



Course: Data Analysis

Subject: Full Report

Group number: 1

Name	SID
Omar Amgad Abouellil	441018030
Mohammed Sami Soqati	441010207

Report Appendix

Subject	Page
Cover page-----	1
Appendix-----	2 - 3
Data collection-----	4 – 9
• Objective-----	5
• What we did-----	6
• Using SNScrape-----	7 - 8
• What we collected-----	8
• Conclusion-----	9
Data Cleaning-----	10 – 18
• Objective-----	11
• What we did-----	12
• Using Excel-----	12 - 16
• What we removed-----	16 - 17
• Conclusion-----	18
• Resources-----	18
Data Rooting(Stemming)-----	19 - 25
• Objective-----	20
• What we did-----	20 - 21
• Using Farasapy-----	22 - 23
• What we modified-----	24
• Conclusion-----	24
• Resources-----	24 - 25

Data Labeling	26 – 34
• Objective	27
• What we did	27 - 28
• Using Excel	29 - 30
• What we accomplished	31 - 33
• Conclusion	33
• Resources	34
Machine Learning in Data Classification	35 – 50
• Objective	36
• What we did	36 - 37
• Using Python (Scikit-learn)	37 - 47
• What we accomplished	47 - 48
• Conclusion	48 - 49
• Resources	49
Excel Files	50






Data collection

Objective

In this assignment we are required to collect a massive amount of data from a social networking app so that we could implement and use data analysis methods on the collected data.

- Data to be collected: 200,000 tweets from Twitter.
- Data is about: hajj and umrah.
- Data is used for: improving future hajj and umrah events in the following years in many districts.

Before we talk about the process, we must introduce the main applications we used and why we used them:

 Twitter	<p>Twitter is a social networking service where users can express their opinion using messages known as “tweets” that include text, images and/or videos and are mostly classified by such users using “hashtags(#)” so that it fits into a certain topic.</p> <p>We used twitter as a source cause it includes a large number of active users and twitter in nature is business-based and has proved quite useful in data analysis for companies that use information as a source of income or for optimizing their user’s experience.</p>
 Python	<p>Python is a high-level programming language that supports many data paradigms and has a big library that can be used in programming while also supporting many packages that other programmers have made that can be utilized for data collection and inputting data in CSV file format.</p>
 SNScrape	<p>SNScrape or Social Network Scrape is a social networking service scraper in Python that is used as a package inside python.</p> <p>You can scrape any type of data from tweets to followers and any info related to that certain info like like and retweet count, date of tweet, and the user who posted that tweet.</p>

What we did:

At first, we read a few articles to get the main idea of data collection & analysis, which helped us understand a bit more about the concept we're studying like

- What data collection is and what's the purpose of it
- The difference between first-party, second-party, third-party data collection.
- What is Quantitative and Qualitative data collection and the difference between them.
- Some common data collection methods like social media monitoring which is what we're using in today's report.

Now we had to search for tools that can help with data collection so we started looking up for some tools and we found a tool called **TAGS** which is a web based tool for getting tweets using certain hashtags but we were limited in the number of tweets that we can collect and the lack of features like searching using keywords instead of hashtags and last but not least that it required permission from twitter using a token/key(also the api is no longer verified by twitter) so instead we kept searching.

At last we found **SNScrape** which is a Python programming language package that has instant queries for data collection(scraping) from twitter and has many features, we can write a query that gets a tweet's id, content, user's username, date of tweet, number of likes and retweets and more based on certain hashtags and/or keywords while also specifying the number of tweets needed at a certain time period(ex. from 1/1/2010 until 1/1/2022).

Using SNScrape:

- **Step 1:** installing SNScrape into python

Easily done using “pip install snscape” command in command line.

- **Step 2:** importing pandas and tqdm which help input data into a CSV file and threading so we can use multithreading to get the data

```
import pandas as pd
from tqdm.notebook import tqdm
import snscape.modules.twitter as sntwitter
import threading
```

- **Step 3:** Setting up lists that contain start and end dates of hajj and an empty list to contain the threads.

```
sdate=["2013-10-12","2014-10-01","2015-09-21","2016-09-09","2017-08-29","2018-08-18","2019-08-08","2020-07-28","2021-07-17","2022-07-06"]
edate=["2013-10-17","2014-10-06","2015-09-26","2016-09-16","2017-09-03","2018-08-23","2019-08-13","2020-08-02","2021-07-22","2022-07-11"]
th=[]
```

- **Step 4:** inserting start and end dates into the **fetch_tweets** method & writing a Query and setting up a tweets list.

```
def fetch_tweets(sdate,edate,n):
    query = "(#الحج OR #الحج_المكة OR #الحج_المكة_2022 OR #الحج_المكة_2021 OR #الحج_المكة_2020 OR #الحج_المكة_2019 OR #الحج_المكة_2018 OR #الحج_المكة_2017 OR #الحج_المكة_2016 OR #الحج_المكة_2015 OR #الحج_المكة_2014 OR #الحج_المكة_2013) lang:ar since:{sdate} until:{edate} -filter:links"
    tweets = []
```

Query will include the hashtags and/or keywords and the date from when to start collecting data and until when plus any filters.

- **Step 5:** scraping tweets using a for-each loop and specifying which data is needed and the number of tweets to scrape:

```
for tweet in sntwitter.TwitterSearchScrapper(query).get_items():
    data = [
        tweet.date,
        tweet.content,
        tweet.user.username,
        tweet.likeCount,
        tweet.retweetCount
    ]
    tweets.append(data)

    if len(tweets) == 200000:
        break
```

- **Step 6:** putting list of tweets into a data frame and specifying columns for each record then inputting the data frame into a CSV file(every 200.000 tweets in file for each year & then merge them together using excel):

```
tweet_df = pd.DataFrame(  
    tweets,columns=["date","tweets","username","like_Count","retweet_Count"]  
)  
tweet_df.to_csv(f"Hajj{n}.csv" , index=False)  
print(n)
```

- **Step 7:** dividing each group of tweets into a thread so they can be collected in a synchronized matter, with the **join** used to wait for the other threads to finish.

```
for i in range(len(sdate)):  
  
    th.append(threading.Thread(target=fetch_tweets,args=(sdate[i],edate[i],i)))  
    th[i].start()  
  
for i in th:  
    i.join()
```

What we Collected:

At first, we set it up to search from 1/1/2010 until 1/1/2022 and collected 200,000 tweets but found that they weren't very precise and had only collected tweets from the past week.

So, we started looking at hajj dates for the past 10 years and used them to collect 250,000 tweets with 25,000 tweets for each year.

We decided we needed more tweets cause cleaning the data will decrease it to a much lower number so to collect data faster we made the program that collects data use multithreading so it becomes much more efficient and we collected 2.000.000(2 million) tweets excluding tweets that contain links.

- All hashtags and keywords are specified in the "Using SnScrape" section

Conclusion

Data collection & analysis is very useful in many ways, it can help us make better choices and have better judgement and overall knowledge about our product's success and our customers/beneficiaries.

The first step is collecting this data and, in this day and age it's much easier to do so there are many tools that we can utilize to collect data faster and have more resources like social networks and a user's behavior within these social networks.

Last but not least we have more awareness about how our info is being used by large corporations as well which in some ways could be a breach to privacy and unethical but it's exchanged for our usage of a product, basically we use these social networking apps and in return they use our information and sell it to third-party companies to gain more money, all and all we should be more careful about what info to share online.



Data Cleaning

Objective

In this assignment we are required to clean the data we collected in assignment 1 by removing unnecessary tweets.

We want to delete tweets that contain:

- Swear/Inappropriate words.
- Advertisements.
- Racism and differentiation between different Islamic groups and races.
- Adultery/expressions of love between users.
- Anything that's not related to the topic at all.
- Five words or less.

But before we show what we did, we must explain what data cleaning is:

- Data cleaning is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset.

