.

Natural Language Processing

Contents

1. Tokenization
2. What Is NLP & it’s applications
3. Sentiment Analysis
4. Sentiment Analysis in Python library “TextBlob“
5. Sentiment Analysis in Python library “vader"

Tokenization

- Tokenization is the process of tokenizing or splitting a string, text into a list of tokens.
- Example : a word is a token in a sentence, and a sentence is a token in a paragraph.

```
# Tokenization
# Import Tokenize from NLTK
# Import data as paragraph string object
```

```
from nltk.tokenize import sent_tokenize, word_tokenize
file_data = open('HR Appraisal process.txt', 'r')
data = file_data.read()

print(sent_tokenize(data))
print(word_tokenize(data))
```

- ❑ **open()** loads a text file in read only format.
- ❑ **read()** reads the text in the object file_data.
- ❑ **sent_tokenize()** tokenizes text data into sentences or sentence tokens.
- ❑ **word_tokenize()** tokenizes text data into word tokens.

* Note : Here we continue using previous data, 'HR Appraisal process' data.

Tokenization

```
# Output :
```

```
['The process was transparent.', 'There is a lot of scope to improve the process, as most questions were subjective.', 'Happy with the process, but salary increment in 2019 is very low as compared to previous years.', 'Many questions were very subjective.', 'Very difficult to measure the performance.', 'Questions could have been specific to function.', 'Very general questions.', 'More research is required to come out with better process next time.', 'Very happy with the process adopted.', 'Fair and transparent.', 'Salary increment is extremely low as compared to industry benchmark.', 'Not happy with rating methodology.', 'Very subjective questions.', 'Excellent effort by HR team.', 'Very fair process.', 'Congratulations to HR department.', 'Very fair process.', 'The process needs lot of improvement.', 'More frequent discussion with manager is required.', 'It is difficult to measure performance using current approach.', 'Very subjective questions.', 'Not possible to evaluate.', 'Excellent work by HR.', 'Congratulations.', 'Happy with the process.', 'Some scope to improve.', 'The process was fair.', 'Most questions were subjective.', 'Salary increment is very low.', 'Little disappointed.', 'Very difficult to measure the performance with this approach.', 'Not happy with the process.', 'Very biased.', 'Need better process next time.', 'Fair and transparent work by HR,\nSalary increment not as expected.', 'Not happy with method used to evaluate performance.', 'Very fair process by HR.', 'Congratulations to HR.', 'Good work.', 'The process needs lot of changes.', 'It is difficult to measure performance using this method.', 'Very subjective questions.', 'Excellent work by HR.', 'Very clear process.', 'Happy with the process.', 'Better to do twice a year.', 'We can hire consultant to come out with better method.', 'Last year method was clearer than this year.', 'Many changes are required in self-assessment questions.', 'The questions were biased toward particular department.', 'The questions were so subjective.', 'Difficult to measure performance.', 'I think HR department should research more on appraisal process.', 'Very happy with the way process was carried out in our organization.', 'I would be happy if few questions are modified during next appraisal process.', 'I am satisfied with overall communication and the process.', 'The process was fair.', 'Some scope to improve remains.', 'Good work by HR.', 'Keep it up.', 'Excellent show by our HR team members.', 'Very happy.', 'Few minor changes will make the process more robust.', 'Subjective questions can be replaced or removed.', 'Nice work by HR head.Very smooth process.', 'Overall good process.Must appreciate HR team.']
```

Interpretation:

- **sent_tokenize()** converts the text into separate sentences.

Tokenization

```
# Output :
['The', 'process', 'was', 'transparent', '.', 'There', 'is', 'a', 'lot', 'of', 'scope', 'to', 'improve', 'the', 'process', ',',
'as', 'most', 'questions', 'were', 'subjective', '.', 'Happy', 'with', 'the', 'process', ',', 'but', 'salary', 'increment', 'in',
'2019', 'is', 'very', 'low', 'as', 'compared', 'to', 'previous', 'years', '.', 'Many', 'questions', 'were', 'very', 'subjective',
'.', 'Very', 'difficult', 'to', 'measure', 'the', 'performance', '.', 'Questions', 'could', 'have', 'been', 'specific', 'to',
'function', '.', 'Very', 'general', 'questions', '.', 'More', 'research', 'is', 'required', 'to', 'come', 'out', 'with',
'better', 'process', 'next', 'time', '.', 'Very', 'happy', 'with', 'the', 'process', 'adopted', '.', 'Fair', 'and',
'transparent', '.', 'Salary', 'increment', 'is', 'extremely', 'low', 'as', 'compared', 'to', 'industry', 'benchmark', '.', 'Not',
'happy', 'with', 'rating', 'methodology', '.', 'Very', 'subjective', 'questions', '.', 'Excellent', 'effort', 'by', 'HR', 'team',
'.', 'Very', 'fair', 'process', '.', 'Congratulations', 'to', 'HR', 'department', '.', 'Very', 'fair', 'process', '.', 'The',
'process', 'needs', 'lot', 'of', 'improvement', '.', 'More', 'frequent', 'discussion', 'with', 'manager', 'is', 'required', '.',
'It', 'is', 'difficult', 'to', 'measure', 'performance', 'using', 'current', 'approach', '.', 'Very', 'subjective', 'questions',
'.', 'Not', 'possible', 'to', 'evaluate', '.', 'Excellent', 'work', 'by', 'HR', '.', 'Congratulations', '.', 'Happy', 'with',
'the', 'process', '.', 'Some', 'scope', 'to', 'improve', '.', 'The', 'process', 'was', 'fair', '.', 'Most', 'questions', 'were',
'subjective', '.', 'Salary', 'increment', 'is', 'very', 'low', '.', 'Little', 'disappointed', '.', 'Very', 'difficult', 'to',
'measure', 'the', 'performance', 'with', 'this', 'approach', '.', 'Not', 'happy', 'with', 'the', 'process', '.', 'Very',
'biased', '.', 'Need', 'better', 'process', 'next', 'time', '.', 'Fair', 'and', 'transparent', 'work', 'by', 'HR', ',', 'Salary',
'increment', 'not', 'as', 'expected', '.', 'Not', 'happy', 'with', 'method', 'used', 'to', 'evaluate', 'performance', '.',
'Very', 'fair', 'process', 'by', 'HR', '.', 'Congratulations', 'to', 'HR', '.', 'Good', 'work', '.', 'The', 'process', 'needs',
'lot', 'of', 'changes', '.', 'It', 'is', 'difficult', 'to', 'measure', 'performance', 'using', 'this', 'method', '.', 'Very',
'subjective', 'questions', '.', 'Excellent', 'work', 'by', 'HR', '.', 'Very', 'clear', 'process', '.', 'Happy', 'with', 'the',
'process', '.', 'Better', 'to', 'do', 'twice', 'a', 'year', '.', 'We', 'can', 'hire', 'consultant', 'to', 'come', 'out', 'with',
'better', 'method', '.', 'Last', 'year', 'method', 'was', 'clearer', 'than', 'this', 'year', '.', 'Many', 'changes', 'are',
'required', 'in', 'self-assessment', 'questions', '.', 'The', 'questions', 'were', 'biased', 'toward', 'particular',
'department', '.', 'The', 'questions', 'were', 'so', 'subjective', '.', 'Difficult', 'to', 'measure', 'performance', '.', 'I',
'think', 'HR', 'department', 'should', 'research', 'more', 'on', 'appraisal', 'process', '.', 'Very', 'happy', 'with', 'the',
'way', 'process', 'was', 'carried', 'out', 'in', 'our', 'organization', '.', 'I', 'would', 'be', 'happy', 'if', 'few',
'questions', 'are', 'modified', 'during', 'next', 'appraisal', 'process', '.', 'I', 'am', 'satisfied', 'with', 'overall',
'communication', 'and', 'the', 'process', '.', 'The', 'process', 'was', 'fair', '.', 'Some', 'scope', 'to', 'improve', 'remains',
'.',                                                                                                                  ', '.',
'Ver
'que                                                                                                                  ', '.',
'Ove
```

Interpretation:

- **word_tokenize()** separates into list of words.
- Words are automatically converted to lowercase.
- Punctuations are also treated as separate tokens in word tokenization.

What Is NLP?

- Natural Language Processing (NLP) is the ability of a computer to analyze and process natural language data.
- NLP is used to extract relevant information from a piece of text which is then used for various purposes.
- NLP works on four levels - lexical, syntactic, semantic, pragmatic.

  - Lexical-pre-processing of the text, such as removal of stop words, making all text lowercase etc.

  - Syntax analysis-It analyses the 'structure' of text, 'correctness' of a sentence in terms of the grammar of the language of origin.

  - Semantic assessment-attempts to study the 'meaning' of the text.

  - Pragmatic -the analysis is aimed at deciphering the 'intended' meaning of the text.

Applications of NLP

Some of the most notable applications of NLP are:

- **Search algorithms** - when you search for “What is the population of India”, the top result shows the actual answer.
- **General websites** - Pop-up windows on websites offering ‘chat with their representative’ These are chatbots trained to correctly answer commonly asked questions.
- **Retail** - Assessment of product feedback using text summarization and sentiment analysis; Query resolution with automated responses.
- **Personalized services** - Email apps predicting next word(s) in an email, tagging emails as important, personal etc.
- **Translation Apps**

Sentiment Analysis

- Sentiment Analysis is the **process of determining whether a piece of writing is positive, negative or neutral**.
- It's also known as opinion mining, deriving the opinion or attitude of a speaker.
- Sentiment analysis is performed using natural language processing, text analysis, computational linguistics and, sometimes, biometrics to systematically identify, extract, quantify, and study affective states and subjective information.
- Basic task in sentiment analysis is classifying the polarity of a given text at the document, sentence, or feature/aspect level—whether the expressed opinion is positive, negative, or neutral. Advanced - "beyond polarity" - sentiment classification looks at emotional states, for instance, "angry", "sad", and "happy".

Sentiment Analysis Using “TextBlob"

```
# Library for sentimental analysis
pip install textblob
```

```
# Calculate sentiment score for the overall feedback
from textblob import TextBlob
sentiment_analysis = list()
```

TextBlob() loads text and calculates the score of each sentence on the basis of the presence of words having positive or negative sentiment and presence of negation.

```
# Display the summary of sentiment score of all the documents
import re
sentences = re.split(r' *[\.\?!][\'"\)\]]* *', data)
for i in range(0,len(sentences)):
    sentiment = TextBlob(sentences[i])
    print("Sentiment Score: ", sentiment.sentiment.polarity)
    sentiment_analysis.append(sentiment)
```

sentiment.sentiment.polarity gives probability of sentiment analysis. Polarity is in float which lies in the range of [-1,1] where 1 indicates positive statement and -1 negative.

Sentiment Analysis Using “TextBlob"

```
# Output
Sentiment Score:  0.0
Sentiment Score:  0.5
Sentiment Score:  0.21111111111111114
Sentiment Score:  0.35
Sentiment Score:  -0.65
Sentiment Score:  0.0
Sentiment Score:  0.06500000000000003
Sentiment Score:  0.3333333333333333
Sentiment Score:  1.0
Sentiment Score:  0.7
Sentiment Score:  0.0
Sentiment Score:  -0.4
Sentiment Score:  0.2
Sentiment Score:  1.0
Sentiment Score:  0.9099999999999999
Sentiment Score:  0.0
Sentiment Score:  0.9099999999999999
Sentiment Score:  0.0
Sentiment Score:  0.3
Sentiment Score:  -0.25
Sentiment Score:  0.2
Sentiment Score:  0.0
Sentiment Score:  1.0
Sentiment Score:  0.0
Sentiment Score:  0.8
Sentiment Score:  0.0
Sentiment Score:  0.7
Sentiment Score:  0.5
Sentiment Score:  0.0
```

Interpretation:

- TextBlob sentiment score varies from -1 to 1. 0 indicates neutral, between 0 to 1 it is positive and 0 to -1 is negative.

