كلية علوم الحاسب والمعلومات

College of Computer and Information Sciences
Department of Software Engineering

SWE485: Selected Topics In Software Engineering
2nd Semester 1440-1441 H King Saud University

Phase #4 | Data Visualization and Findings

Sun. 11:59 p.m April 19th.

Team members:

- Reema AlHarbi
  ID: 436202371
- Raghad AlOliwi
  ID: 437200294
- Aljohara AlRshaid
  ID: 437200410
- Aljohara AlDahmigh
  ID: 437201812
- Lama AlQasem
  ID: 437201546
- Albtool AlKhudairy
  ID: 437202224

Section: 54130

Supervisors: Ms. Bayan AlArifi, Ms. Monira AlDelaimi and Ms. Mona Hakami

Table of content

1. Project Description

1.1 Problem Description:

Riyadh season is a mega entertainment event that took place from mid October to December 2019 for seventy days. The season was held in twelve different zones in the capital city Riyadh, which hosted more than one hundred activities and programs.

On the 17th of October, Hittin district's residents noticed irregular traffic around their houses, that was aligned with Riyadh season starting their first festivity in Zone One: Riyadh Boulevard, that is located in the same district.

Although Riyadh Boulevard's capacity can hold up to 60,000 visitors[1], the number of actual visitors was way more than that, it resulted in making people park their cars anywhere around that place, including in front of residences' garages.

Hittin residents started complaining on **Twitter** about the levels of noise, the traffic and the difficulty of leaving the district without facing the severe overcrowding. Furthermore, they started a hashtag (called #معاناة_سكان_حطين) that is dedicated to delivering their message to the official authorities.

Hittin district residents' struggle enlightens us with many questions concerning the appropriate place to hold a mega event and the importance of crowd control.

[1] Arab News. (n.d.). Festivities around Riyadh Boulevard irk Hittin residents. [online] Available at: https://www.arabnews.com/node/1572426/saudi-arabia [Accessed 22 Feb. 2020].

1.2 Analytical Solution Objective:

In this project, we aim to analyze number of tweets to achieve several objectives, such as:

- Measure and describe the consequences of events that are held closely to residential neighborhoods.
- Specify the importance of appropriate traffic and crowd management for events.
- Explore the possible solutions provided by the resident to solve this struggle, such as:
    - The suitable times for events based on the reactions of the residents who live in or near the neighborhood.
    - Minimize the number of individuals who are accepted to attend an event.
    - Buses that can be provided at an event to reduce the traffic.

1.3 Tools and Libraries:

According to our needs and objectives in this project, which include cleaning and visualizing the extracted data, applying some calculations on them and analyzing the sentiments, we decided to use the following libraries to help us with our aims:

- For data visualization: *Pandas, Plotly.*
- For mathematical calculations: *Numpy*.
- For sentiment analysis: *Mazajak*.
- For cleaning: *Regex or Preprocessor.*

2. Data Collection

In order to achieve our analysis objectives, and as mentioned in the problem section, our main concern in our analysis is Hittin district residents' tweets about Riyadh Season.

We ended up with three ways to reach out the residents' reactions, suggestions and understand the severe overcrowding situation from the people who have experienced it.

The **first** one is tweets tagged with the hashtag (#معاناة_سكان_حطين) or just (معاناة حطين), the **second** is collecting tweets that contain, along with the word 'حطين', one or more of the following key words (حي، إزعاج، هيئة الترفيه، فعاليات، زحمة، إغلاق شوارع، مرور، إزعاج، أقترح، المفترض، نرجو، نتمنى، معالي، نقترح، نطالب، يحتاج، هدوء، موسم، موسم الرياض، الموسم، أصوات، عالية، مزعجة، مغلق، أغلقوا), and the **last** one is the geo-tagged tweets in Hittin district.

Initially, we decided to go with the first approach (the tweets with معاناة حطين), since we have noticed that it contains various and huge amount of data (almost 881 tweets) and very little spam tweets comparing to searching for random words like (إزعاج، زحمة) that contains other contexts that we are not interested in.

2.1 Experience:

At first, we had a problem using Tweepy library to connect to Twitter API, since most of the tweets were between October and December, Twitter did not allow us to request old tweets (although some were days old only). We have discovered this by tweeting (معاناة حطين) to test if this was the problem, and it was.

We had to find another way to retrieve older tweets. We have used `GetOldTweets3` library to help us access old tweets without Twitter limitations.

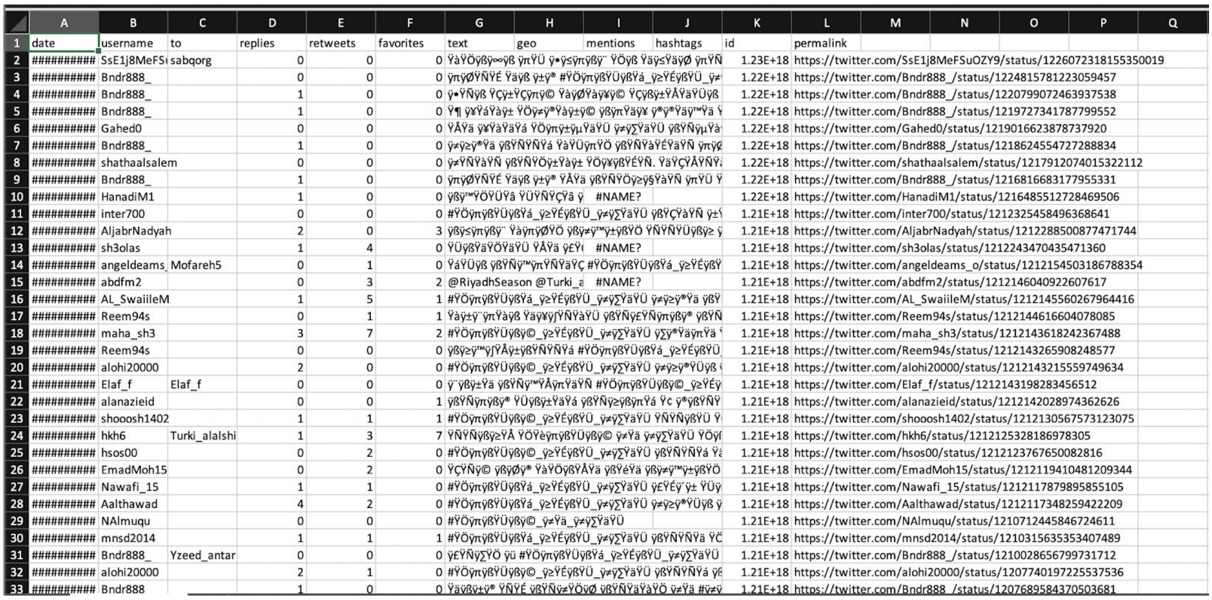
We had to install the library with:

```
pip install GetOldTweets3
```

And then specifying keywords, output file name and extension, and maximum number of tweets:

```
GetOldTweets3 --querysearch "معاناة حطين" --output "HittenTweets.csv" --maxtweets
1000
```