What we did:

So like in assignment 1 we read about data cleaning and have learned the following:

- Data cleaning is used after collecting the data to remove unwanted and irrelevant data
- Data cleaning is different from data transforming
- Data transforming is the act of changing your data from one form to another to help analyze it.
- There are multiple ways to clean data like, removing duplicates, removing irrelevant data, fixing errors(typos and incorrect spellings),filtering and handling missing data.

Now, since our data has been collected on excel we looked for ways to remove duplicates and records based on certain keywords and have found ways that are already included in excel.

- **Resources listed below(at the end of the report)...**

Using Excel(example):

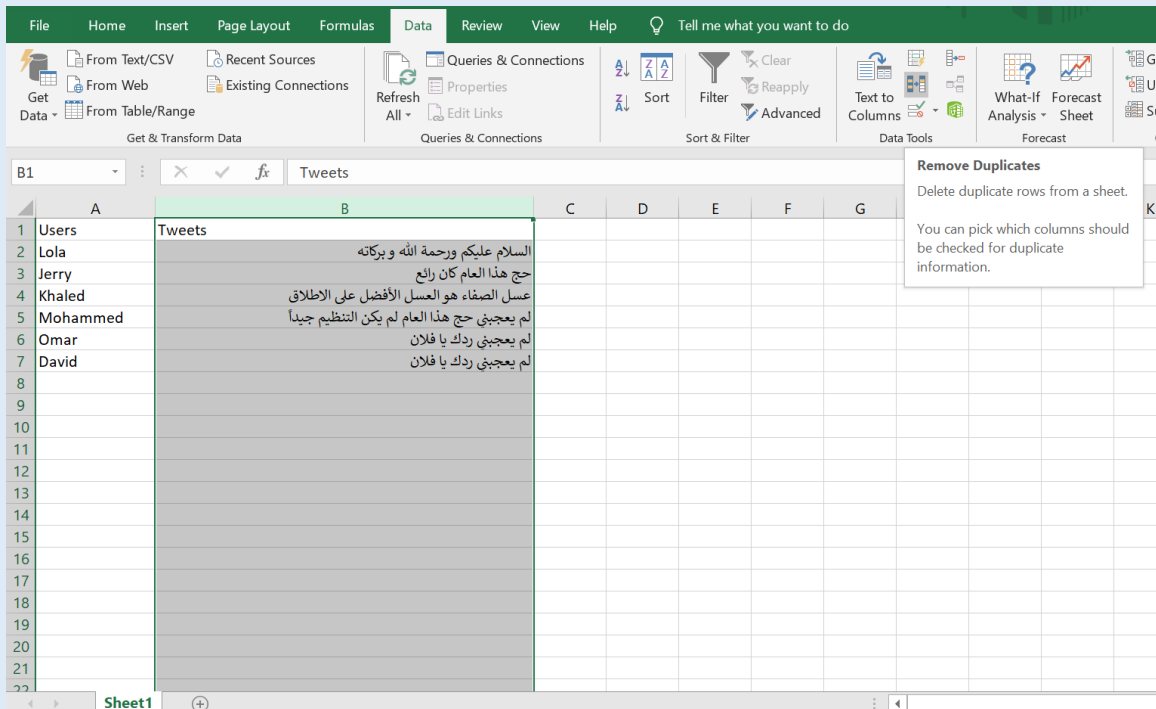
Removing duplicates:

- **Step 1:** Selecting which columns to remove from:

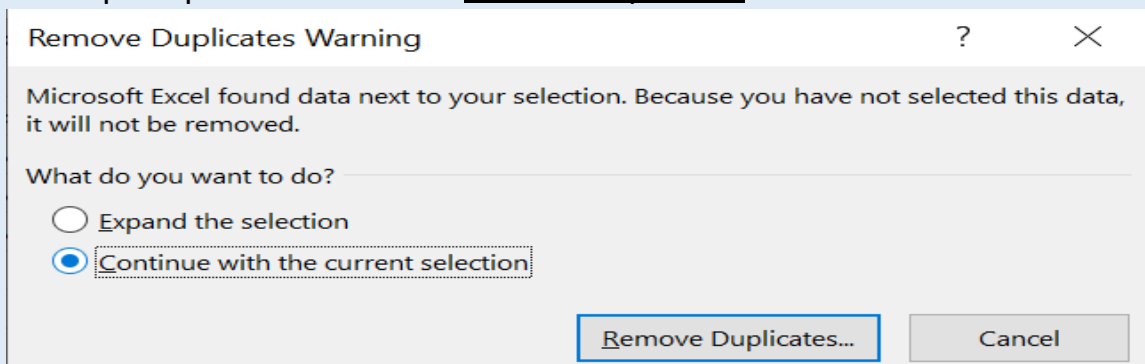
in this case we are using only the tweets column

	A	B	C
1	Users	Tweets	
2	Lola	السلام عليكم ورحمة الله وبركاته	
3	Jerry	حج هذا العام كان رائع	
4	Khaled	عسل الصفاء هو العسل الأفضل على الإطلاق	
5	Mohammed	لم يعجبني حج هذا العام لم يكن التنظيم جيداً	
6	Omar	لم يعجبني ردك يا فلان	
7	David	لم يعجبني ردك يا فلان	
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			
21			
22			

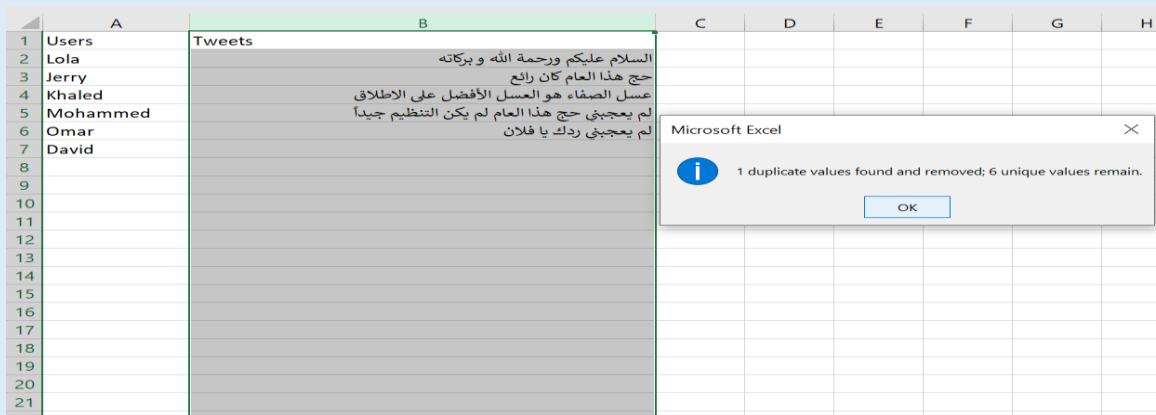
- **Step 2:** Select Data from the menu bar then click on the remove duplicates icon:



- **Step 3:** Click on “Continue with the current selection” when prompted then click on remove duplicates:



- **Step 4:** Check if data deleted duplicates:



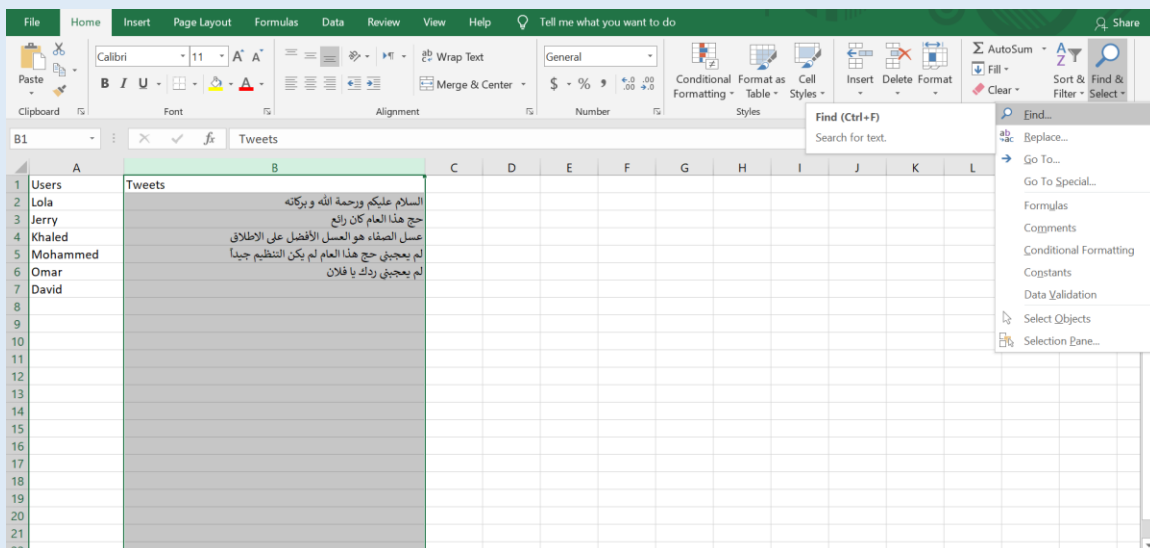
Removing based on a keyword:

- **Step 1:** Selecting which columns to remove from:

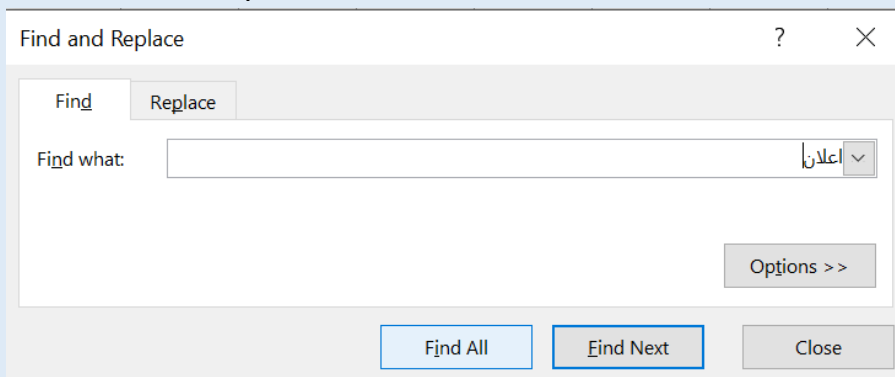
in this case we are using only the tweets column

	A	B	C
1	Users	Tweets	
2	Lola	السلام عليكم ورحمة الله وبركاته	
3	Jerry	حج هذا العام كان رائع	
4	Khaled	عسل الصفاء هو العسل الأفضل على الإطلاق	
5	Mohammed	لم يعجبني حج هذا العام لم يكن التنظيم جيداً	
6	Omar	لم يعجبني ردك يا فلان	
7	David	لم يعجبني ردك يا فلان	
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			
21			
22			

- **Step 2:** Click (**Ctrl+F**) or select **Find & Select** then **Find...** in the Home menu bar:



- **Step 3:** write the word you want to look for and click **Find All** like in the example below:



- **Step 4:** Click (Ctrl+A) or just drag your mouse over all the records in the pop up box then close the pop up box:

Find and Replace

Find Replace

Find what: اعلان

Options >>

Find All Find Next Close

Book	Sheet	Name	Cell	Value	Formula
Book1	Sheet1		\$B\$4	اعلان # غسل الصفاء هو الغسل الأفضل على الإطلاق	
Book1	Sheet1		\$B\$7	اعلان # عطر العود جريه الآن في العربية للعود	

2 cell(s) found

- **Step 5:** Click delete to remove all the selected records:

Users	Tweets
Lola	السلام عليكم ورحمة الله وبركاته
Jerry	حج هذا العام كان رائع
Mohammed	لم يعجبني حج هذا العام لم يكن التنظيم جيداً
Omar	لم يعجبني ردك يا فلان

Removing tweets that have five words or less:

- **Step 1:** Select a column next to the tweets column:

	A	B	C
1	Mohammed	السلام عليكم ورحمة الله وبركاته	
2	Khaled	اسمي خالد	
3	Yasser	كان موسم الحج لهذا العام موسم جميل جدا الصراحة	
4	Yahya	كان تجربة الحج سهلة وميسرة الحمد لله شكراً لكم يا ابطال مكة	
5	Omar	لم تعجبني تجربة الحج لهذا العام بسبب الزحام و عدم التنظيم	
6	Malik	كان تجربة الحج رائعة جدا الحمد لله شكراً لكم	
7	Wejdan		

Step 2: Input this formula” =LEN(B1)-LEN(SUBSTITUTE(B1," ",""))+1” in the formula bar and click enter:

SUM		✕ ✓ fx =LEN(A1)-LEN(SUBSTITUTE(A1," ",""))+1	
	A	B	C
1	Mohammed	السلام عليكم ورحمة الله وبركاته	1
2	Khaled	اسمي خالد	
3	Yasser	كان موسم الحج لهذا العام موسم جميل جدا الصراحة	
4	Yahya	كان تجربة الحج سهلة وميسرة الحمدلله شكراً لكم يا ابطال مكة	
5	Omar	لم تعجبني تجربة الحج لهذا العام بسبب الزحام و عدم التنظيم	
6	Malik	كان تجربة الحج رائعة جدا الحمدلله شكراً لكم	
7	Wejdan		
8			
9			

Step 3: Select records that are contain 5 or less words(you can also use the find... functionality to look for cells that contain a certain number) and delete them:

	A	B	C
1	Mohammed	السلام عليكم ورحمة الله وبركاته	3
2	Khaled	اسمي خالد اليوسفي	5
3	Yasser	كان موسم الحج لهذا العام موسم جميل جدا الصراحة	8
4	Yahya	كان تجربة الحج سهلة وميسرة الحمدلله شكراً لكم يا ابطال مكة	9
5	Omar	لم تعجبني تجربة الحج لهذا العام بسبب الزحام و عدم التنظيم	11
6	Malik	كانت تجربة الحج رائعة جدا الحمدلله شكراً لكم	11
	Yasser	كان موسم الحج لهذا العام موسم جميل جدا الصراحة	8
	Yahya	كان تجربة الحج سهلة وميسرة الحمدلله شكراً لكم يا ابطال مكة	9
	Omar	لم تعجبني تجربة الحج لهذا العام بسبب الزحام و عدم التنظيم	11
	Malik	كانت تجربة الحج رائعة جدا الحمدلله شكراً لكم	11

What we Removed:

Since we used so many words to remove from the file we categorized them here for simpler explanation:

- Swear/Inappropriate words.

Like: (كلب,حيوان,حمار,الخ...)

- Advertisements.

Like: (اعلان, عسل, الخ...)

- Racism and differentiation between different Islamic groups and races.

Like: (كربلاء, شيعي, كافر, الخ...)

- Adultery/expressions of love between users.

Like: (احبك, جنس, الخ...)

- Anything that's not related to the topic at all.

Like: (كيفك, اهلا, الخ...)

We had decided to remove any tweets that contain 5 words or less since they seemed mostly useless and doesn't give us much data to analyze and we used a formula that counts words in each record and sorts them by word number then we manually deleted them since they were all close to each other.

We Started with 2,000,000 tweets separated in two files a million each (since excel can't handle that much data in one sheet) and have reduced our data to approximately 650,000 tweets at the moment.

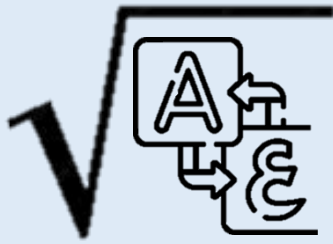
Conclusion:

Cleaning data is necessary before analyzing cause the data will usually include irrelevant and unwanted data that could hinder and falsify our data analysis.