Sentiment Analysis Using “TextBlob"

```
Sentiment Score:  -0.4
Sentiment Score:  0.2
Sentiment Score:  0.25
Sentiment Score:  0.3
Sentiment Score:  -0.4
Sentiment Score:  0.9099999999999999
Sentiment Score:  0.0
Sentiment Score:  0.7
Sentiment Score:  0.0
Sentiment Score:  -0.5
Sentiment Score:  0.2
Sentiment Score:  1.0
Sentiment Score:  0.13000000000000003
Sentiment Score:  0.8
Sentiment Score:  0.5
Sentiment Score:  0.5
Sentiment Score:  0.0
Sentiment Score:  0.5
Sentiment Score:  0.16666666666666666
Sentiment Score:  0.0
Sentiment Score:  -0.5
Sentiment Score:  0.5
Sentiment Score:  1.0
Sentiment Score:  0.20000000000000004
Sentiment Score:  0.25
Sentiment Score:  0.7
Sentiment Score:  0.0
Sentiment Score:  0.7
Sentiment Score:  0.0
Sentiment Score:  1.0
Sentiment Score:  1.0
Sentiment Score:  0.08333333333333333
Sentiment Score:  0.0
```

Sentiment Analysis Using “vader"

- **VADER (Valence Aware Dictionary and sEntiment Reasoner)** is a lexicon and rule-based sentiment analysis tool in python. It is used mostly for analyzing sentiments on social media.
- In this approach, each of the words in the lexicon are rated as to whether it is positive or negative, and in many cases, how positive or negative.
- While analysing, **VADER checks the text data it finds if any of the words are present in the lexicon.** For example, in the sentence “Happy with the process, but salary increment in 2019 is very low as compared to previous years.” has two words in the lexicon (**happy and low**).
- VADER produces **four sentiment categories**. The first three categories positive, negative, neutral gives proportion of the same. The compound score gives the sum of all the lexicon ratings which are standardised between -1 and 1.

Sentiment Analysis Using “vader"

```
# Download “vader_lexicon” from “nltk”
nltk.download('vader_lexicon')
```

```
# Calculate Sentiment Values
from nltk.sentiment.vader import SentimentIntensityAnalyzer
import pandas as pd
sent_analysis = pd.DataFrame(columns =
['sentence','compound','negative', 'neutral','positive'])
sid = SentimentIntensityAnalyzer()
for i in range(0,len(text)):
    ss =  sid.polarity_scores(text[i])
    compound = ss['compound']
    negative = ss['neg']
    neutral = ss['neu']
    positive = ss['pos']
    sent_analysis = sent_analysis.append({"sentence": text[i],
"compound": compound,"negative":  negative,"neutral":
neutral,"positive":  positive}, ignore_index=True)
sent_analysis.head(10)
```

- **SentimentIntensityAnalyzer()** is used in sentiment analysis using Python nltk. It has four values ‘compound’, ‘neg’, ‘neu’, ‘pos’.

* Note : When imported nltk, nltk.download() will download the required libraries from NLTK for text mining. Run nltk.download() only for the first time

Sentiment Analysis Using “vader"

	sentence	compound	negative	neutral	positive
0	The process was transparent.	0.0000	0.000	1.000	0.000
1	There is a lot of scope to improve the process...	0.4404	0.000	0.818	0.182
2	Happy with the process, but salary increment i...	-0.1875	0.151	0.734	0.115
3	Many questions were very subjective. Very diff...	-0.4690	0.234	0.766	0.000
4	Questions could have been specific to function...	0.0000	0.000	1.000	0.000
5	More research is required to come out with bet...	0.4404	0.000	0.791	0.209
6	Very happy with the process adopted. Fair and ...	0.7425	0.000	0.527	0.473
7	Salary increment is extremely low as compared ...	-0.3384	0.210	0.790	0.000
8	Not happy with rating methodology. Very subjec...	-0.4585	0.300	0.700	0.000
9	Excellent effort by HR team. Very fair process.	0.7425	0.000	0.488	0.512

Interpretation:

- Negative compound score indicates negative sentiments. Compound score gives the sum of all the lexicons standardized between -1 and 1.

Quick Recap

In this session, we continued to learn about **Text Mining & NLP in Python** :

Tokenization

- Tokenization is the process of tokenizing or splitting a string, text into a list of tokens.
- **`sent_tokenize()`** tokenizes text data into sentences or sentence tokens.
- **`word_tokenize()`** tokenizes text data into word tokens.

NLP

- Natural Language Processing (NLP) is the ability of a computer to analyze and process natural language data.

Sentiment Analysis

- Sentiment Analysis is the process of determining whether a piece of writing is positive, negative or neutral.
- Download '**vader_lexicon**' from NLTK for sentiment analysis.

Text Mining

using Twitter Data

Contents

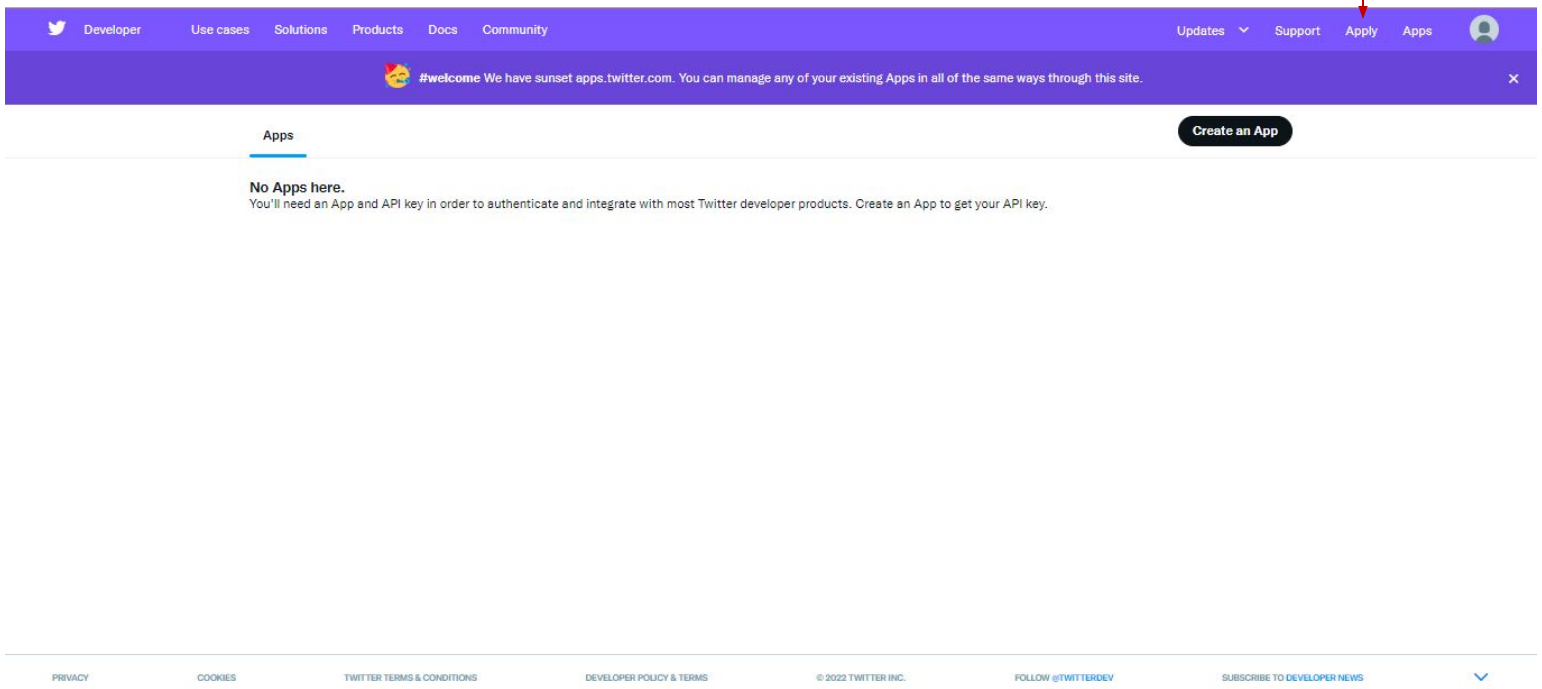
1. Why is Twitter Data Useful
2. Creating Twitter developer account
3. Creating App
4. Text mining in Python
    - Authentication of Twitter Account in Python
    - Fetching data from Twitter using ‘tweepy’ library
    - Cleaning data for text analysis
    - Generating WordCloud
    - Sentiment Analysis

Why is Twitter Data Useful

- Text mining is getting a lot attention in last few years, due to an exponential increase in digital text data from web pages and social media services.
- Twitter data constitutes a rich source that can be used for capturing information about any topic. This data can be used for finding trends related to a specific keyword, measuring brand sentiment, and gathering feedback about new products and services etc.

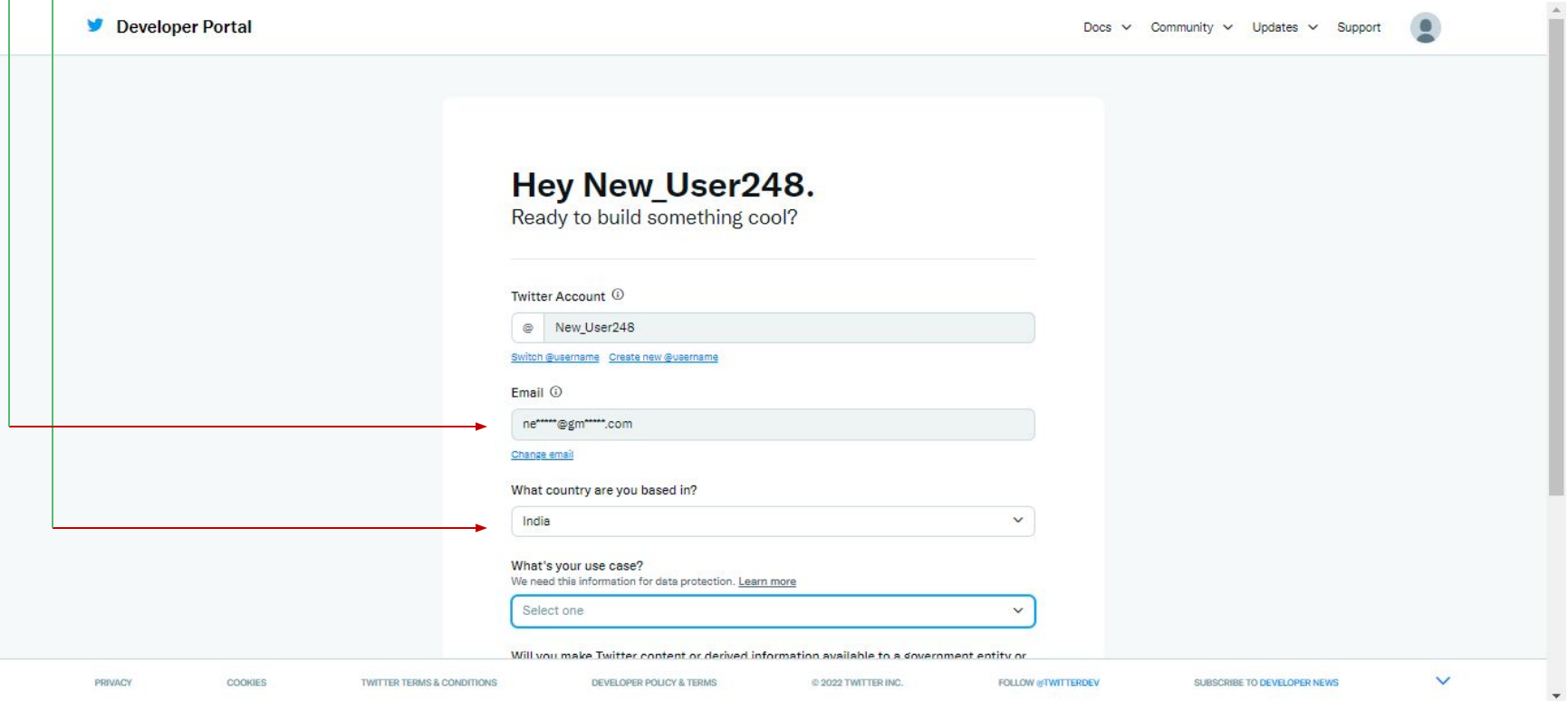
Creating Twitter Developer Account

- Twitter account is required to fetch the data from it. If not, first create a Twitter account.
- Keep the Twitter account running on one webpage.
- Open another webpage of **apps.twitter.com**
- Click on the **apply**.