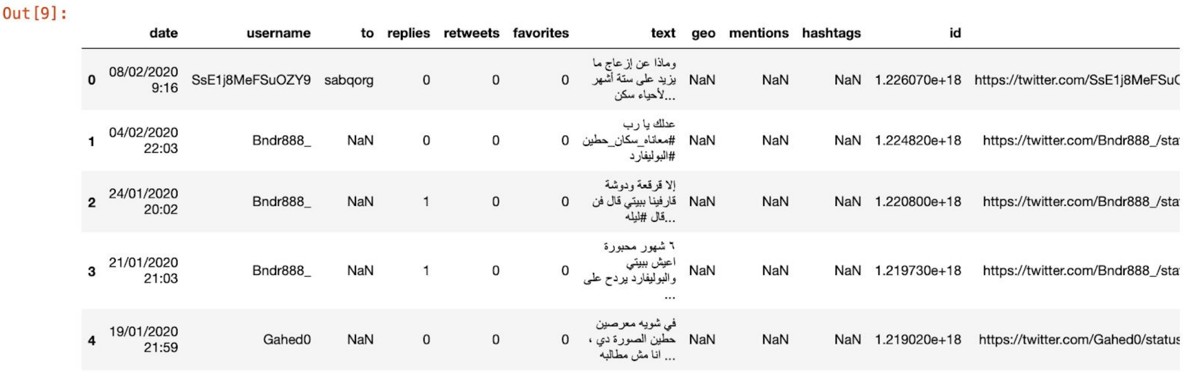
Which outputs the retrieved tweets in the following file as shown:



	A	B	C	D	E	F	G	H	I	J	K	L
1	date	username	to	replies	retweets	favorites	text	geo	mentions	hashtags	id	permalink
2	##########	SsE1j8MeFSı	sabqorg	0	0	0	ŸàŸÖÿßÿ∞ÿß ÿπŸÜ ÿ•ÿ≤ÿπÿßÿ¨ ŸÖÿß Ÿäÿ≤Ÿäÿø ÿπŸÑ				1.23E+18	https://twitter.com/SsE1j8MeFSuOZY9/status/1226072318155350019
3	##########	Bndr888_		0	0	0	ÿπÿØŸÑŸÉ Ÿäÿß ÿ±ÿ® #ŸÖÿπÿßŸÜÿßÿ©_ÿ≥ŸÉÿßŸÜ_ÿ≠				1.22E+18	https://twitter.com/Bndr888_/status/1224815781223059457
4	##########	Bndr888_		1	0	0	ÿ•ŸÑÿß ŸÇÿ±ŸÇÿπÿ© ŸàÿØŸà¥ÿ© ŸÇÿßÿ±ŸÅŸäŸÜÿß				1.22E+18	https://twitter.com/Bndr888_/status/1220799072463937538
5	##########	Bndr888_		1	0	0	Ÿ¶ ÿ¥ŸáŸàÿ± ŸÖÿ≠ÿ®ŸàÿÆÿ© ÿßÿπŸäÿ¥ ÿ®ÿ®Ÿäÿ™Ÿä Ÿ				1.22E+18	https://twitter.com/Bndr888_/status/1219727341787799552
6	##########	Gahed0		0	0	0	ŸÅŸä ÿ¥ŸàŸäŸá ŸÖÿπÿ±ÿµŸäŸÜ ÿ≠ÿ∑ŸäŸÜ ÿßŸÑÿµŸà				1.22E+18	https://twitter.com/Gahed0/status/1219016623878737920
7	##########	Bndr888_		1	0	0	ÿ≠ÿ≥ÿ®Ÿä ÿßŸÑŸÑŸá ŸàŸÜÿπŸÖ ÿßŸÑŸàŸÉŸäŸÑ ÿπÿ				1.22E+18	https://twitter.com/Bndr888_/status/1218624554727288834
8	##########	shathaalsalem		0	0	0	ÿ≠ŸÑŸàŸÑ ÿßŸÑŸÖÿ±Ÿàÿ± ŸÖÿ¥ÿßŸÉŸÑ. ŸäŸÇŸÅŸÑŸä				1.22E+18	https://twitter.com/shathaalsalem/status/1217912074015322112
9	##########	Bndr888_		1	0	0	ÿπÿØŸÑŸÉ Ÿäÿß ÿ±ÿ® ŸÅŸä ÿßŸÑŸÖÿ≥ÿ§ŸàŸÑ ÿπŸÜ Ÿ				1.22E+18	https://twitter.com/Bndr888_/status/1216816683177955331
10	##########	HanadiM1		1	0	0	ÿßÿ™ŸÖŸÜŸâ ŸÜŸÑŸÇŸâ ÿ		#NAME?		1.22E+18	https://twitter.com/HanadiM1/status/1216485512728469506
11	##########	inter700		0	0	0	#ŸÖÿπÿßŸÜÿßÿ©_ÿ≥ŸÉÿßŸÜ_ÿ≠ÿ∑ŸäŸÜ ÿßŸÇŸàŸÑ ÿ±Ÿ				1.21E+18	https://twitter.com/inter700/status/1212325458496368641
12	##########	AljabrNadyah		2	0	3	ÿßÿ≤ÿπÿßÿ¨ ŸàÿπÿØŸÖ ÿßÿ≠ÿ™ÿ±ÿßŸÖ ŸÑŸÑŸÜÿßÿ≥ ÿ				1.21E+18	https://twitter.com/AljabrNadyah/status/1212288500877471744
13	##########	sh3olas		1	4	0	ŸÜÿßŸäŸÖŸäŸÜ ŸÅŸä ÿ£Ÿ(		#NAME?		1.21E+18	https://twitter.com/sh3olas/status/1212243470435471360
14	##########	angeldeams_	Mofareh5	0	1	0	ŸáŸÜÿß ÿßŸÑÿ™ÿπŸÑŸäŸÇ #ŸÖÿπÿßŸÜÿßÿ©_ÿ≥ŸÉÿßŸ				1.21E+18	https://twitter.com/angeldeams_o/status/1212154503186788354
15	##########	abdfm2		0	3	2	@RiyadhSeason @Turki_a		#NAME?		1.21E+18	https://twitter.com/abdfm2/status/1212146040922607617
16	##########	AL_SwaiileM		1	5	1	#ŸÖÿπÿßŸÜÿßÿ©_ÿ≥ŸÉÿßŸÜ_ÿ≠ÿ∑ŸäŸÜ ÿ≠ÿ≥ÿ®Ÿä ÿßŸ				1.21E+18	https://twitter.com/AL_SwaiileM/status/1212145560267964416
17	##########	Reem94s		0	1	1	Ÿàÿ±ÿ¨ÿπŸàÿß Ÿäÿ¥ÿ∫ŸÑŸàŸÜ ÿßŸÑŸÖŸÉŸÜÿßÿ≥ ÿßŸÑ				1.21E+18	https://twitter.com/Reem94s/status/1212144616604078085
18	##########	maha_sh3		3	7	2	#ŸÖÿπÿßŸÜÿßÿ©_ÿ≥ŸÉÿßŸÜ_ÿ≠ÿ∑ŸäŸÜ ÿ∑ÿ®ŸäÿπŸä Ÿ				1.21E+18	https://twitter.com/maha_sh3/status/1212143618242367488
19	##########	Reem94s		0	0	0	ÿßÿ≥ÿ™ÿ∫ŸÅÿ±ÿßŸÑŸÑŸá #ŸÖÿπÿßŸÜÿßÿ©_ÿ≥ŸÉÿßŸÜ_				1.21E+18	https://twitter.com/Reem94s/status/1212143265908248577
20	##########	alohi20000		2	0	0	#ŸÖÿπÿßŸÜÿßÿ©_ÿ≥ŸÉÿßŸÜ_ÿ≠ÿ∑ŸäŸÜ ÿ≠ÿ≥ÿ®ŸÜÿß Ÿ				1.21E+18	https://twitter.com/alohi20000/status/1212143215559749634
21	##########	Elaf_f	Elaf_f	0	0	0	ÿ¨ÿßÿ±Ÿä ÿßŸÑÿ™ŸÅÿπŸäŸÑ #ŸÖÿπÿßŸÜÿßÿ©_ÿ≥ŸÉÿ				1.21E+18	https://twitter.com/Elaf_f/status/1212143198283456512
22	##########	alanazieid		0	0	1	ÿßŸÑÿπÿßÿ® ŸÜÿßÿ±Ÿäÿ© ÿßŸÑÿ≥ÿßÿπÿ© Ÿ¢ ÿ®ÿßŸÑŸ				1.21E+18	https://twitter.com/alanazieid/status/1212142028974362626
23	##########	shooosh1402		1	1	1	#ŸÖÿπÿßŸÜÿßÿ©_ÿ≥ŸÉÿßŸÜ_ÿ≠ÿ∑ŸäŸÜ ŸÑŸÑÿßŸÜ Ÿ(				1.21E+18	https://twitter.com/shooosh1402/status/1212130567573123075
24	##########	hkh6	Turki_alalshi	1	3	7	ŸÑŸÑÿßÿ≥Ÿ ŸÖŸèÿπÿßŸÜÿßÿ© ÿ≠Ÿä ÿ≠ÿ∑ŸäŸÜ ŸÖÿ(				1.21E+18	https://twitter.com/hkh6/status/1212125328186978305
25	##########	hsos00		0	2	0	#ŸÖÿπÿßŸÜÿßÿ©_ÿ≥ŸÉÿßŸÜ_ÿ≠ÿ∑ŸäŸÜ ÿßŸÑŸÑŸá Ÿä				1.21E+18	https://twitter.com/hsos00/status/1212123767650082816
26	##########	EmadMoh15		0	2	0	ŸÇŸÑÿ© ÿßÿØÿ® ŸàŸÖÿßŸÅŸä ÿßŸéŸä ÿßÿ≠ÿ™ÿ±ÿßŸÖ				1.21E+18	https://twitter.com/EmadMoh15/status/1212119410481209344
27	##########	Nawafi_15		1	1	0	#ŸÖÿπÿßŸÜÿßÿ©_ÿ≥ŸÉÿßŸÜ_ÿ≠ÿ∑ŸäŸÜ ÿ£ŸÉÿ´ÿ± ŸÜÿ				1.21E+18	https://twitter.com/Nawafi_15/status/1212117879895855105
28	##########	Aalthawad		4	2	0	#ŸÖÿπÿßŸÜÿßÿ©_ÿ≥ŸÉÿßŸÜ_ÿ≠ÿ∑ŸäŸÜ ÿ≠ÿ≥ÿ®ŸÜÿß ÿ				1.21E+18	https://twitter.com/Aalthawad/status/1212117348259422209
29	##########	NAlmuqu		0	0	0	#ŸÖÿπÿßŸÜÿ©_ÿ≠Ÿä_ÿ≠ÿ∑ŸäŸÜ				1.21E+18	https://twitter.com/NAlmuqu/status/1210712445846724611
30	##########	mnsd2014		1	1	1	#ŸÖÿπÿßŸÜÿßÿ©_ÿ≥ŸÉÿßŸÜ_ÿ≠ÿ∑ŸäŸÜ ÿßŸÑŸÑŸä ŸÇ				1.21E+18	https://twitter.com/mnsd2014/status/1210315635353407489
31	##########	Bndr888_	Yzeed_antar	0	0	0	ÿ£ŸÑÿ∑Ö ÿü #ŸÖÿπÿßŸÜÿßÿ©_ÿ≥ŸÉÿßŸÜ_ÿ≠ÿ∑ŸäŸÜ				1.21E+18	https://twitter.com/Bndr888_/status/1210028656799731712
32	##########	alohi20000		2	1	0	#ŸÖÿπÿßŸÜÿßÿ©_ÿ≥ŸÉÿßŸÜ_ÿ≠ÿ∑ŸäŸÜ ÿßŸÑŸÑŸá ÿ				1.21E+18	https://twitter.com/alohi20000/status/1207740197225537536
33	##########	Bndr888		1	0	0	Ÿäÿßÿ±ÿ® ŸÑŸÉ ÿßŸÑÿ≠ŸÖÿØ ÿßŸÑŸäŸàŸÖ ÿ≠Ÿä #ÿ≠ÿ				1.21E+18	https://twitter.com/Bndr888_/status/1207689584370503681

Figure 1: HittenTweets.csv

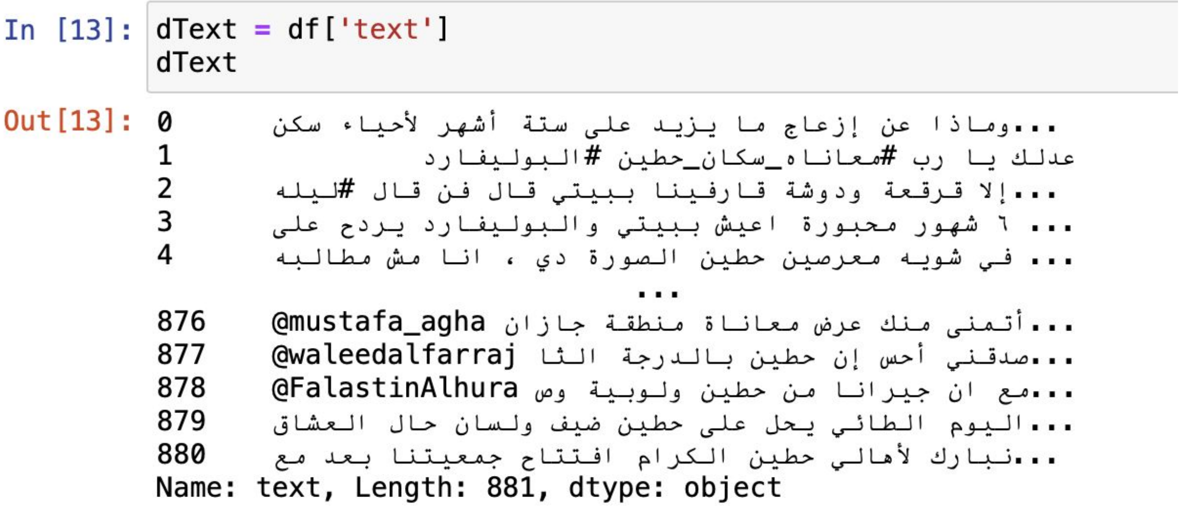
The next step was to upload the CSV file in *Jupyter Notebook* and load the data in a dataframe, therefore splitting them into different columns based on what we need:



Out[9]:

	date	username	to	replies	retweets	favorites	text	geo	mentions	hashtags	id	
0	08/02/2020 9:16	SsE1j8MeFSuOZY9	sabqorg	0	0	0	وماذا عن إزعاج ما يزيد على ستة أشهر لأحياء سكن...	NaN	NaN	NaN	1.226070e+18	https://twitter.com/SsE1j8MeFSuC
1	04/02/2020 22:03	Bndr888_	NaN	0	0	0	عدلك يا رب #معاناه_سكان_حطين #البوليفارد	NaN	NaN	NaN	1.224820e+18	https://twitter.com/Bndr888_/sta
2	24/01/2020 20:02	Bndr888_	NaN	1	0	0	إلا قرقعة ودوشة قارفينا ببيتي قال فن قال #ليله...	NaN	NaN	NaN	1.220800e+18	https://twitter.com/Bndr888_/sta
3	21/01/2020 21:03	Bndr888_	NaN	1	0	0	٦ شهور محبورة اعيش ببيتي والبوليفارد يردح على ...	NaN	NaN	NaN	1.219730e+18	https://twitter.com/Bndr888_/sta
4	19/01/2020 21:59	Gahed0	NaN	0	0	0	في شويه معرصين حطين الصورة دي ، انا مش مطالبه ...	NaN	NaN	NaN	1.219020e+18	https://twitter.com/Gahed0/status

Figure 2: HittenTweets.csv as a dataframe

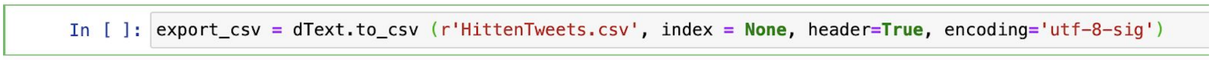


```
In [13]: dText = df['text']
         dText

Out[13]: 0      ...وماذا عن إزعاج ما يزيد على ستة أشهر لأحياء سكن
         1             عدلك يا رب #معاناه_سكان_حطين #البوليفارد
         2      ...إلا قرقعة ودوشة قارفينا ببيتي قال فن قال #ليله
         3      ... ٦ شهور محبورة اعيش ببيتي والبوليفارد يردح على
         4      ... في شويه معرصين حطين الصورة دي ، انا مش مطالبه
                                     ...
         876    @mustafa_agha ...أتمنى منك عرض معاناة منطقة جازان
         877    @waleedalfarraj ...صدقني أحس إن حطين بالدرجة الثا
         878    @FalastinAlhura ...مع ان جيرانا من حطين ولوبية وص
         879    ...اليوم الطائي يحل على حطين ضيف ولسان حال العشاق
         880    ...نبارك لأهالي حطين الكرام افتتاح جمعيتنا بعد مع
         Name: text, Length: 881, dtype: object
```

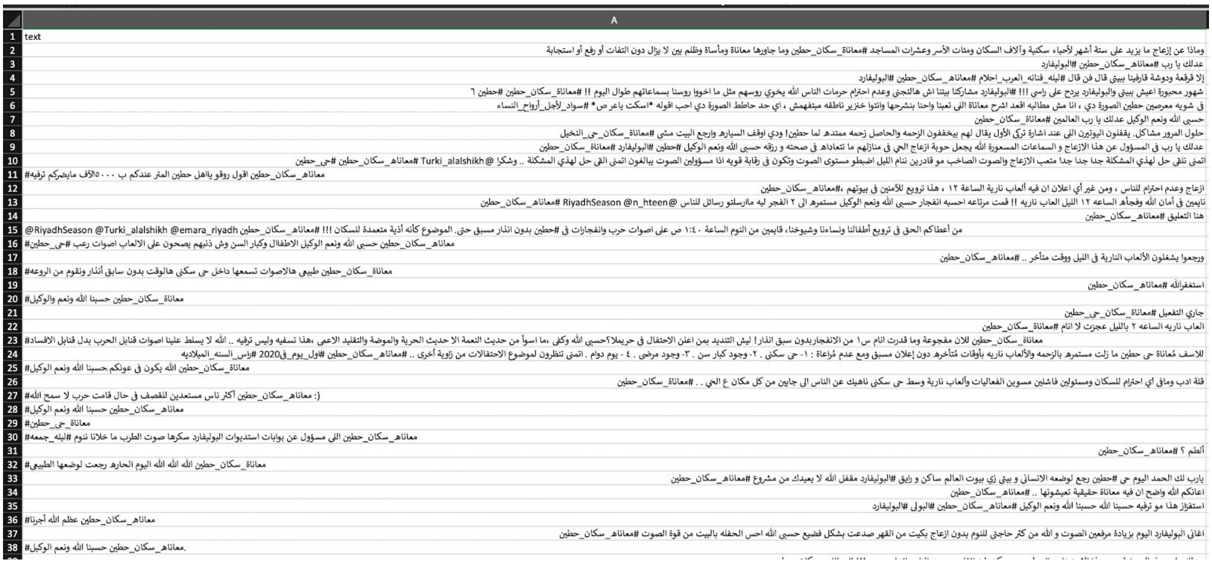
Figure 3: displaying (Text) column dataframe

After saving the tweets' texts in another data frame, we have exported it to a CSV file:



```
In [ ]: export_csv = dText.to_csv (r'HittenTweets.csv', index = None, header=True, encoding='utf-8-sig')
```