It's a lengthy and time-consuming process but certainly there's AI that can do data cleaning based on certain criteria but it's paid and we were asked to not spend any money for doing course tasks.

Resources:

Resource	Link
Data cleaning article	https://www.tableau.com/learn/articles/what-is-data-cleaning
Removing duplicates	https://support.microsoft.com/en-us/office/find-and-remove-duplicates-00e35bea-b46a-4d5d-b28e-66a552dc138d
Removing based on a certain keyword	- https://support.microsoft.com/en-us/office/find-or-replace-text-and-numbers-on-a-worksheet-0e304ca5-ecef-4808-b90f-fdb42f892e90



Data Arabic word rooting(Stemming)

Objective

In this assignment we are required to return all Arabic words in our tweets back to their root for easier data analysis.

Examples on this:

الكلمة	اصلها
الحجاج	حاج و اصلها حج
يرمي	رمي
أذان	أذن
العمره	عمره

But we need to understand the concept of bringing an Arabic word to its root first(Also known as Stemming):

It is the act of simplifying the word so that it becomes understood in a basic manner, where many words could be returned to the same root, and a root can originate from it many forms of words.

What we did:

At first like always we started looking up what bringing an Arabic word back to its root is and have learned the following:

- Getting the root of an Arabic word requires us to do four steps to get to the root:
 - 1: get the singular form of the word if it was plural
 - 2: get the past verb form of the word if it was in the present form or an imperative or or a source or a derived word.

- 3: remove any extra words so that it's pronounced the same as (فَعَلَ)
- 4: if the second or third letter is (أ) then we should try to replace it with either (و, ي) and thus we'd get the root of the word.

So at first we thought about using the replace function inside excel and replace all Arabic words with their root but getting an Arabic word back to its root or from which word it originated is not an easy task and with the large amount of data it would take a long time to update it manually so we had to look for applications that would do that on the web.

We found a bunch of stemming tools and tried to use all of them but most of them were complicated and using them was rather difficult since they were stemmers for multiple languages not just Arabic or were not precise enough these tools were the following:

- AraRooter(Java) *there were better options
- Assem's Arabic light stemmer(Java)*difficult to use
- SnowballStemmer(Python) *usable but not precise
- Nltk(Python) *better than the ones above but still not precise enough.

In the end we found Farasapy which is a Python library to do more than just stemming in Arabic it can also be used for Segmentation, Named Entity recognition, Part of speech tagging(POS Tagging) and Diacritization.

Using Farasapy(Python library):

Step 1: Installing Farasapy into python:

- In python terminal write (pip install farasapy) to install farasapy

```
PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
Successfully installed farasapy-0.0.14
PS C:\Users\Owner\Desktop\University CS year 3 term 1\Internet Applications\WebPages> pip install farasapy
```

Step 2: importing farasapy stemmer:

- In a new python file write the following

```
1 from farasa.stemmer import FarasaStemmer
```

Step 3: Making a Stemmer object:

```
3 stemmer = FarasaStemmer()
```

Step 4: copy tweets from excel file and input them into a String variable inside 3 single quotation marks in python ("tweets"):

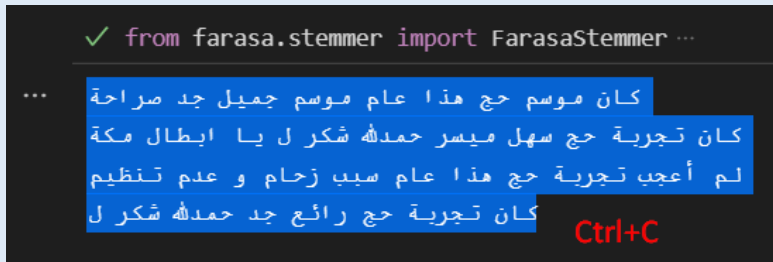
B	1
	كان موسم الحج لهذا العام موسم جميل جدا الصراحة
	كان تجربة الحج سهلة وميسرة الحمدلله شكراً لكم يا ابطال مكة
	لم تعجبني تجربة الحج لهذا العام بسبب الزحام و عدم التنظيم
	كانت تجربة الحج رائعة جدا الحمدلله شكراً لكم
2	
Ctrl+C	

```
4 sample = '''كان موسم الحج لهذا العام موسم جميل جدا الصراحة
5 كان تجربة الحج سهلة وميسرة الحمدلله شكراً لكم يا ابطال مكة
6 لم تعجبني تجربة الحج لهذا العام بسبب الزحام و عدم التنظيم
7 كانت تجربة الحج رائعة جدا الحمدلله شكراً لكم
8 '''
```

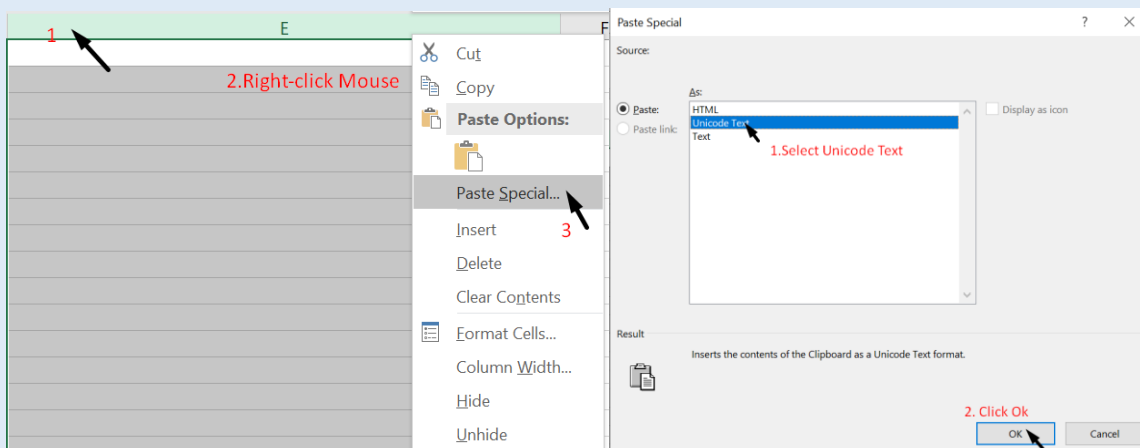
Step 5: use the `stemmer.stem(sample)` method and print the results:

```
12 stemmed_text = stemmer.stem(sample)
13 print(stemmed_text)
```

Step 6: copy the results to place them back into excel:



Step 7: in Excel select a new empty column then use the paste special method to paste your results line by line in each cell:



Step 8: View results:

	A	B	C	D	E
1	Yasser	كان موسم الحج لهذا العام موسم جميل جدا الصراحة	8		كان موسم حج هذا عام موسم جميل جد صراحة
2	Yahya	كان تجربة الحج سهلة وميسرة الحمدلله شكراً لكم يا ابطال مكة	9		كان تجربة حج سهل ميسر حمدلله شكر ل يا ابطال مكة
3	Omar	لم تعجبني تجربة الحج لهذا العام بسبب الزحام و عدم التنظيم	11		لم أعجب تجربة حج هذا عام سبب زحام و عدم تنظيم
4	Malik	كانت تجربة الحج رائعة جدا الحمدلله شكراً لكم	11		كان تجربة حج رائع جد حمدلله شكر ل
5	Wejdan				

What we Modified:

We have stemmed all Arabic words in our tweets to their original stem by stemming 50,000 tweets at a time cause our Computers couldn't handle them all at once and now it's closer to being ready for data analysis.

All the tweets have been added to the pure tweets excel file in a separate row so comparison is easier if needed.

Conclusion:

We found out that in reducing our words to their original stem/root made it easier to categorize and analyze tweets while still keeping the original tweets for comparison.

Finding a good program to do the stemming process was taking us a while to finish cause some of them weren't the best out there but we're confident in Farasapy and our data.