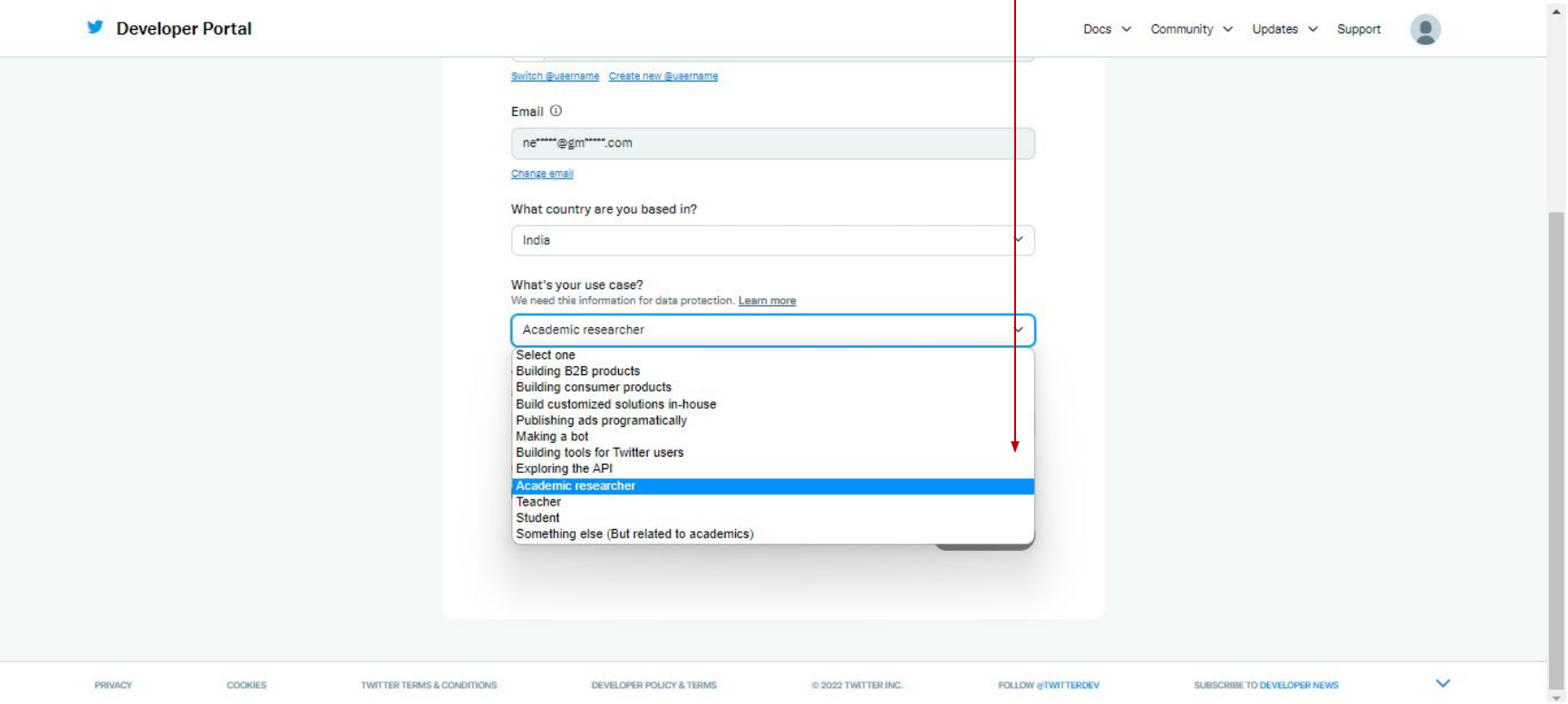
Creating Twitter Developer Account

- Fill in your verified email.
- Select the **Country.**
- If you already have developer account, you can go to slide number 24.



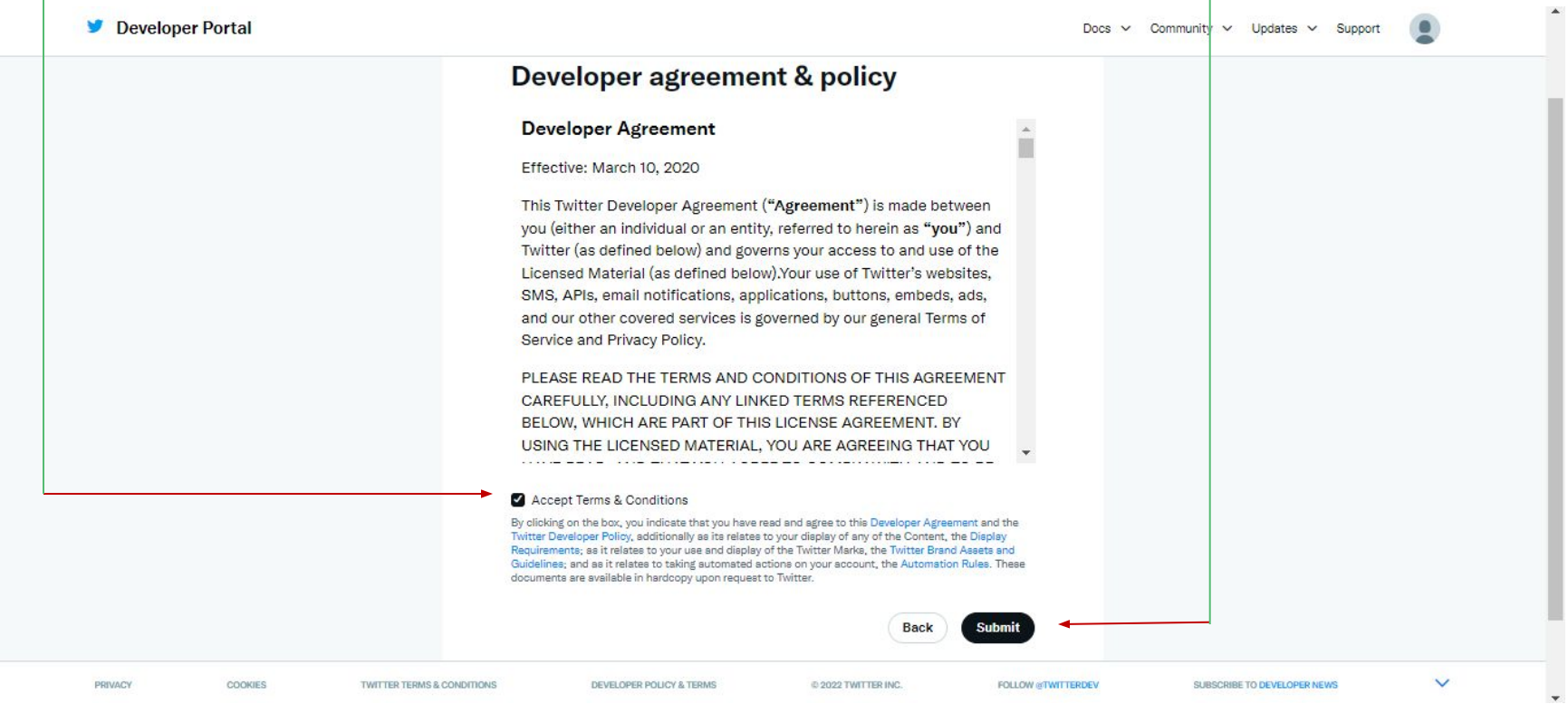
Creating Twitter Developer Account

- Click on Academic Researcher.



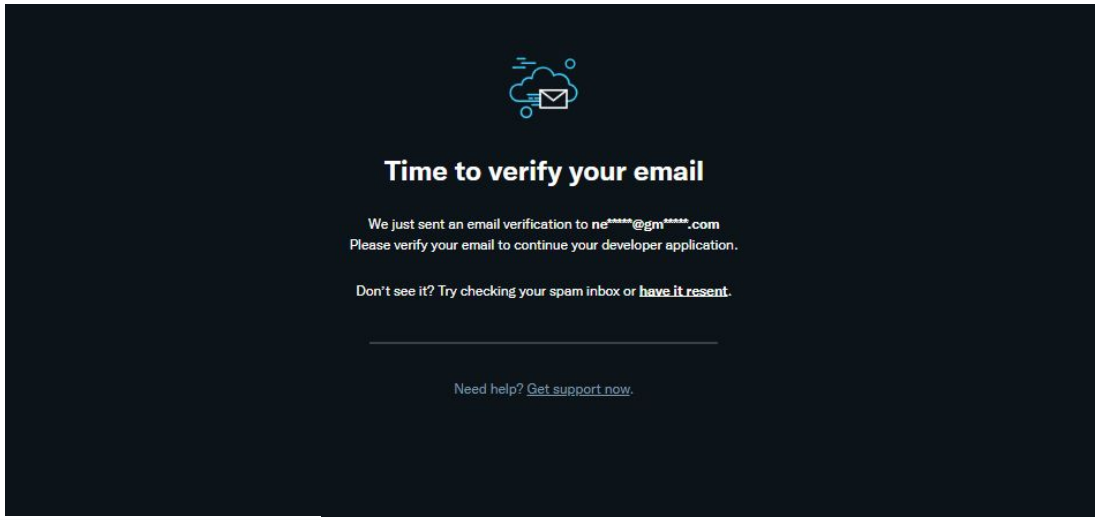
Creating Twitter Developer Account

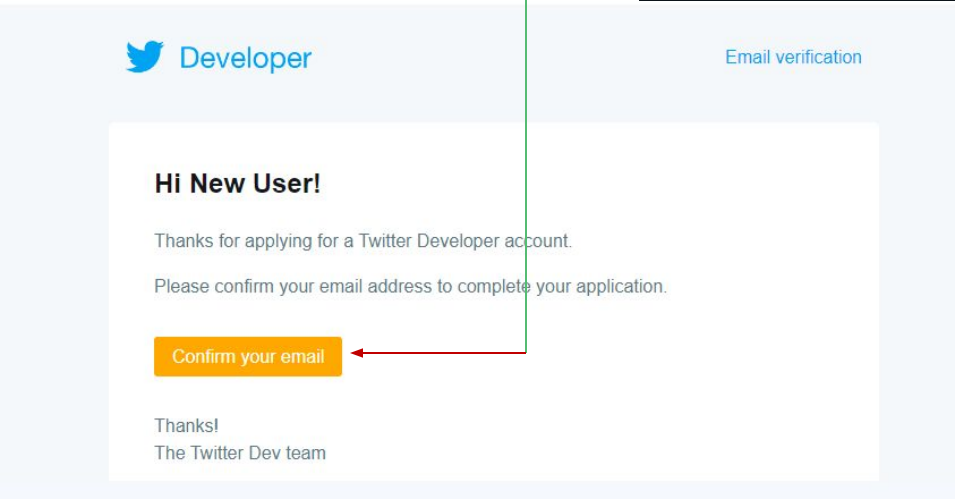
- Review and accept the developer agreement.
- Scroll down and tick the box. Click on **Submit.**