Figure 4: Exporting dataframe to HittenTweets.CSV



	A
1	text
2	وماذا عن إزعاج ما يزيد على ستة أشهر لأحياء سكنية وآلاف السكان ومئات الأسر وعشرات المساجد #معاناة_سكان_حطين وما جاورها معاناة ومأساة وظلم بين لا يزال دون التفات أو رفع أو استجابة
3	عدلك يا رب #معاناه_سكان_حطين #البوليفارد
4	إلا قرقعة ودوشة قارفينا ببيتي قال فن قال #ليله_فنانه_العرب_احلام #معاناه_سكان_حطين #البوليفارد
5	شهور محبورة اعيش ببيتي والبوليفارد يردح على راسي !!! #البوليفارد مشاركنا بيتنا اش هالتجني وعدم احترام حرمات الناس الله يخوي روسهم مثل ما اخووا روسنا بسماعاتهم طوال اليوم !! #معاناة_سكان_حطين #حطين ٦
6	في شويه معرصين حطين الصورة دي ، انا مش مطالبه اقعد اشرح معاناة اللى تعبنا واحنا بنشرحها وانتوا خنزير ناطقه مبتفهمش ، اي حد حاطط الصورة دي احب اقوله *اسكت ياعر ص* #سواد_لأجل_أرواح_النساء
7	حسبي الله ونعم الوكيل عدلك يا رب العالمين #معاناة_سكان_حطين
8	حلول المرور مشاكل. يقفلون اليوتيرن اللى عند اشارة تركي الأول يقال لهم بيخففون الزحمه والحاصل زحمه ممتده لما حطين! ودي اوقف السياره وارجع البيت مشي #معاناة_سكان_حي_النخيل
9	عدلك يا رب في المسؤول عن هذا الازعاج و السماعات المسعورة الله يجعل حوبة ازعاج الحي في منازلهم ما تتعداه في صحته و رزقه حسبي الله ونعم الوكيل #حطين #البوليفارد #معاناة_سكان_حطين
10	اتمنى نلقى حل لهذي المشكلة جدا جدا جدا متعب الازعاج والصوت الصاخب مو قادرين ننام الليل اضبطو مستوى الصوت وتكون في رقابة قويه اذا مسؤولين الصوت يبالغون اتمنى القى حل لهذي المشكلة .. وشكرا @Turki_alalshikh #معاناه_سكان_حطين #حي_حطين
11	معاناه_سكان_حطين اقول روقو يااهل حطين المتر عندكم ب ٥٠٠٠الآف مايضركم ترفيه#
12	ازعاج وعدم احترام للناس ، ومن غير أي اعلان ان فيه ألعاب نارية الساعة ١٢ ، هذا ترويع للآمنين في بيوتهم ،#معاناه_سكان_حطين
13	نايمين في أمان الله وفجأه الساعه ١٢ الليل العاب ناريه !! قمت مرتاعه احسبه انفجار حسبي الله ونعم الوكيل مستمره الى ٢ الفجر ليه ماارسلتو رسائل للناس @n_hteen@ RiyadhSeason #معاناه_سكان_حطين
14	هنا التعليق #معاناه_سكان_حطين
15	من أعطاكم الحق في ترويع أطفالنا ونساءنا وشيوخنا، قايمين من النوم الساعة ١:٤٠ ص على اصوات حرب وانفجارات في #حطين بدون انذار مسبق حتى. الموضوع كأنه أذية متعمدة للسكان !!! #معاناه_سكان_حطين @emara_riyadh @Turki_alalshikh @RiyadhSeason
16	معاناه_سكان_حطين حسبي الله ونعم الوكيل الاطفاال وكبار السن وش ذنبهم يصحون على الالعاب اصوات رعب #حي_حطين#
17	ورجعوا يشغلون الألعاب النارية في الليل ووقت متأخر .. #معاناه_سكان_حطين
18	معاناة_سكان_حطين طبيعي هالاصوات تسمعها داخل حي سكني هالوقت بدون سابق أنذار ونقوم من الروعه#
19	استغفرالله #معاناه_سكان_حطين
20	معاناة_سكان_حطين حسبنا الله ونعم والوكيل#
21	جاري التفعيل #معاناة_سكان_حي_حطين
22	العاب ناريه الساعه ٢ بالليل عجزت لا انام #معاناة_سكان_حطين
23	معاناة_سكان_حطين للان مفجوعة وما قدرت انام س١ من الانفجاربدون سبق انذار! ليش التنديد بمن اعلن الاحتفال في حريملا؟حسبي الله وكفى ،ما اسوأ من حديث النعمة الا حديث الحرية والموضة والتقليد الاعمى ،هذا تسفيه وليس ترفيه .. الله لا يسلط علينا اصوات قنابل الحرب بدل قنابل الافساد#
24	للاسف مُعاناة حي حطين ما زلت مستمره بالزحمه والألعاب ناريه بأوقات مُتأخره دون إعلان مسبق ومع عدم مُراعاة : ١- حي سكني . ٢- وجود كبار سن . ٣- وجود مرضى . ٤ - يوم دوام . اتمنى تنظرون لموضوع الاحتفالات من زاوية أخرى .. #معاناه_سكان_حطين #اول_يوم_في2020 #راس_السنه_الميلاديه
25	معاناة_سكان_حطين الله يكون في عونكم حسبنا الله ونعم الوكيل#
26	قلة ادب وماقي اي احترام للسكان ومسئولين فاشلين مسوين الفعاليات وألعاب نارية وسط حي سكني ناهيك عن الناس الي جايين من كل مكان ع الحي . . #معاناة_سكان_حطين
27	(: معاناه_سكان_حطين أكثر ناس مستعدين للقصف في حال قامت حرب لا سمح الله#
28	معاناه_سكان_حطين حسبنا الله ونعم الوكيل#
29	معاناة_حي_حطين#
30	معاناه_سكان_حطين اللي مسؤول عن بوابات استديوات البوليفارد سكرها صوت الطرب ما خلانا ننوم #ليله_جمعه#
31	ألطم ؟ #معاناه_سكان_حطين
32	معاناة_سكان_حطين الله الله الله اليوم الحاره رجعت لوضعها الطبيعي#
33	يارب لك الحمد اليوم حي #حطين رجع لوضعه الانساني و بيتي زي بيوت العالم ساكن و رايق #البوليفارد مقفل الله لا يعيدك من مشروع #معاناه_سكان_حطين
34	اعانكم الله واضح ان فيه معاناة حقيقية تعيشونها .. #معاناه_سكان_حطين
35	استفزاز هذا مو ترفيه حسبنا الله حسبنا الله ونعم الوكيل #معاناه_سكان_حطين #البولي #البوليفارد
36	معاناه_سكان_حطين عظم الله أجرنا#
37	اغاني البوليفارد اليوم بزيادة مرفعين الصوت و الله من كثر حاجتي للنوم بدون ازعاج بكيت من القهر صدعت بشكل فضيع حسبي الله احس الحفله بالبيت من قوة الصوت #معاناه_سكان_حطين
38	.معاناه_سكان_حطين حسبنا الله ونعم الوكيل#

Figure 5: HittenTweets.CSV

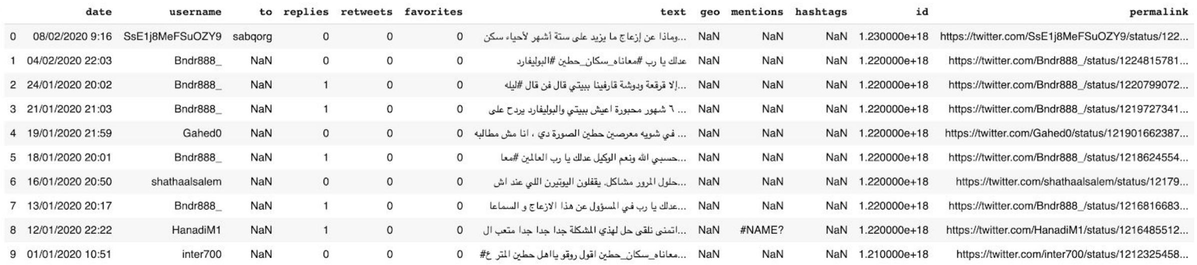
3. Data preprocessing

3.1 Exploring data:

In order to prepare the data for the analysis phase, the first step is to explore and assess the tweets we have fetched by using **Pandas** to identify any tidiness or quality issues to consider in the data cleaning. Below, we show some of the exploration we did, the full code is in `[SWE485] Hitten Tweets - Phase2.ipynb`:

· Displaying a sample bunch of records to assess them visually (Fig. 6).

```
df.head(10)
```

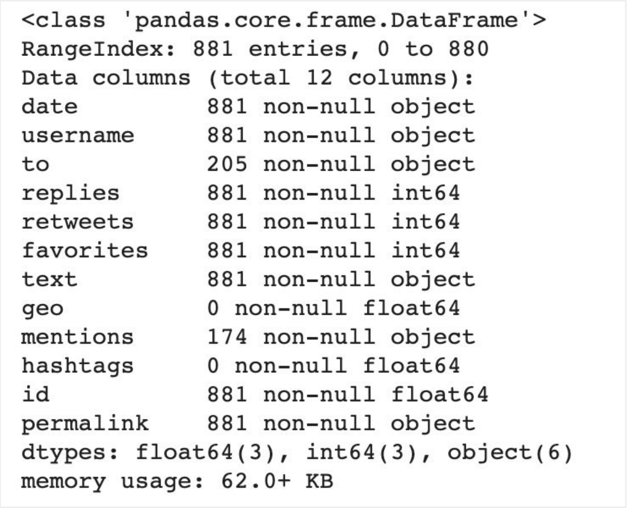


	date	username	to	replies	retweets	favorites	text	geo	mentions	hashtags	id	permalink
0	08/02/2020 9:16	SsE1j8MeFSuOZY9	sabqorg	0	0	0	وماذا عن إزعاج ما يزيد على ستة أشهر لأحياء سكن...	NaN	NaN	NaN	1.230000e+18	https://twitter.com/SsE1j8MeFSuOZY9/status/122...
1	04/02/2020 22:03	Bndr888_	NaN	0	0	0	عدلك يا رب #معاناه_سكان_حطين #البوليفارد	NaN	NaN	NaN	1.220000e+18	https://twitter.com/Bndr888_/status/1224815781...
2	24/01/2020 20:02	Bndr888_	NaN	1	0	0	إلا قرقعة ودوشة قارفينا ببيتي قال فن قال #ليله...	NaN	NaN	NaN	1.220000e+18	https://twitter.com/Bndr888_/status/1220799072...
3	21/01/2020 21:03	Bndr888_	NaN	1	0	0	٦ شهور محبورة اعيش ببيتي والبوليفارد يردح على ...	NaN	NaN	NaN	1.220000e+18	https://twitter.com/Bndr888_/status/1219727341...
4	19/01/2020 21:59	Gahed0	NaN	0	0	0	في شويه معرصين حطين الصورة دي ، انا مش مطالبه ...	NaN	NaN	NaN	1.220000e+18	https://twitter.com/Gahed0/status/121901662387...
5	18/01/2020 20:01	Bndr888_	NaN	1	0	0	حسبي الله ونعم الوكيل عدلك يا رب العالمين #معا...	NaN	NaN	NaN	1.220000e+18	https://twitter.com/Bndr888_/status/1218624554...
6	16/01/2020 20:50	shathaalsalem	NaN	0	0	0	حلول المرور مشاكل. يقفلون اليوتيرن اللي عند اش...	NaN	NaN	NaN	1.220000e+18	https://twitter.com/shathaalsalem/status/12179...
7	13/01/2020 20:17	Bndr888_	NaN	1	0	0	عدلك يا رب في المسؤول عن هذا الازعاج و السماعا...	NaN	NaN	NaN	1.220000e+18	https://twitter.com/Bndr888_/status/1216816683...
8	12/01/2020 22:22	HanadiM1	NaN	1	0	0	اتمنى نلقى حل لهذي المشكلة جدا جدا جدا متعب ال...	NaN	#NAME?	NaN	1.220000e+18	https://twitter.com/HanadiM1/status/1216485512...
9	01/01/2020 10:51	inter700	NaN	0	0	0	#معاناه_سكان_حطين اقول روقو ياهل حطين المتر ع...	NaN	NaN	NaN	1.210000e+18	https://twitter.com/inter700/status/1212325458...

Figure 6: Scanning the first 10 records

· Displaying the detailed columns specifications (Fig. 7)

```
df.info()
```



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 881 entries, 0 to 880
Data columns (total 12 columns):
date         881 non-null object
username     881 non-null object
to           205 non-null object
replies      881 non-null int64
retweets     881 non-null int64
favorites    881 non-null int64
text         881 non-null object
geo          0 non-null float64
mentions     174 non-null object
hashtags     0 non-null float64
id           881 non-null float64
permalink    881 non-null object
dtypes: float64(3), int64(3), object(6)
memory usage: 62.0+ KB
```

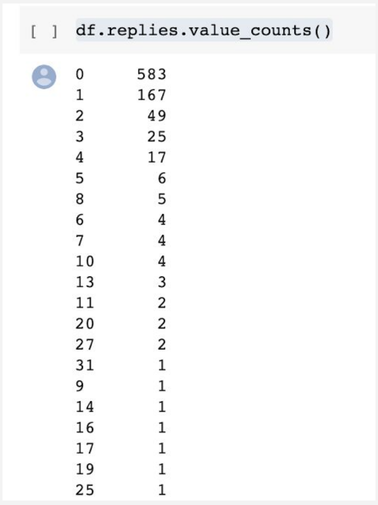
Figure 7: Columns' specifications

- Checking if there are any duplicate records.