Resources:

Resource	Link
Arabic word rooting	http://www.schoolarabia.net/arabic/alm3ajem_al3rabia/m3ajem_3.htm
Farasapy in Github	https://github.com/MagedSaeed/farasapy
Farasapy User Guide	https://colab.research.google.com/drive/1xjzYwmfAszNzfR6Z2ISQi3nKYcjarXAW?usp=sharing#scrollTo=xIE4sCELNBXZ
Python file opening and editing(optional)	https://www.pythontutorial.net/python-basics/python-read-text-file/

AraRooter(not used)	https://github.com/omarzd/AraRooter
Assem's Arabic light stemmer(not used)	https://arabicstemmer.com/
SnowBall Stemmer	https://www.geeksforgeeks.org/snowball-stemmer-nlp/
Nltk	https://www.nltk.org/



Data Labeling

Objective

In this assignment we are required to label the tweets we collected based on a certain category and response type whether if it's positive, negative or neutral.

Examples on data labeling:

Where (0 = Negative, 1 = Positive, 2 = Neutral)

Tweet	Category	Response type
انا احب يوم عرفة	العبادة	1
المركز الصحي كان سيئاً للأسف	الصحة	0
الان يتم تقديم الطعام في حملة رفاة	حملات الحج	2
حادثة التدافع بمنى يارب ارحمهم واغفر لهم	التنظيم	0

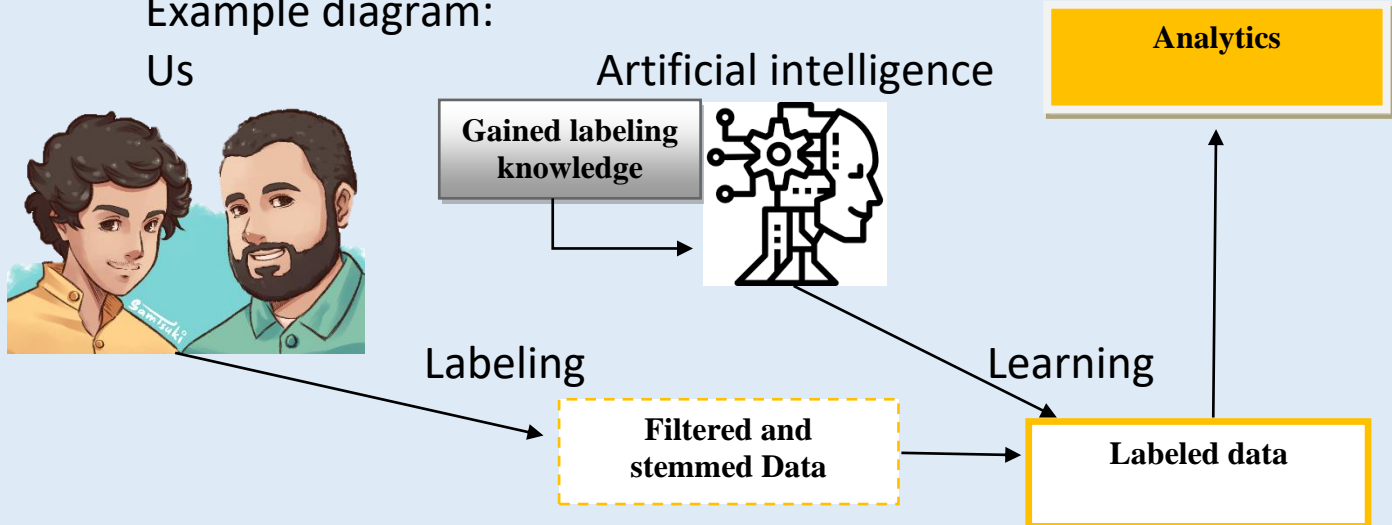
But we need to understand the importance of data labeling to know how we can benefit from it in the next stage.

What we did:

- We learned that data labeling also known as data annotation, tagging, or classification is the process of preparing datasets for algorithms that learn to recognize repetitive patterns in labeled data.
- It helps ai algorithms that are capable of machine learning to learn more about the data and generate analytics through data analysis.

- It is important that we are confident in how we labeled the data ourselves so that if we want to use AI to label in the future it can use our data set to recognize how we labeled the data and learn how to label by itself.

Example diagram:



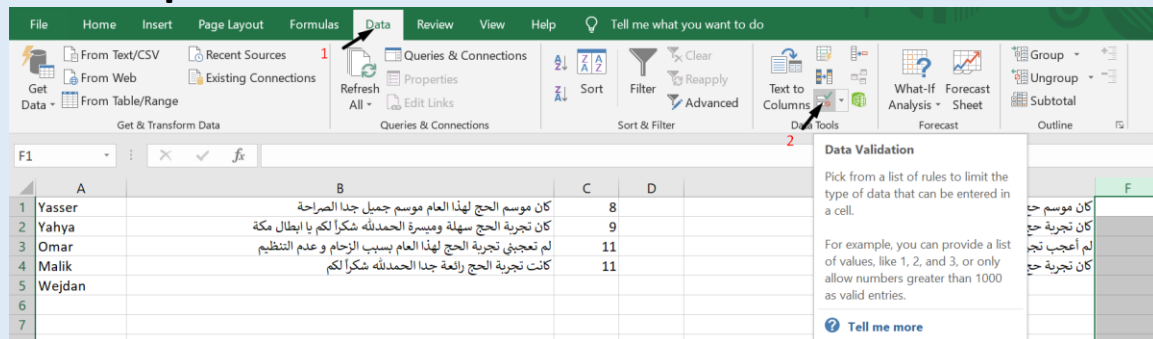
- We also learned about sentiment analysis which is based on the emotions of the person who is writing the tweet in our case.
 - Sentiment analysis (or opinion mining) is a natural language processing (NLP) technique.
 - The result of a sentiment analysis is either positive, negative or neutral.
 - sentiment analysis is important because it helps us understand our user's feelings and opinions about what we're providing, in our instance their feelings about hajj and umrah and the many different services provided during this time.
-

Using Excel(labeling): Creating and using a drop down list:

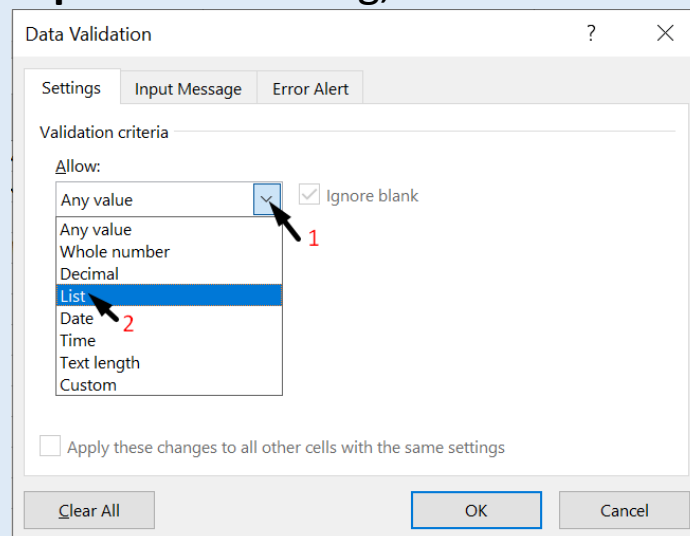
Step 1: Select the cells that you want to contain the lists:

A	B	C	D	E	F
Yasser	كان موسم الحج لهذا العام موسم جميل جدا الصراحة	8		كان موسم حج هذا عام موسم جميل جد صراحة	
Yahya	كان تجربة الحج سهلة وميسرة الحمدلله شكرا لكم يا ابطال مكة	9		كان تجربة حج سهل ميسر حمدلله شكر ل يا ابطال مكة	
Omar	لم تعجبنى تجربة الحج لهذا العام بسبب الزحام و عدم التنظيم	11		لم أعجب تجربة حج هذا عام سبب زحام و عدم تنظيم	
Malik	كانت تجربة الحج رائعة جدا الحمدلله شكرا لكم	11		كان تجربة حج رائع جد حمدلله شكر ل	
Wejdan					