Creating Twitter Developer Account

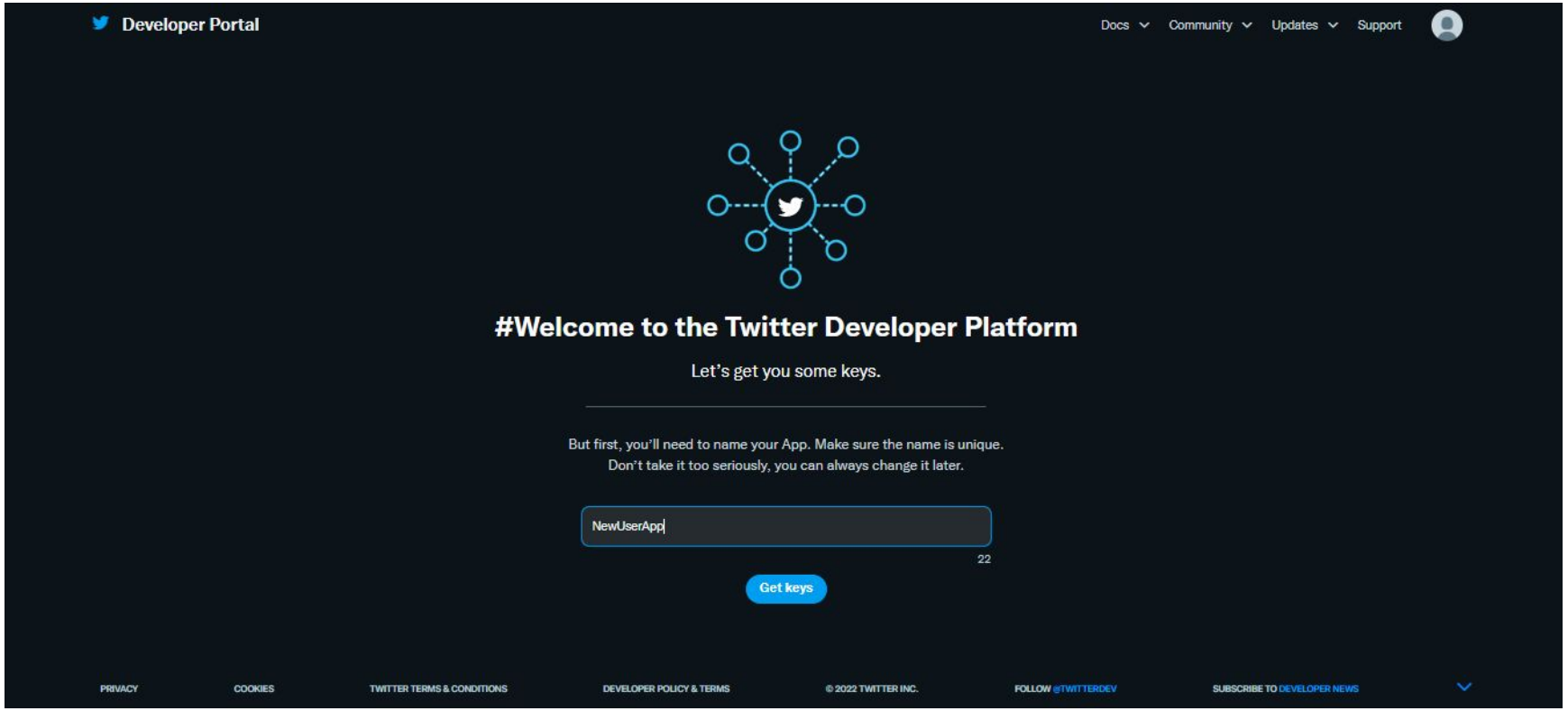
- Once this page is opened, open the email received to registered email id and click on the **confirm your email.**





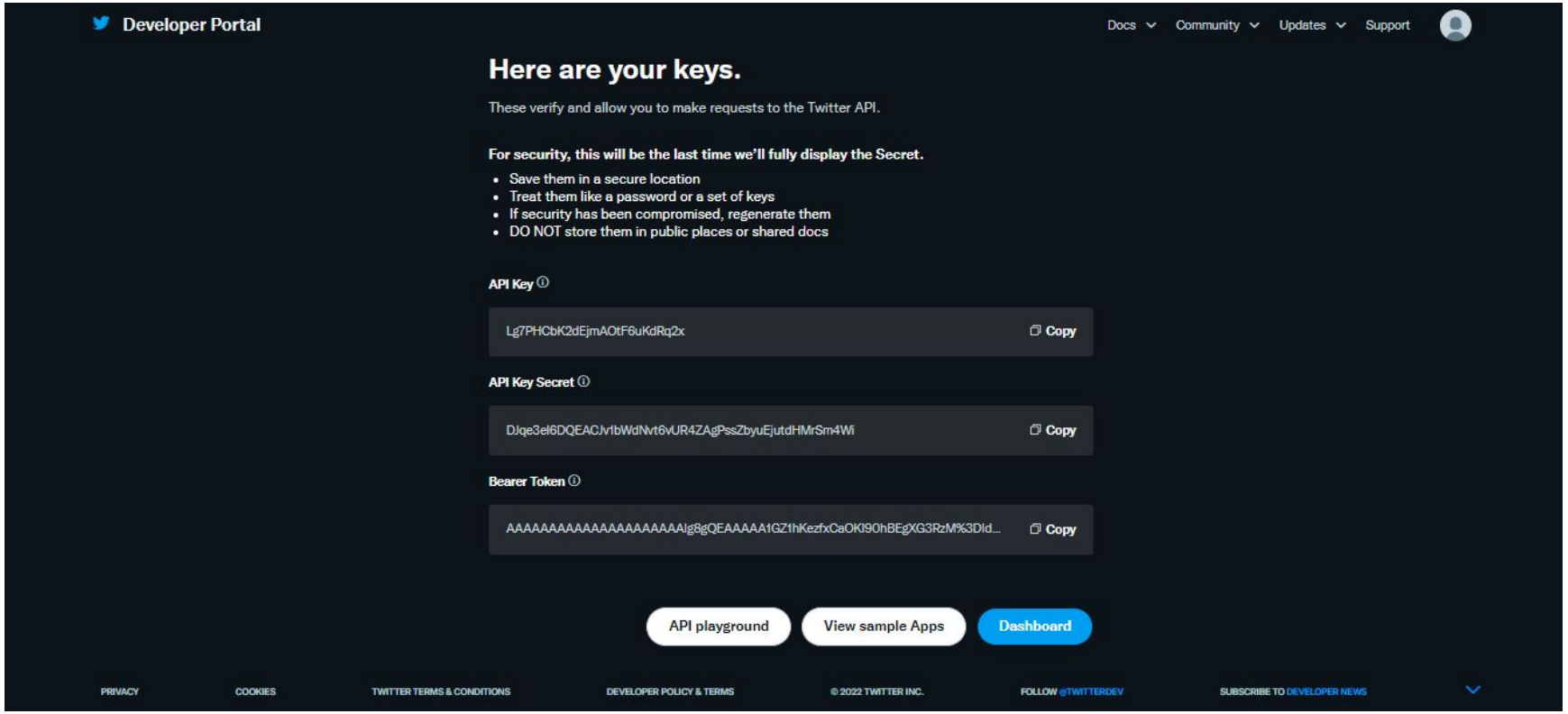
Creating App

- Once you click on the **confirm your email,** it will direct to this page.
- Give an **App Name**. (It should be UNIQUE for each developer).
- Once done, click on **Get Keys**.



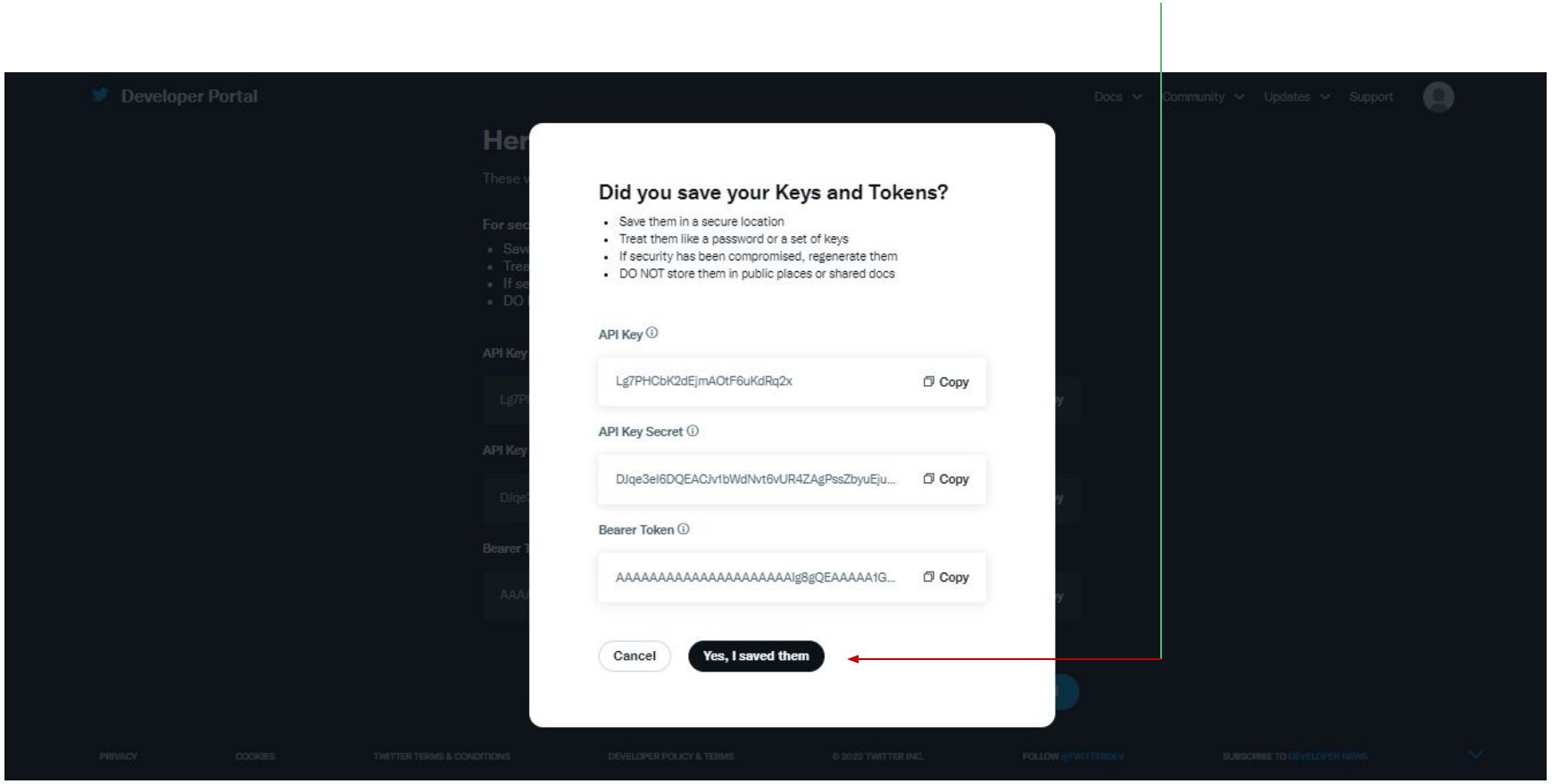
Creating App

- Save the **API Key and API Secret Key** into a notepad/external file as they are required for further analysis and will not be available later.
- Once copied, click on **Dashboard**.