```
print(sum(df.duplicated()))
Result: 0
```

- Displaying number of retweets, replies and likes in (Fig. 8).
  The number of replies, retweets and favorites, will help us assess the quality of the way used to gather the tweets. It is shown that the hashtag is active and people use it frequently not only by the number of tweets, but also the replies, retweets and favorites.

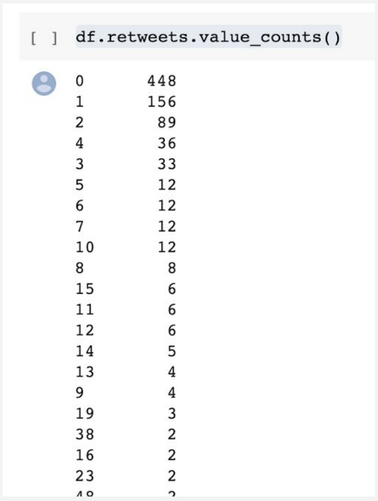
`df.replies.value_counts()` `df.retweets.value_counts()` `df.favorites.value_counts()`



```
[ ] df.replies.value_counts()

0      583
1      167
2       49
3       25
4       17
5        6
8        5
6        4
7        4
10       4
13       3
11       2
20       2
27       2
31       1
9        1
14       1
16       1
17       1
19       1
25       1
```

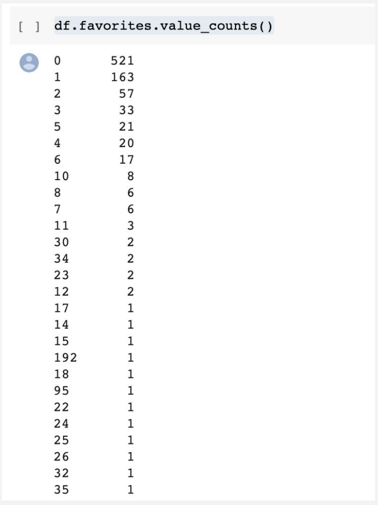
Figure A



```
[ ] df.retweets.value_counts()

0      448
1      156
2       89
4       36
3       33
5       12
6       12
7       12
10      12
8        8
15       6
11       6
12       6
14       5
13       4
9        4
19       3
38       2
16       2
23       2
```

Figure B



```
[ ] df.favorites.value_counts()

0      521
1      163
2       57
3       33
5       21
4       20
6       17
10       8
8        6
7        6
11       3
30       2
34       2
23       2
12       2
17       1
14       1
15       1
192      1
18       1
95       1
22       1
24       1
25       1
26       1
32       1
35       1
```

Figure C

Figure 8: Tweet's engagements

3.2 Quality and tidiness issues:

After exploring and assessing the data, a number of issues regarding the quality and the tidiness of data were clearly noticed.

In the following table (Table 1), we present seven of the issues. Categorized to **quality** and **tidiness** issues, along with the explanation of the used way/tool to clean them.

	Issue	**Fix**	**Library/tool**
1	Date column's type is not Timestamp, it can not be used to follow the timeline of the tweets and opinions.	Convert its type to datetime: `pd.to_datetime()`	Pandas

2	Some Arabic Hashtag in the tweet will affect the classification, since they contain words indicating negative contents.	Remove arabic hashtags:
		`df_clean['text'].str.replace`
		`("#[\\\p{أ-ي}_]+",' ')`
			Python "`replace()`" method with Regular Expressions

3	Punctuation	Remove punctuation:
		`df_clean['text'].str.replace`
		`('[^\w\s]','')`
			Python "`replace()`" method with Regular Expressions

4	Normalizing text - letters replacement.	Replacing some letters with others:
		`df_clean['text'].str.replace`
		`('أ','ا')`
		`df_clean['text'].str.replace`
		`('إ','ا')`
		`df_clean['text'].str.replace`
		`('ة','ه')`
			Python "`replace()`" method

5	Mentions.	Removing mentions:
		`df_clean['text'].str.replace`
		`('@[^\s]+',' ')`
			Python "`replace()`" method with Regular Expressions

6	Date column includes day and time together.	Splitting date column into day and time:
		`dt.strftime('%d/%m/%Y')`
		`dt.strftime('%H:%M')`
			Pandas

7	Some data are not needed in the analysis	Dropping the columns that are irrelevant
		`df_clean.drop('geo', 1)`
		`df_clean.drop('mentions', 1)`
		`df_clean.drop('hashtags', 1)`
		`df_clean.drop('id', 1)`
		`df_clean.drop('to', 1)`
		`df_clean.drop('permalink', 1)`
		`df_clean.drop('date', 1)`
		`df_clean.drop('username', 1)`
			Pandas

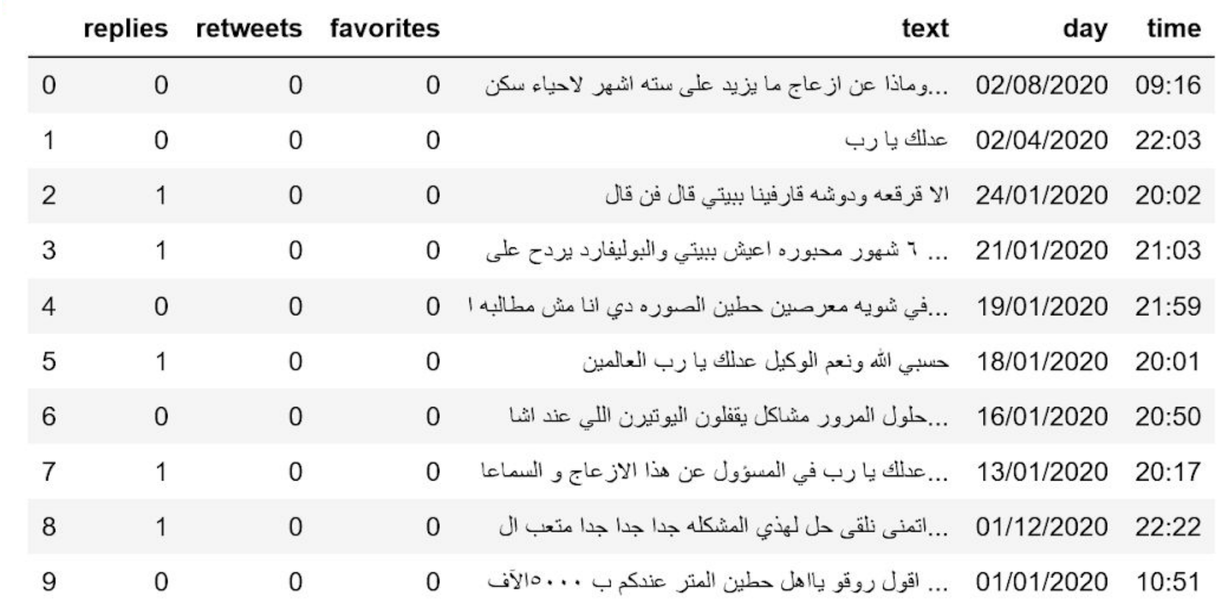
Table 1: Quality and tidiness issues.

3.3 Data after cleaning:

The data are assessed and cleaned from the issues mentioned in section 3.2 as shown in (Fig. 9), where the file contains the day and time, tweet as a text, number of replies, retweets and favorites, which will assist us in meeting the objectives from this analysis mentioned in section 1.2 when building our model.

Saving the number of replies, retweets and favorites, will help us in our model computations and visualization, such as the most retweeted tweet.

Saving the date and time of the tweets separately will help us in identifying the specific time the neighborhood was trending and active, to investigate whether it was related to a specific event or not (e.g. fireworks, concert).



	replies	retweets	favorites	text	day	time
0	0	0	0	...وماذا عن ازعاج ما يزيد على سته اشهر لاحياء سكن	02/08/2020	09:16
1	0	0	0	عدلك يا رب	02/04/2020	22:03
2	1	0	0	الا قرقعه ودوشه قارفينا ببيتي قال فن قال	24/01/2020	20:02
3	1	0	0	... ٦ شهور محبوره اعيش ببيتي والبوليفارد يردح على	21/01/2020	21:03
4	0	0	0	...في شويه معرصين حطين الصوره دي انا مش مطالبه ا	19/01/2020	21:59
5	1	0	0	حسبي الله ونعم الوكيل عدلك يا رب العالمين	18/01/2020	20:01
6	0	0	0	...حلول المرور مشاكل يقفلون اليوتيرن اللي عند اشا	16/01/2020	20:50
7	1	0	0	...عدلك يا رب في المسؤول عن هذا الازعاج و السماعا	13/01/2020	20:17
8	1	0	0	...اتمنى نلقى حل لهذي المشكله جدا جدا جدا متعب ال	01/12/2020	22:22
9	0	0	0	... اقول روقو يااهل حطين المتر عندكم ب ٥٠٠٠الآف	01/01/2020	10:51

Figure 9: Screenshot of HittenTweetsCleaned.csv

4. Data analysis and Modeling

4.1 Introduction:

In this phase, we are going to apply some preprocessing techniques to help us increase the descriptive and predictive analytics accuracy. We will choose our model based on the higher accuracy between the three data mining techniques: Linear Regression, Logistic Regression and Naive Bayes.

4.2 Preprocessing

In the table below, we view some more issues that needed to be preprocessed before starting the analysis step, such as removing duplicated letters, in order to increase our model's accuracy result.

	Issue	Fix	Library/tool
1	Removing stopwords requires tokenizing each tweet first.	Tokenize tweets: `word_list = row.split()`	Python `split()` method

2	Some tweets contain stopwords that may affect the accuracy of the results.	Remove Arabic stopwords from the tweets:
		`word_list = [word for word in`
		`word_list if word not in ar_stops]`
			nltk

3	Remove more Punctuation	Combine Arabic and English punctuation and remove them:
		`Tweet.str.maketrans('','',punctuatio`
		`ns_list)`
			Python "`replace()`" method with Regular Expressions

4	Normalizing text requires detokenize the tweets.	`def listToString(s):`	Building a customized method.

5	Normalizing text - letters replacement. [2]	Replacing some letters with others:
		`str.replace('ا','أ')`
		`str.replace('ا','إ')`
		`str.replace('ه','ة')`
		`re.sub("[ا" ,"[اآأإ", text)`
		`re.sub("ي" ,"ى", text)`
		`re.sub("ه" ,"ة", text)`
		`re.sub("ك" ,"گ", text)`
			Python "`replace()`" method and Regular Expressions.

[2] This step was done in the last phase, we had to shift it to this phase since for example, the stopword (إن) won't be detected because it will be (ان) due to the problem of normalizing the tweets before removing the stopwords from it. We also added more letters to replace.

6	Repeated characters.	`re.sub(r'(.)\1+', r'\1', text)`	Python “`replace()`” method with Regular Expressions

7	Some tweets contain just tab spaces and newlines after deleting mentions and hashtags.	Remove new lines and tab spaces with white space:
		`tweet.replace('\n',' ')`
		`tweet.replace('\t',' ')`
			Python “`replace()`” method.

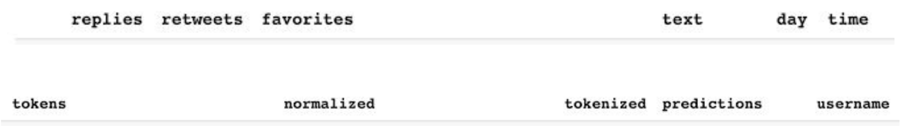
8	Some tweets *only* contain spaces.	Removing them from the normalized column in the dataframe: `df[df.normalized != '']`	Use not equal (`!=`) operator to the column in the dataframe.
9	Some tweets contain media URLs.	re.sub(`'((www\.[^\s]+)|(https?://[^\s]+))',' '`,tweet)	Using regular expressions.

Table 2: Second issues cleansing.

4.3 Descriptive analysis:

As a preceding step to predictive analysis, a descriptive analysis should be done to give a deeper understanding of the associations among attributes.

The first step is to visually present the tweets using **Pandas** library. And since the available data are the listed attributes in the following figure 4.3.1, different questions have been asked to discover interesting findings from them.