Step 2: Click on data then click on data validation:



Step 3: In the dialog, set Allow to List:



Step 4: a Source text box will appear, click it then type each of your catagories seperated by a comma then click ok:

Data Validation

Settings Input Message Error Alert

Validation criteria

Allow:

List ☐ Ignore blank

Data:

between ☐ In-cell dropdown

Source:

العبادة,الصحة,حملات الحج,التنظيم

☐ Apply these changes to all other cells with the same settings

Clear All OK Cancel

Step 5: select a cell then the arrow next to it to show the catagories then select one of them:

	A	B	C	D	E	F
1	Yasser	كان موسم الحج لهذا العام موسم جميل جدا الصراحة	8		كان موسم حج هذا عام موسم جميل جد صراحة	2
2	Yahya	كان تجربة الحج سهلة وميسرة الحمدلله شكراً لكم يا ابطال مكة	9		كان تجربة حج سهل ميسر حمدلله شكر ل يا ابطال مكة	1
3	Omar	لم تعجبني تجربة الحج لهذا العام بسبب الزحام و عدم التنظيم	11		لم أعجب تجربة حج هذا عام سبب زحام و عدم تنظيم	1
4	Malik	كانت تجربة الحج رائعة جدا الحمدلله شكراً لكم	11		كان تجربة حج رائع جد حمدلله شكر ل	3
5	Wejdan					

Labeling based on response type:

Step 1: manually check whether the tweet Is positive, negative or neutral based on the feeling of the user:

A	B	C	D	E	F	G
Yasser	كان موسم الحج لهذا العام موسم جميل جدا الصراحة	8		كان موسم حج هذا عام موسم جميل جد صراحة	التنظيم	
Yahya	كان تجربة الحج سهلة وميسرة الحمدلله شكراً لكم يا ابطال مكة	9		كان تجربة حج سهل ميسر حمدلله شكر ل يا ابطال مكة	التنظيم	
Omar	لم تعجبني تجربة الحج لهذا العام بسبب الزحام و عدم التنظيم	11		لم أعجب تجربة حج هذا عام سبب زحام و عدم تنظيم	التنظيم	
Malik	كانت تجربة الحج رائعة جدا الحمدلله شكراً لكم	11		كان تجربة حج رائع جد حمدلله شكر ل	التنظيم	

Step 2: in an empty column select the cell corresponding to the tweet you want to label and type (0 if negative, 1 if positive, 2 if neutral):

A	B	C	D	E	F	G
Yasser	كان موسم الحج لهذا العام موسم جميل جدا الصراحة	8		كان موسم حج هذا عام موسم جميل جد صراحة	التنظيم	1
Yahya	كان تجربة الحج سهلة وميسرة الحمدلله شكراً لكم يا ابطال مكة	9		كان تجربة حج سهل ميسر حمدلله شكر ل يا ابطال مكة	التنظيم	1
Omar	لم تعجبني تجربة الحج لهذا العام بسبب الزحام و عدم التنظيم	11		لم أعجب تجربة حج هذا عام سبب زحام و عدم تنظيم	التنظيم	0
Malik	كانت تجربة الحج رائعة جدا الحمدلله شكراً لكم	11		كان تجربة حج رائع جد حمدلله شكر ل	التنظيم	1
Wejdan	لا اله الا الله محمد رسول الله صلوا على رسول الله			لا اله الا الله محمد رسول الله صلوا على رسول الله	العبادة	2

What we accomplished:

After learning how to label data we were given categories to label our data from and these were the following:

- السكن : Anything related to Pilgrims opinions about their place of living for the period of hajj or umrah.
- الأمن : Anything related to police officer's actions or law related topics/crimes within the period of hajj and umrah.
- العبادة : Anything related to Islamic actions or days and celebrations.
- المواصلات : Anything related to the Pilgrims opinions about means of travel(taxi, train, etc.)within the period of hajj and umrah.
- الصحة : Anything related to health in general from Pilgrims opinions on hospitals or health centers to accomplishments of the health industry.
- التنظيم : Anything related to the organization of Pilgrims movements within the different locations of Makkah.
- حملات الحج : Anything related to the organizations that are directly hosting services for Pilgrims.

- الإعاشة والتغذية : Anything related to the catering of Pilgrims during the period of hajj and umrah.
- الطقس : Anything related to the weather situation whether it was hot or rainy or else.
- التعامل : Anything related to the way the Pilgrims were treated by service providers, Police officers, staff or other pilgrims.
- خدمات الدعم اللوجيستي (التقنية - الاتصالات - التأشيرة - الطيران) : Anything related to logistic services whether technological, flight landing and departure, visa, etc.

And also we were given a way to label response type(sentiments) in the following way:

- 0 for Negative responses where the user felt bad, unsatisfied, annoyed, hurt, remorseful or any bad feeling.
- 1 for Positive responses where the user felt good, satisfied, pleased, glad to come, grateful or any good feeling
- 2 for Neutral responses where the user felt neither good or bad or didn't express their opinion at all.

So we started manually looking through the tweets one by one and since in the beginning our data was very big we decided if we couldn't finish the tweets in time we'll use the tweets we already labeled and with the help of machine learning label the rest of the data but hopefully we'll be able to finish it all.

But we have labeled about 7000 tweets manually divided between the two of us.

We also noted that most of the tweets are either Islamic prayers(duaa) or mentions of the Mina Stampede accidents...which is quite unfortunate but we also noticed that the following years were often better than the ones before.

Conclusion:

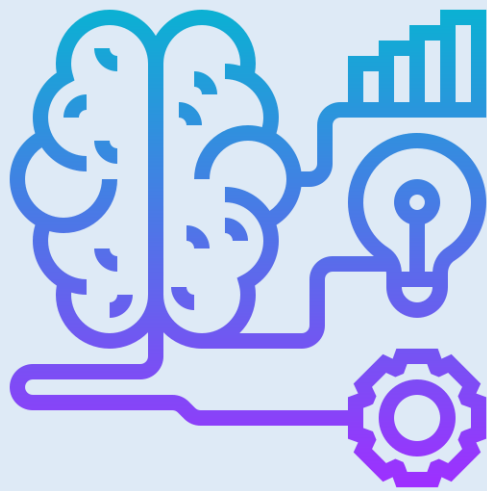
In the end, we learned how important this part of the data collecting process is to the data analysis section hence why we need to mostly do it manually.

Categorizing the data helps us collect data of a certain field or topic/product and know what to do to make it better or fix it if it has errors.

Sentiment Analysis helps us know our user base better and what they like and dislike which helps in future improvement or our services.

Resources:

Resource	Link
Importance of data labeling	https://www.tasq.ai/blog/the-importance-of-data-labeling-and-how-we-ensure-high-quality/
Sentiment Analysis Guide	https://monkeylearn.com/sentiment-analysis/
How to create a drop-down list in excel	https://support.microsoft.com/en-us/office/video-create-and-manage-drop-down-lists-28db87b6-725f-49d7-9b29-ab4bc56cefc2



Machine Learning in Data Classification

Objective

Now that we have a reasonable amount of labeled data we can use machine learning to help us classify and label the rest of our data.

Using the labeled part of our data to teach and test the accuracy of the three algorithms we will use and decide which one we will use to label the rest of the data.

But before that we need to learn a bit more about what machine learning is and how we can use it to classify our data.