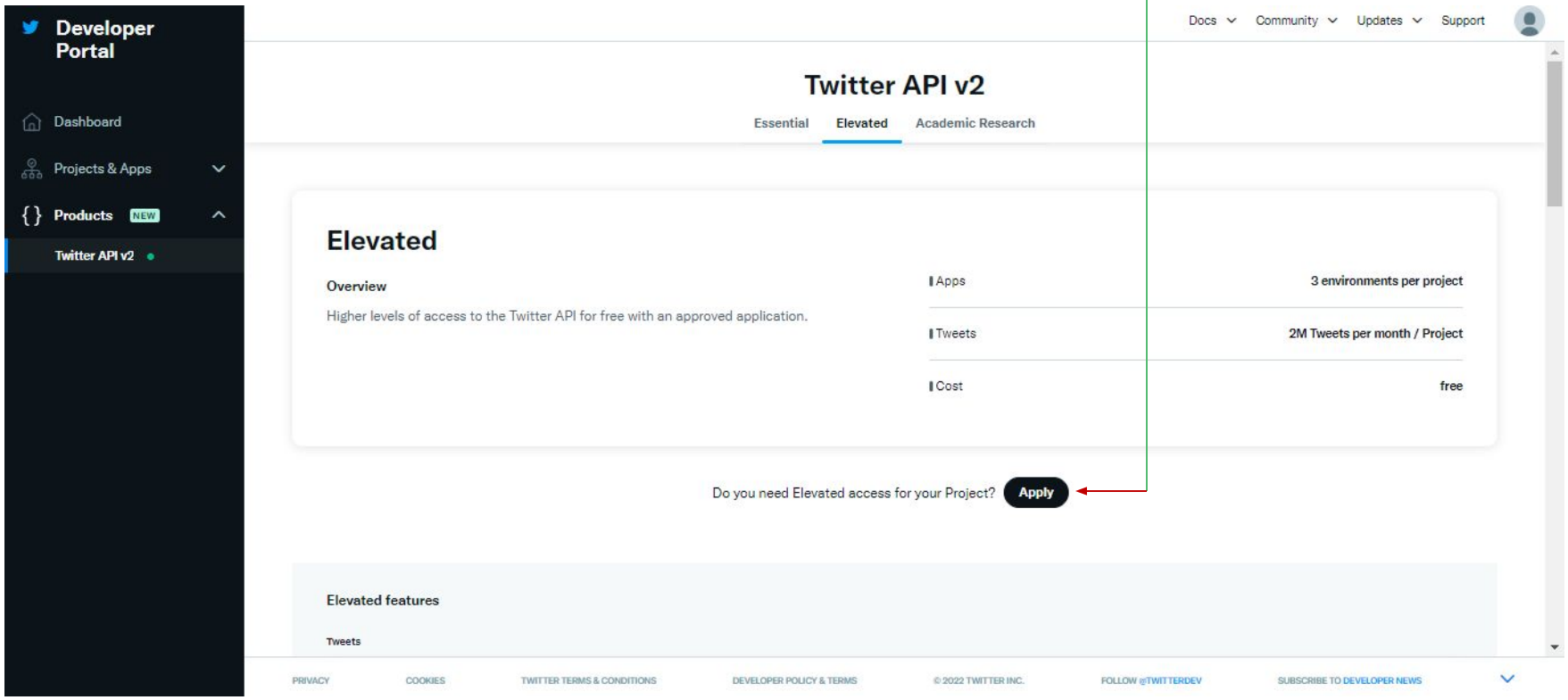
Creating App

- It will load a popup form to confirm. Click on **Yes, I saved them**.



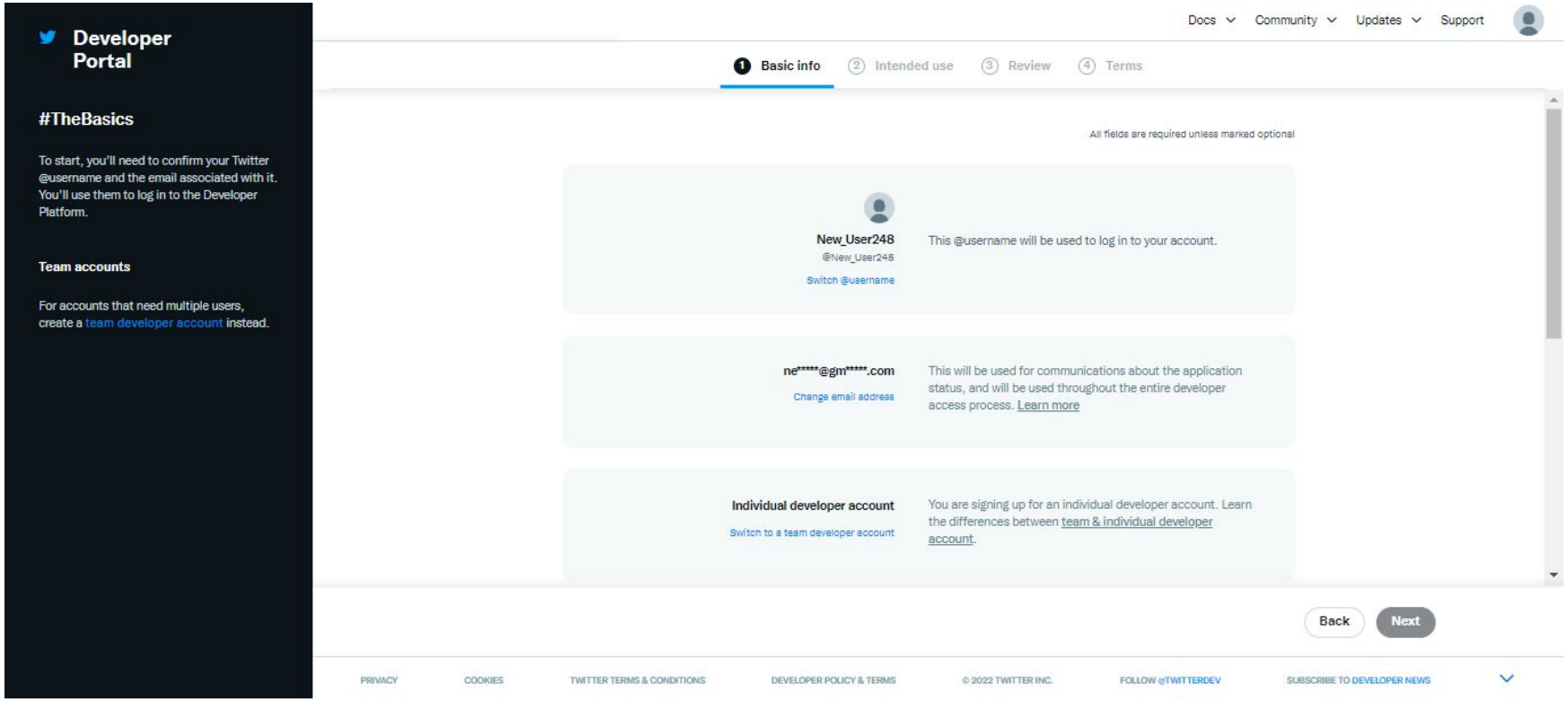
Elevated Access

- Now that the app is created, essential access is granted. But in order to extract tweets, elevated access is required.
- From the side menu, go to Products > Twitter API v2.
- Switch to Elevated tab and click Apply.



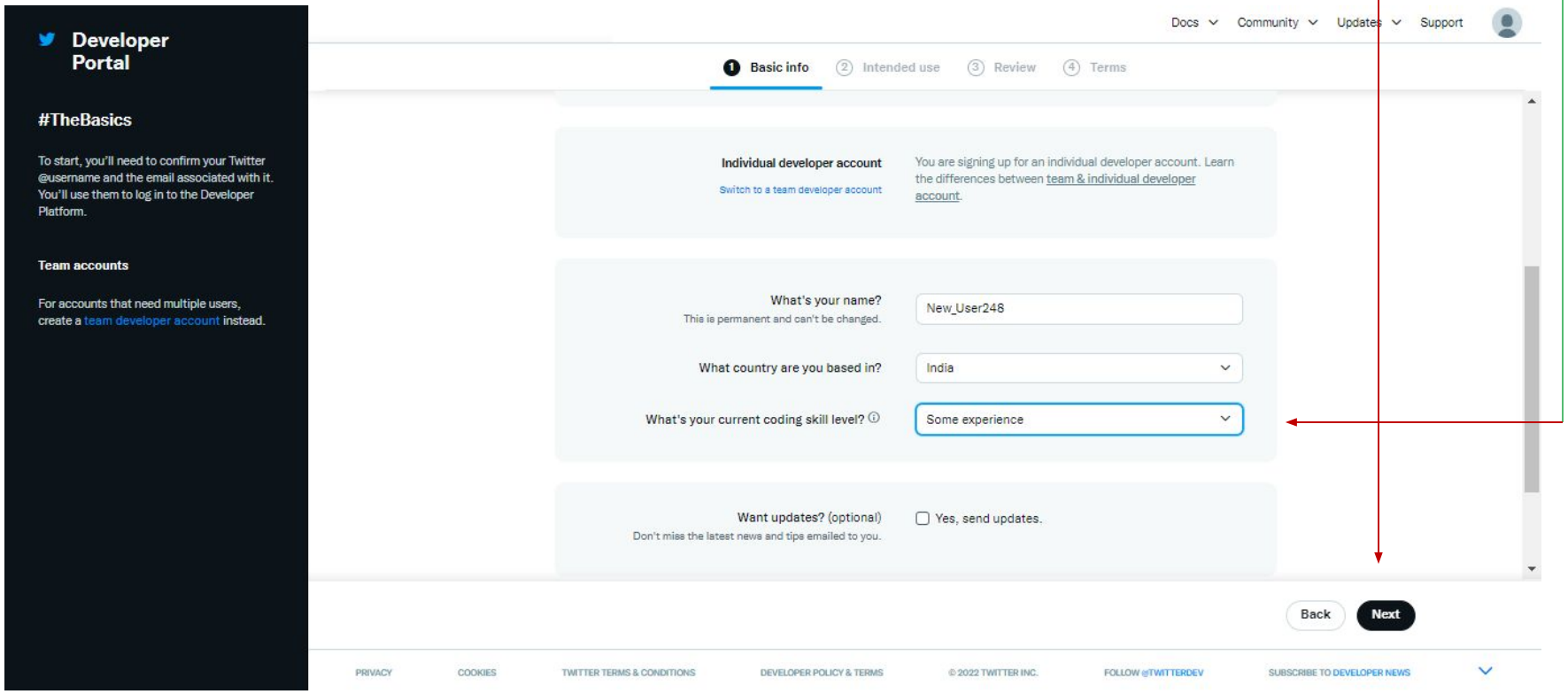
Elevated Access

- Double check the **username and email id** and if correct, proceed to the rest of the form.



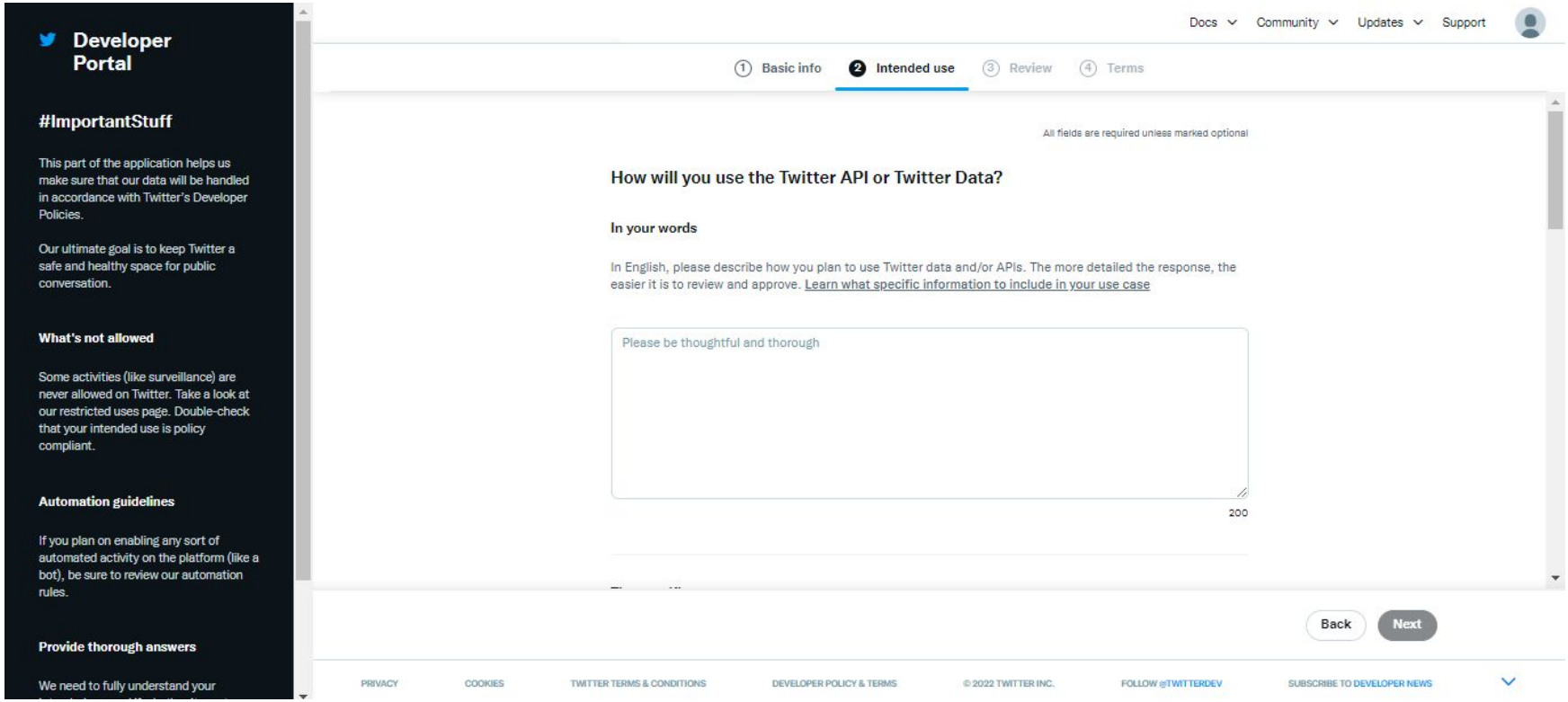
Elevated Access

- Double check your name and country.
- Select **Some experience** for the question **what's your current coding skill level?**
- Once all fields are verified, click **Next.**



Elevated Access

- Write the description for each question under **How will you use the Twitter API or Twitter data? Minimum number of characters for answers are specified for each question.**
- You can take help from this link: https://www.jcchouinard.com/apply-for-a-twitter-developer-account/



Elevated Access

Docs Communi

1 Basic info 2 Intended use 3 Review 4 Terms

The specifics

Please answer each of the following with as much detail and accuracy as possible. Failure to do so could result in delays to your access to Twitter developer platform or rejected applications. Need help? Get support now.

Are you planning to analyze Twitter data? Yes

Please describe how you will analyze Twitter data including any analysis of Tweets or Twitter users.

Please be thoughtful and thorough

100

Will your App use Tweet, Retweet, Like, Follow, or Direct Message functionality? Yes

TWITTER TERMS & CONDITIONS DEVELOPER POLICY & TERMS © 2022 TWITTER INC. FOLLOW @TWITTERDEV SUB

Docs Community

1 Basic info 2 Intended use 3 Review 4 Terms

Will your App use Tweet, Retweet, Like, Follow, or Direct Message functionality? Yes

Please describe your planned use of these features.

Please be thoughtful and thorough

100

Do you plan to display Tweets or aggregate data about Twitter content outside Twitter? Yes

Please describe how and where Tweets and/or data about Twitter content will be displayed outside of Twitter.

Please be thoughtful and thorough

OKIES TWITTER TERMS & CONDITIONS DEVELOPER POLICY & TERMS © 2022 TWITTER INC. FOLLOW @TWITTERDEV SUBSC

Developer Portal

#ImportantStuff

This part of the application helps us make sure that our data will be handled in accordance with Twitter's Developer Policies.

Our ultimate goal is to keep Twitter a safe and healthy space for public conversation.

What's not allowed

Some activities (like surveillance) are never allowed on Twitter. Take a look at our restricted uses page. Double-check that your intended use is policy compliant.

Automation guidelines

If you plan on enabling any sort of automated activity on the platform (like a bot), be sure to review our automation rules.

Provide thorough answers

We need to fully understand your

Docs Community Updates Support

1 Basic info 2 Intended use 3 Review 4 Terms

The specifics

Please answer each of the following with as much detail and accuracy as possible. Failure to do so could result in delays to your access to Twitter developer platform or rejected applications. Need help? Get support now.

Are you planning to analyze Twitter data? Yes

Please describe how you will analyze Twitter data including any analysis of Tweets or Twitter users.

Please be thoughtful and thorough

100

Will your App use Tweet, Retweet, Like, Follow, or Direct Message functionality? Yes

Back Next

PRIVACY COOKIES TWITTER TERMS & CONDITIONS DEVELOPER POLICY & TERMS © 2022 TWITTER INC. FOLLOW @TWITTERDEV SUBSCRIBE TO DEVELOPER NEWS

In case, the images in the above slide need a different layout.

Developer Portal

#ImportantStuff

This part of the application helps us make sure that our data will be handled in accordance with Twitter's Developer Policies.

Our ultimate goal is to keep Twitter a safe and healthy space for public conversation.

What's not allowed

Some activities (like surveillance) are never allowed on Twitter. Take a look at our restricted uses page. Double-check that your intended use is policy compliant.

Automation guidelines

If you plan on enabling any sort of automated activity on the platform (like a bot), be sure to review our automation rules.

Provide thorough answers

We need to fully understand your

Docs Community Updates Support

1 Basic info 2 Intended use 3 Review 4 Terms

Will your App use Tweet, Retweet, Like, Follow, or Direct Message functionality? Yes

Please describe your planned use of these features.

Please be thoughtful and thorough

100

Do you plan to display Tweets or aggregate data about Twitter content outside Twitter? Yes

Please describe how and where Tweets and/or data about Twitter content will be displayed outside of Twitter.

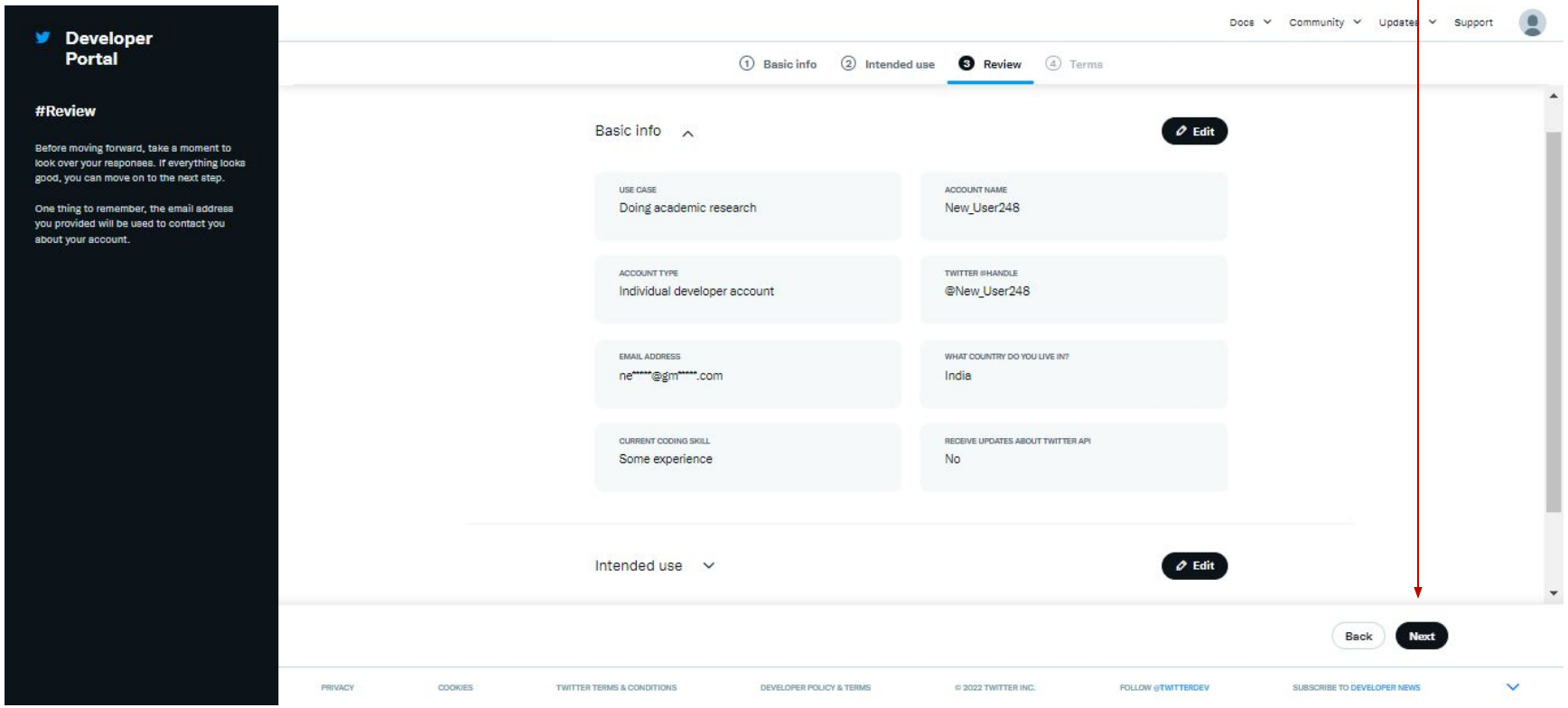
Please be thoughtful and thorough

Back Next

PRIVACY COOKIES TWITTER TERMS & CONDITIONS DEVELOPER POLICY & TERMS © 2022 TWITTER INC. FOLLOW @TWITTERDEV SUBSCRIBE TO DEVELOPER NEWS