What we did:

We learned:

- Machine learning is a branch of artificial intelligence (AI) and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, while gradually improving its accuracy.
- Machine learning models have three main categories which are supervised, unsupervised and semi-supervised learning.
- Supervised learning, also known as supervised machine learning, is defined by its use of labeled datasets to train algorithms to classify data or predict outcomes accurately.
- Unsupervised learning, also known as unsupervised machine learning, uses machine learning algorithms to analyze and cluster unlabeled datasets. These

- algorithms discover hidden patterns or data groupings without the need for human intervention.
- Supervised learning can solve two types of problems which are classification and regression.
 - Unsupervised learning can solve three types of problems which are clustering, association and Dimensionality reduction.
 - The main difference between the two approaches is the use of labeled datasets. To put it simply, supervised learning uses labeled input and output data, while an unsupervised learning algorithm does not.
 - There are many different algorithms that can be used to help in machine learning and some of them are, naïve bayes, decision trees, random forest, support vector machine and K- nearest neighbor which vary in helpfulness and accuracy of correct results.

What we want in the end is to test three different machine learning algorithms using supervised learning to classify our data correctly, these algorithms are:

- Naïve Bayes (NB)
- Support Vector Machine (SVM)
- K-Nearest Neighbor (KNN)

Using Python(Scikit-learn) for machine learning:

Naïve Bayes:

Step 1: Importing needed packages:

```
1  import os
2  import re
3  import string
4  import math
5  from decimal import Decimal
6  import pandas as pd
```

Step 2: Getting the information from the excel file(which was reduced to only two columns, rooted pure tweets and their sentiment classification) and removing any null values if they exist and last but not least removing the indexes since we don't need them in the machine learning process:

```
8  dataset = pd.read_excel('C://Users//Owner//Downloads//labe_tweets (2).xlsx')
9  dataset=dataset.dropna()
10 dataset=dataset.reset_index(drop=True)
```

Step 3: assigning each column to a variable and converting the tweets column variable to a dictionary to be ready to use by scikit-learn:

```
12 x=dataset.iloc[:,0]
13 y=dataset.iloc[:,1]
14 X=x.to_dict()
```

Step 4: Creating an empty array and inserting all the elements from the dictionary(tweets) into it:

```
16 X=[]
17 for d in range(len(x)):
18     b=x[d]
19     X.append(b)
```

Step 5: Importing Scikit-learn's feature extraction module that extracts features from raw data then making a count vectorizer(which converts a collection of text documents to a matrix of token counts) then using the **fit.transform()** method that's within the count vectorizer which learns the vocabulary of the text and returns a document term matrix (DTM) then turning it back into an array for the next step:

```
22 from sklearn.feature_extraction.text import CountVectorizer, TfidfTransformer
23 count_vect=CountVectorizer()
24 a=count_vect.fit_transform(X)
25 a.toarray()
```

Step 6: Importing Scikit-learn's model_selection module splitting the data into train and test subsets(80% train and 20% test) where the train subset is used by the machine to learn how to categorize and the test subset is used to apply what it learned and compare it with the actual classification to know how accurate it is:

```
27 from sklearn.model_selection import train_test_split
28 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)
```

Step 7: Making a new count vectorizer then using the **fit.transform()** method but this time with the train data set(it was converted to an array before so we need to do this again) then making a tf-idf transformer(term-frequency times inverse document-frequency)which is a weighing scheme often used in data classification to also apply the **fit.transform()** method to transform the train data into tf-idf then finally converting it into an array:

```
30 count_vect=CountVectorizer()
31 X_train_counts=count_vect.fit_transform(X_train)
32 tfidf_transformer = TfidfTransformer()
33 X_train_tfidf = tfidf_transformer.fit_transform(X_train_counts)
34 X_train_tfidf.toarray()
```

Step 8: importing the Naïve Bayes method from Scikit-learn package and creating a Naïve Bayes object then fitting our train and test subsets into the Naïve Bayes object and using the **score(train_data,test_data)** method to test our accuracy:

```
38  from sklearn.naive_bayes import MultinomialNB
39
40  nb= MultinomialNB()
41  nb.fit(X_train_tfidf, y_train)
42  nb.score(X_train_tfidf, y_train)
```

Step 9: transforming the test data into a document term matrix(DTM) to make a prediction then using the **predict(test data as DTM)** method to do prediction(which is how to trigger the machine to do the classification process on the data in the parameter):

```
44  #transform X_text for prediction
45  X_test_tfidf=count_vect.transform(X_test)
46
47  #prediction
48  y_pred=nb.predict(X_test_tfidf)
```

Step 10: importing from the scikit-learn package the metrics module which enables us to calculate the confusion matrix(how many data did it classify correctly) and the accuracy score(how accurate was the machine in predicting the classification for the test data in the form of a percentage)then printing both of them:

```
51  from sklearn.metrics import confusion_matrix,accuracy_score
52  cm = confusion_matrix(y_test, y_pred)
53  Accuracy_Score = accuracy_score(y_test, y_pred)
54
55  print(cm)
56  print(Accuracy_Score)
```

```
✓ import os ...
...  [[ 91  34   0]
      [  3 754   0]
      [  2 122   0]]
      0.8399602385685885
```


Support Vector Machine (SVM):

Step 1: Importing needed packages:

```
1  import os
2  import re
3  import string
4  import math
5  from decimal import Decimal
6  import pandas as pd
```

Step 2: Getting the information from the excel file(which was reduced to only two columns, rooted pure tweets and their sentiment classification) and removing any null values if they exist and last but not least removing the indexes since we don't need them in the machine learning process:

```
8  dataset = pd.read_excel('C://Users//Owner//Downloads//lable_tweets (2).xlsx')
9  dataset=dataset.dropna()
10 dataset=dataset.reset_index(drop=True)
```

Step 3: assigning each column to a variable and converting the tweets column variable to a dictionary to be ready to use by scikit-learn:

```
12 x=dataset.iloc[:,0]
13 y=dataset.iloc[:,1]
14 X=x.to_dict()
```

Step 4: Creating an empty array and inserting all the elements from the dictionary(tweets) into it:

```
16 X=[]
17 for d in range(len(x)):
18     b=x[d]
19     X.append(b)
```

Step 5: Importing Scikit-learn's feature extraction module that extracts features from raw data then making a count vectorizer(which converts a collection of text documents to a matrix of token counts) then using the **fit.transform()** method that's within the count vectorizer which learns the vocabulary of the text and returns a document term matrix (DTM) then turning it back into an array for the next step:

```
22 from sklearn.feature_extraction.text import CountVectorizer, TfidfTransformer
23 count_vect=CountVectorizer()
24 a=count_vect.fit_transform(X)
25 a.toarray()
```

Step 6: Importing Scikit-learn's model_selection module splitting the data into train and test subsets(80% train and 20% test) where the train subset is used by the machine to learn how to categorize and the test subset is used to apply what it learned and compare it with the actual classification to know how accurate it is:

```
27 from sklearn.model_selection import train_test_split
28 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)
```

Step 7: Making a new count vectorizer then using the **fit.transform()** method but this time with the train data set(it was converted to an array before so we need to do this again) then making a tf-idf transformer(term-frequency times inverse document-frequency)which is a weighing scheme often used in data classification to also apply the **fit.transform()** method to transform the train data into tf-idf then finally converting it into an array:

```
30 count_vect=CountVectorizer()
31 X_train_counts=count_vect.fit_transform(X_train)
32 tfidf_transformer = TfidfTransformer()
33 X_train_tfidf = tfidf_transformer.fit_transform(X_train_counts)
34 X_train_tfidf.toarray()
```

Step 8: Importing the random over_sampling module from the imblearn package using random oversampling which is the duplication of some elements from the minority class in the training data set which is then inserted back into the rest of the training data(which helps when our data isn't very balanced):

```
37 from imblearn.over_sampling import RandomOverSampler
38 sm=RandomOverSampler()
39 X_train_res, y_train_res = sm.fit_resample(X_train_tfidf, y_train)
```

Step 9: importing the SVM algorithm from Scikit-learn package and creating an SVM object then fitting our train and test subsets into the SVM object and using the **score(train_data,test_data)** method to test our accuracy:

```
43 from sklearn.svm import SVC
44
45 clf= SVC(kernel = 'rbf', random_state = 0)
46 clf.fit(X_train_res, y_train_res)
47 clf.score(X_train_res, y_train_res)
```

Step 10: transforming the test data into a document term matrix(DTM) to make a prediction then using the **predict(test data as DTM)** method to do prediction(which is how to trigger the machine to do the classification process on the data in the parameter):

```
44 #transform X_text for prediction
45 X_test_tfidf=count_vect.transform(X_test)
46
47 #prediction
48 y_pred=nb.predict(X_test_tfidf)
```

Step 11: importing from the scikit-learn package the metrics module which enables us to calculate the confusion matrix(how many data did it classify correctly) and the accuracy score(how accurate was the machine in predicting the classification for the test data in the form of a percentage)then printing both of them:

```
51 from sklearn.metrics import confusion_matrix, accuracy_score
52 cm = confusion_matrix(y_test, y_pred)
53 Accuracy_Score = accuracy_score(y_test, y_pred)
54
55 print(cm)
56 print(Accuracy_Score)
```

```
[[ 0 125  0]
 [ 0 757  0]
 [ 0 124  0]]
0.7524850894632207
```

K-Nearest Neighbor (KNN):

Step 1: Importing needed packages:

```
1 import os
2 import re
3 import string
4 import math
5 from decimal import Decimal
6 import pandas as pd
```

Step 2: Getting the information from the excel file(which was reduced to only two columns, rooted pure tweets and their sentiment classification) and removing any null values if they exist and last but not least removing the indexes since we don't need them in the machine learning process:

```
8 dataset = pd.read_excel('C://Users//Owner//Downloads//lable_tweets (2).xlsx')
9 dataset=dataset.dropna()
10 dataset=dataset.reset_index(drop=True)
```

Step 3: assigning each column to a variable and converting the tweets column variable to a dictionary to be ready to use by scikit-learn:

```
12 x=dataset.iloc[:,0]
13 y=dataset.iloc[:,1]
14 X=x.to_dict()
```

Step 4: Creating an empty array and inserting all the elements from the dictionary(tweets) into it:

```
16 X=[]
17 for d in range(len(x)):
18     b=x[d]
19     X.append(b)
```

Step 5: Importing Scikit-learn's feature extraction module that extracts features from raw data then making a count vectorizer(which converts a collection of text documents to a matrix of token counts) then using the **fit.transform()** method that's within the count vectorizer which learns the vocabulary of the text and returns a document term matrix (DTM) then turning it back into an array for the next step:

```
22 from sklearn.feature_extraction.text import CountVectorizer, TfidfTransformer
23 count_vect=CountVectorizer()
24 a=count_vect.fit_transform(X)
25 a.toarray()
```

Step 6: Importing Scikit-learn's model_selection module splitting the data into train and test subsets(80% train and 20% test) where the train subset is used by the machine to learn how to categorize and the test subset is used to apply what it learned and compare it with the actual classification to know how accurate it is:

```
27 from sklearn.model_selection import train_test_split
28 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)
```

Step 7: Making a new count vectorizer then using the **fit.transform()** method but this time with the train data set(it was converted to an array before so we need to do this again) then making a tf-idf transformer(term-frequency times inverse document-frequency)which is a weighing scheme often used in data classification to also apply the **fit.transform()** method to transform the train data into tf-idf then finally converting it into an array:

```
30 count_vect=CountVectorizer()  
31 X_train_counts=count_vect.fit_transform(X_train)  
32 tfidf_transformer = TfidfTransformer()  
33 X_train_tfidf = tfidf_transformer.fit_transform(X_train_counts)  
34 X_train_tfidf.toarray()
```

Step 8: Importing the Smote(synthetic minority oversampling technique) module from the imblearn package using smote oversampling which is the randomly increasing minority class examples by replicating them which are then inserted back into the rest of the training data among the minority instances(which helps when our data isn't very balanced):

```
36 from imblearn.over_sampling import SMOTE  
37 sm=SMOTE()  
38 X_train_res, y_train_res = sm.fit_resample(X_train_tfidf, y_train)
```

Step 9: importing the KNNclassifier algorithm from Scikit-learn package and creating an KNNclassifier object and also specifying number of neighbors and using the minkowski metric system the assigning $p = 2$ which will make the method use Euclidian distance then fitting our train and test subsets into the KNNclassifier object:

```
41 from sklearn.neighbors import KNeighborsClassifier  
42 clf = KNeighborsClassifier(n_neighbors = 2, metric = 'minkowski', p = 2)  
43 clf.fit(X_train_res, y_train_res)
```

Step 10: transforming the test data into a document term matrix(DTM) to make a prediction then using the **predict(test data as DTM)** method to do prediction(which is how to trigger the machine to do the classification process on the data in the parameter):

```
44 #transform X_text for prediction
45 x_test_tfidf=count_vect.transform(X_test)
46
47 #prediction
48 y_pred=nb.predict(x_test_tfidf)
```

Step 11: importing from the scikit-learn package the metrics module which enables us to calculate the confusion matrix(how many data did it classify correctly) and the accuracy score(how accurate was the machine in predicting the classification for the test data in the form of a percentage)then printing both of them:

```
51 from sklearn.metrics import confusion_matrix,accuracy_score
52 cm = confusion_matrix(y_test, y_pred)
53 Accuracy_Score = accuracy_score(y_test, y_pred)
54
55 print(cm)
56 print(Accuracy_Score)
```

```
... [[100  20   5]
     [ 24 683  50]
     [  6  87  31]]
0.8091451292246521
```

What we accomplished:

At first, we wanted to use the R language to do our machine learning we have used it but had discovered that the methods we used couldn't analyze Arabic text so we started looking for alternatives.

Our eyes were on Python's package scikit-learn so we looked up ways where we can use it with Arabic text and

we found some code in GitHub that we carefully studied and modified a little bit to do machine learning on our dataset.

The code's sample excel file used similar data and the same categorization for a tweet's sentiment(0,1,2) so we used the code with our data.

In the end, out of the three algorithms we used

- Naïve Bayes (NB)

Accuracy = roughly 84%

- Support Vector Machine (SVM)

Accuracy = 75%

- K-Nearest Neighbor (KNN)

Accuracy = 80%

Naïve Bayes proved to be the best among the three methods. (where in each algorithm we divided the data into 80% for training and 20% for testing).

Conclusion:

In the end we learned that using machine learning for data classification can really accelerate the process even if there's a chance that some of the data won't be classified correctly, we can at least get most of the data to be correctly classified.

Machine learning can also be used in the analysis process later on so having some knowledge of how to use it really does help.

After all of this we are finally ready to move onto the next step and we hope we can achieve what we want.

Resources:

Resource	Link
What is machine learning?	https://www.ibm.com/topics/machine-learning
Supervised VS unsupervised learning	https://www.ibm.com/cloud/blog/supervised-vs-unsupervised-learning
How to use Naïve Bayes method in R (Not used anymore)	https://medium.com/@ibtissam.makdoun/steps-to-classify-documents-in-r-ae79c51828c4
How to use Random Forest algorithms in R (Not used anymore)	https://www.pluralsight.com/guides/machine-learning-text-data-using-r
Scikit-learn website	https://scikit-learn.org/stable/
GitHub code for Arabic text classification	https://github.com/FantacherJOY/Arabic-text-classification
Understanding the confusion matrix	https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62
What is Random Over sampling	https://machinelearningmastery.com/random-oversampling-and-undersampling-for-imbalanced-classification/
What Is oversampling using the SMOTE method?	https://www.geeksforgeeks.org/ml-handling-imbalanced-data-with-smote-and-near-miss-algorithm-in-python/

Excel Files

- **Data collection**
 - Task 1_1
 - Task 1_2
- **Data Cleaning**
 - Task 2
- **Data Rooting(Stemming)**
 - Task 3
- **Data Labeling**
 - Task 4
- **Machine Learning in Data Classification**
 - Task 5(File used to teach the machine)