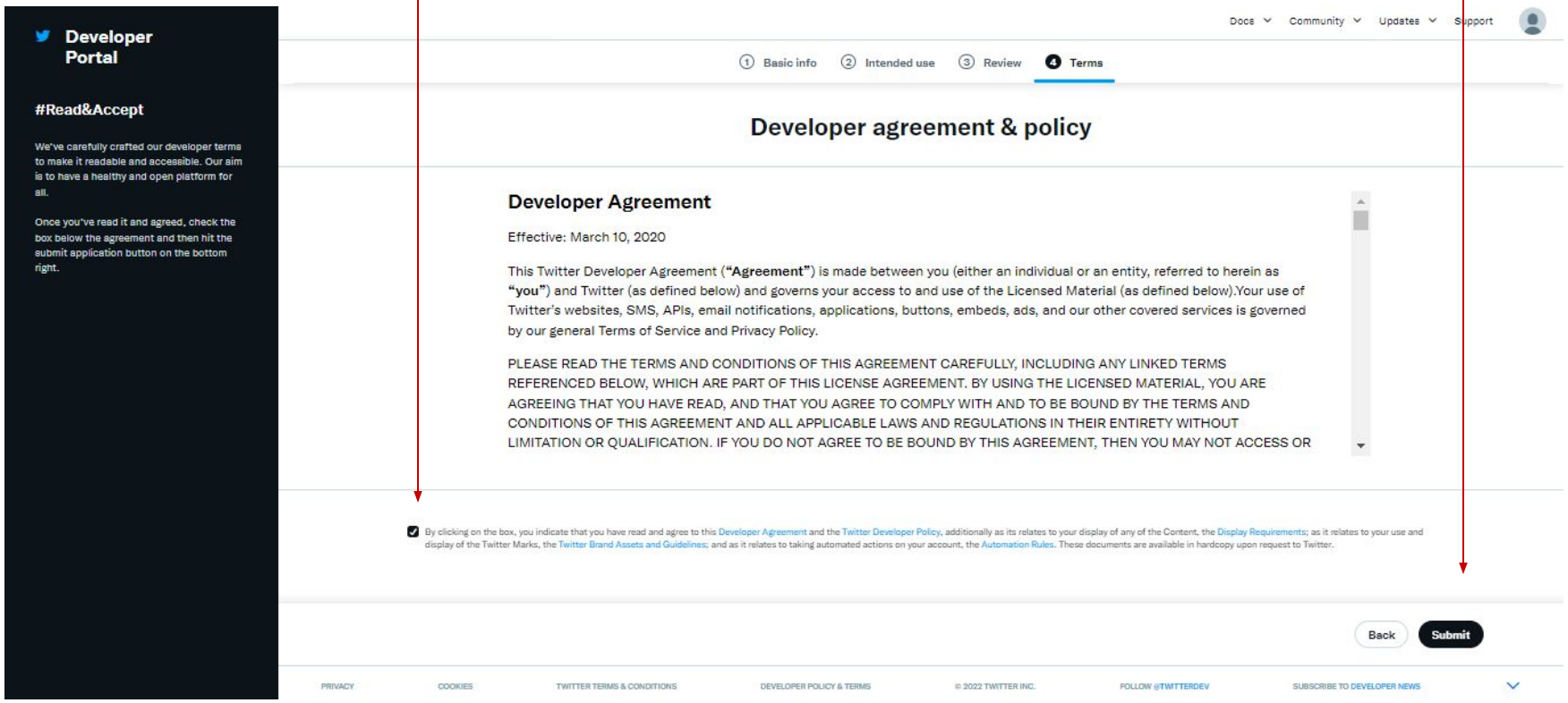
Elevated Access

- Double check the inputs and if correct, click **Next.** Else click **Edit.**



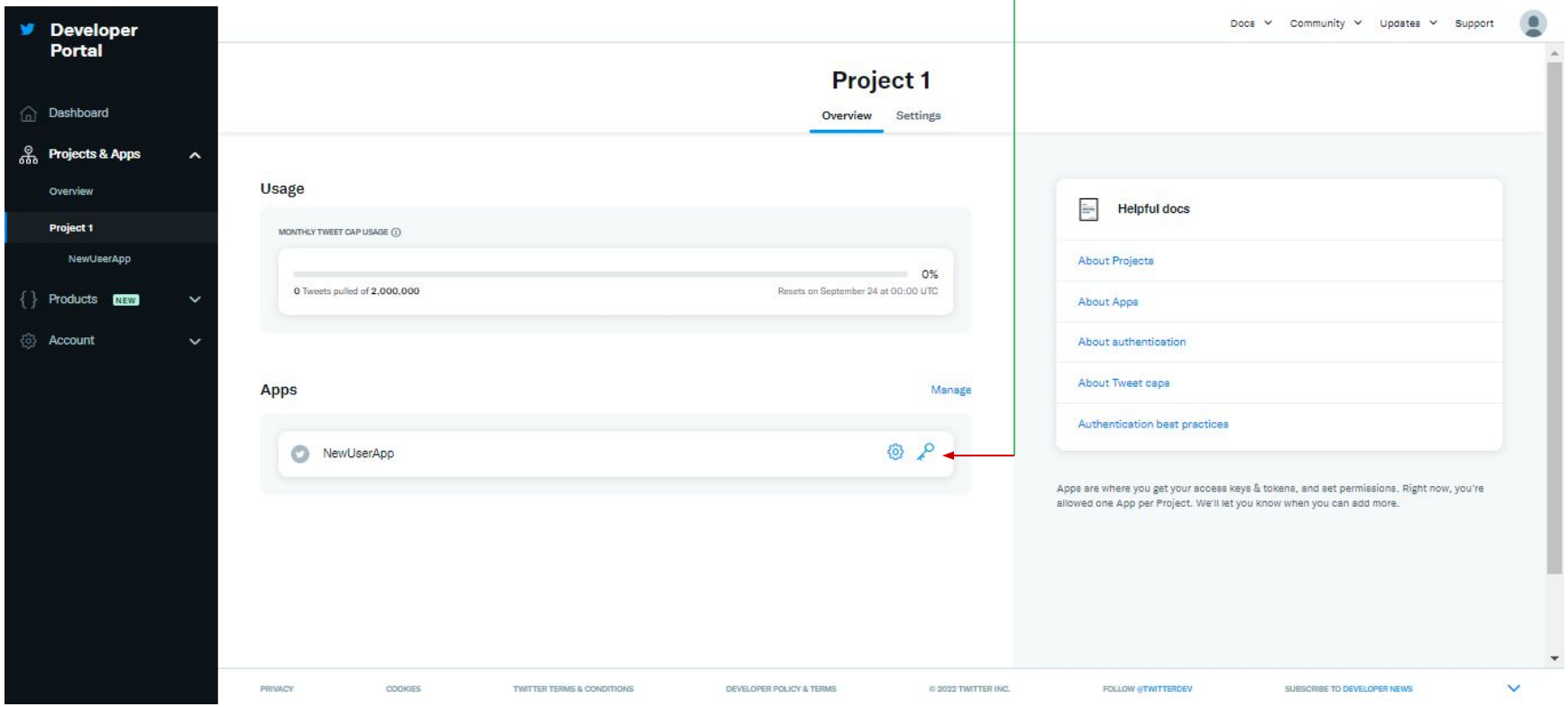
Elevated Access

- Review and accept the **Developer agreement & policy**.
- Scroll down and tick the box. Click on **Submit.**



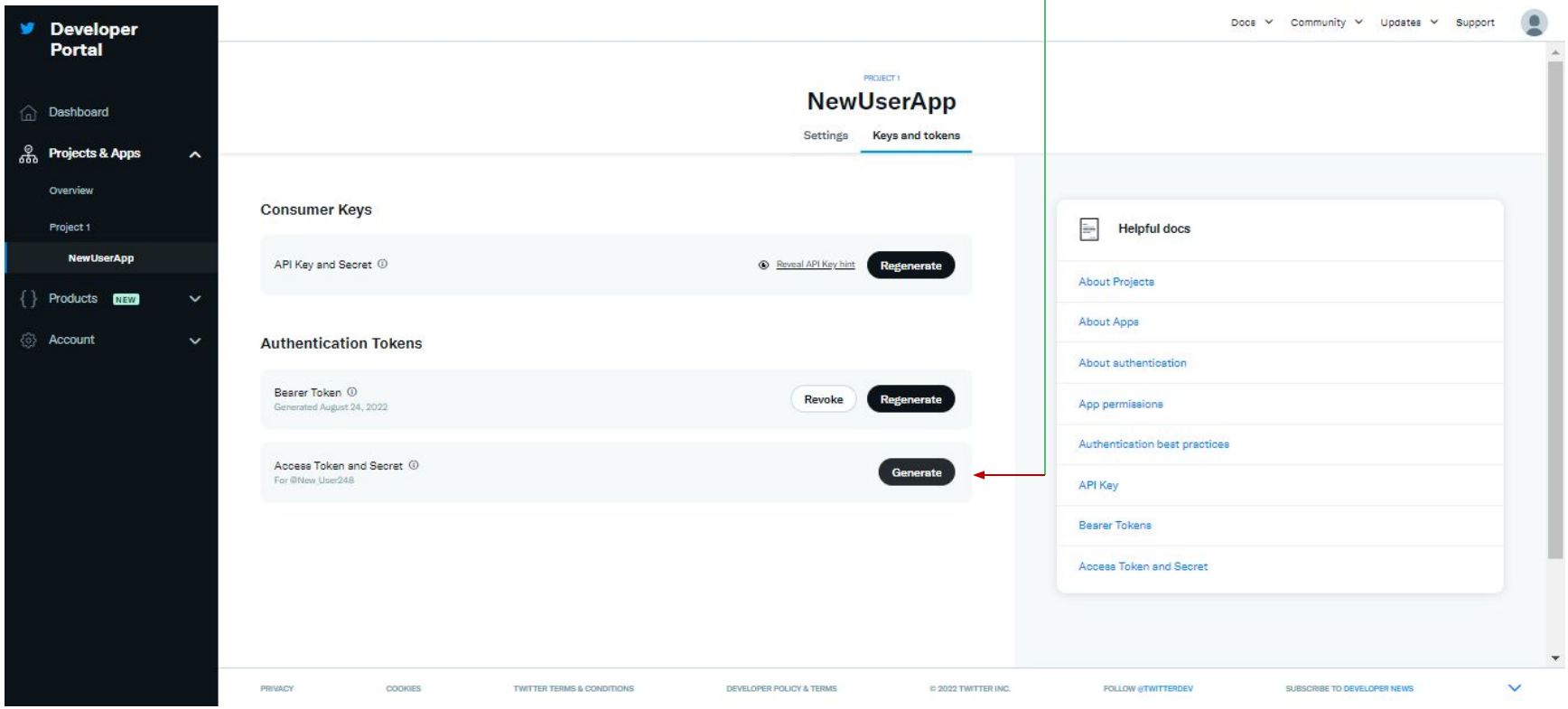
Access Key

- Lastly, generate **Access Key and Access Token Key**.
- From the side menu, go to **Projects & Apps** >**Project 1**.
- Under **Apps**, click on the **key icon**.



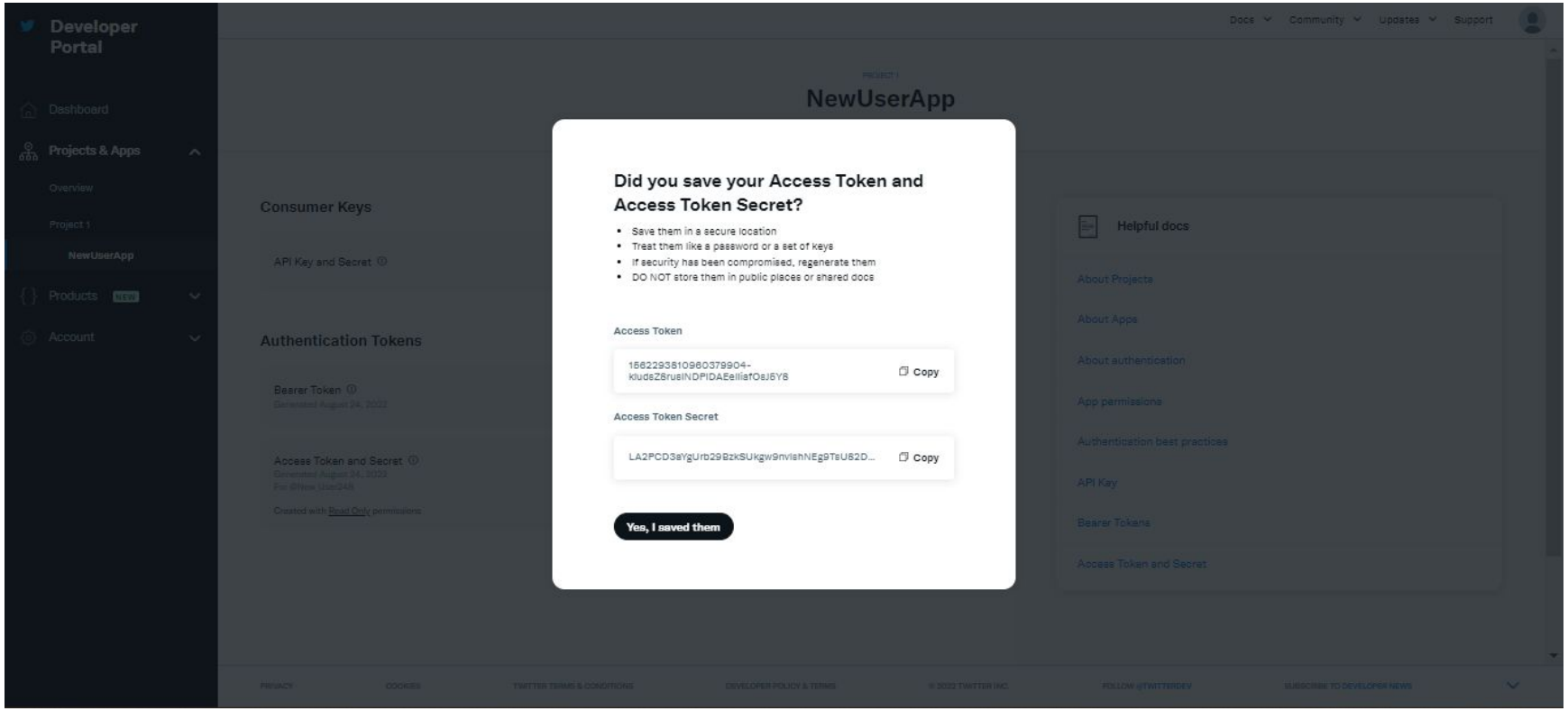
Access Key

- Click on **Generate.**



Access Key

- It will load a popup form with the **Access Key and Access Token Key**.
- Save them into a notepad/external file as they are required for further analysis and will not be available later.
- Click on **Yes, I saved them**.



Authentication of Twitter Account in Python

```
# Install “tweepy” library from Anaconda Prompt
pip install tweepy
# Import “tweepy” library in python
import tweepy
```

- ❑ **tweepy** is a Python library which provides access to the Twitter API.

- Copy the API key and API secret key from slide 11 and Access token and Access token secret from slide 23 and paste in Python code.

```
consumer_key = " paste here your API key "
consumer_secret = " paste here your API secret key "
access_token = " paste here your Access token "
access_token_secret = " paste here your Access token secret "
```

- Complete the twitter authorization process.

```
auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)
api = tweepy.API(auth)
```

- Twitter account is now connected to Python for fetching tweets.

- ❑ **OAuthHandler()** creates the authentication object.
- ❑ **set_access_token()** sets your access token and secret
- ❑ **API()** creates the API object while passing in auth information

Fetching Data From Twitter

```
# Fetch tweets for Delhi pollution.
```

```
tweets = tweepy.Cursor(api.search_tweets ,
              q='pollution + delhi',
              lang="en").items(500)
pollution_tweets=[tweet.text for tweet in tweets]
pollution_tweets[0:5]
```

- **Cursor()** function will issue a search of Twitter based on a supplied search string.
- First argument search query to issue to twitter. Use "+" to separate query terms.
- **.items()** specifies the maximum number of tweets to return
- **lang="en"** search tweets in English language.
- Twitter developer account searches tweets for maximum 7 days.

* **Note : The tweets fetched will be different depending upon when they are extracted.**

Fetching Data From Twitter

```
# Output:
['RT @AamAadmiParty: "I acknowledge that the high-level of pollution in Delhi is
mainly due to its local sources… and thus, we are conducting…',
 'T-1 \nAs the Winds started Blowing in delhi the concern about the pollution also
get blown away #DelhiPollution #DelhiGovt Shame Shame',
 '@blkahn Plus Amazon fires, Delhi pollution, China, ocean plastics...must find
solutions',
 '@Sharmistha_GK @ArvindKejriwal Pollution is not just a Delhi problem. Its a
north India problem. If you view the ma… https://t.co/gDHPbiYc2T',
 'RT @AamAadmiParty: "I acknowledge that the high-level of pollution in Delhi is
mainly due to its local sources… and thus, we are conducting…']
```

Cleaning Data For Text Analysis

```
# Clean this corpus before we make the word cloud.
# Convert all data to lowercase
pollution_tweets2 = [item.lower() for item in pollution_tweets]
# Remove twitter handles
import re
pollution_tweets2 = [re.sub('@[^\s]+','',item) for item in
pollution_tweets2]
# Remove twitter hyperlinks
pollution_tweets2 = [re.sub('(http\S+)','',item) for item in
pollution_tweets2]
# Remove Punctuation
from string import punctuation
remove_punc = str.maketrans("","", punctuation)
pollution_tweets2 = [item.translate(remove_punc) for item in
pollution_tweets2]
# Remove Numbers
from string import digits
remove_digits = str.maketrans('', '', digits)
pollution_tweets2 = [item.translate(remove_digits) for item in
pollution_tweets2]
```

Cleaning Data For Text Analysis

```
# Remove stopwords
```

- **set().union(stopwords.words('english'), additional)** creates a set of all the english stopwords which enables us to use that set entirely to remove stop words from the data.

```
import nltk
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
additional  = ['rt','rts','retweet']
swords = set().union(stopwords.words('english'),additional)
pc=[]
for item in pollution_tweets2:
    word_tokens = word_tokenize(item)
    pol_clean = [w for w in word_tokens if not w in swords]
    pc.append(pol_clean)
pc[2]
```

```
# Remove all white space created due to above cleaning
```

```
rm_ws=[]
for item in pc:
    remove_whitespace = [x.strip() for x in item]
    rm_ws.append(remove_whitespace)
```

```
# Remove search words that are more frequent
```

```
import itertools
combined = list(itertools.chain.from_iterable(rm_ws))
remove_common = [w for w in combined if not w in "pollution+delhi"]
```

Generating WordCloud

```
# Generate wordcloud of clean data
from wordcloud import WordCloud
wordcloud =
WordCloud(background_color="white").generate(str(remove_common))
import matplotlib.pyplot as plt
# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud); plt.axis("off")
plt.tight_layout(pad = 0); plt.show()
```

Output

Sentiment Analysis Using ‘vader’

```
# Import “SentimentIntensityAnalyzer” from “vader”
# Import “pandas”
```

```
from nltk.sentiment.vader import SentimentIntensityAnalyzer
import pandas as pd
```

```
# Perform Sentiment analysis
```

```
sent_analysis = pd.DataFrame(columns =
['sentence','compound','negative', 'neutral','positive'])
sid = SentimentIntensityAnalyzer()
for i in range(0,len(pollution_tweets2)):
    ss =  sid.polarity_scores(pollution_tweets2[i])
    compound = ss['compound']
    negative = ss['neg']
    neutral = ss['neu']
    positive = ss['pos']
    sent_analysis = sent_analysis.append({"sentence":
pollution_tweets2[i], "compound": compound,"negative":
negative,"neutral": neutral,"positive":  positive},
ignore_index=True)
sent_analysis
```

Sentiment Analysis Using “vader"

Output :

	sentence	compound	negative	neutral	positive
0	California, even with the fires, doesn't compare to Singapore's worst day and is a tiny fraction of Delhi's best da… https://t.co/KeogdwDVtG	0.0258	0.156	0.684	0.16
1	RT @ndtv: Devotees stand knee-deep in toxic foam in Delhi's Yamuna for #ChhathPuja. https://t.co/tiHnluBdNz https://t.co/D8r6sVVpI5	0.128	0	0.903	0.097
2	RT @PopovichN: We visualized microscopic air pollution that wreaks havoc on human health. Compare your city's air quality to some of the wo…	-0.5994	0.151	0.849	0
3	RT @PopovichN: We visualized microscopic air pollution that wreaks havoc on human health. Compare your city's air quality to some of the wo…	-0.5994	0.151	0.849	0
4	RT @KailasK86985883: @BBMP_MAYOR We are not far from Delhi in Air Pollution!! Time is NOW to take right decision. Please stop the proposals…	0.1739	0.084	0.805	0.111
5	RT @nytclimate: We visualized the damaging, tiny particles that wreak havoc on human health. From the Bay Area to New Delhi, see how the wo…	-0.802	0.238	0.762	0
6	RT @nytclimate: We visualized the damaging, tiny particles that wreak havoc on human health. From the Bay Area to New Delhi, see how the wo…	-0.802	0.238	0.762	0
7	RT @nytclimate: We visualized the damaging, tiny particles that wreak havoc on human health. From the Bay Area to New Delhi, see how the wo…	-0.802	0.238	0.762	0
8	RT @EuroGeosciences: See how the world’s most polluted air compares with your city’s: @nytclimate has visualized the damaging, tiny partic…	-0.7645	0.268	0.732	0
9	RT @nytimes: Particulate pollution in the air soared last month in New Delhi. But the city struggles with air quality throughout the year:…	-0.3612	0.102	0.898	0
10	RT @ismaelnafria: Gran interactivo del @nytimes. Perfecto ejemplo de info útil y personalizada - See How the World’s Most Polluted Air Comp…	-0.2484	0.14	0.763	0.097

Interpretation:

- Negative compound score indicates, negative sentiments. Compound score give sum of all the lexicons standardized between -1 and 1.

Quick Recap

Twitter Data	• Twitter data constitutes a rich source that can be used for capturing information about any topic.
Libraries in Python	• Install libraries **'tweepy'** and **'nltk'**
Steps required before fetching data from Twitter	• Create twitter account. • Create developer account. • Create app. • Authentication of Twitter account in Python
Function in Python	• **`tweepy.Cursor()`** function extracts data from Twitter.