```
        replies  retweets  favorites                          text        day  time

tokens                    normalized                  tokenized  predictions      username
```

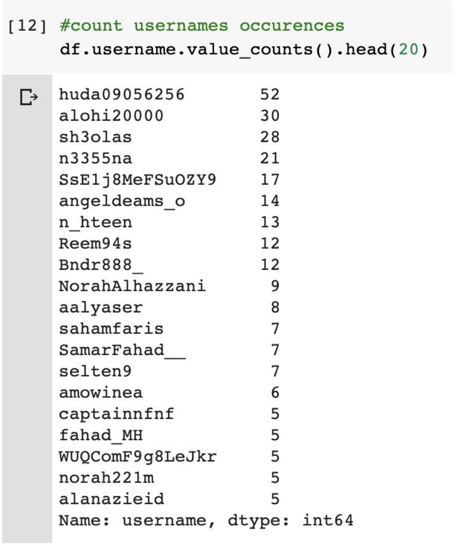
(Figure 4.3.1 Dataframe head)

A tweet can be seen from different perspectives, the text itself, its sentiment, the timing it has been posted at, its author and the engagements associated with it.

Now, we will list each **analysis question along with its result**, those results have been extracted using different python and Pandas library methods, such as `.value_counts(), .sort_values(), .group_by(), and mean()`.

1. Starting from the ones who are behind all of this, the authors.

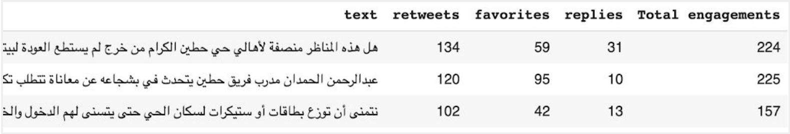
Question	Code and result

Is there anyone who significantly **tweets more frequently** than others?	
	[12] `#count usernames occurences`
	`df.username.value_counts().head(20)`
	`huda09056256       52`
	`alohi20000         30`
	`sh3olas            28`
	`n3355na            21`
	`SsE1j8MeFSuOZY9    17`
	`angeldeams_o       14`
	`n_hteen            13`
	`Reem94s            12`
	`Bndr888_           12`
	`NorahAlhazzani      9`
	`aalyaser            8`
	`sahamfaris          7`
	`SamarFahad__        7`
	`selten9             7`
	`amowinea            6`
	`captainnfnf         5`
	`fahad_MH            5`
	`WUQComF9g8LeJkr     5`
	`norah221m           5`
	`alanazieid          5`
	`Name: username, dtype: int64`
	(Figure 4.3.2 Top 20 author's tweeting frequency)

Comment	It's clear that the tweeper with the username (**@huda09056256**) has been tweeting more than others about the issue.

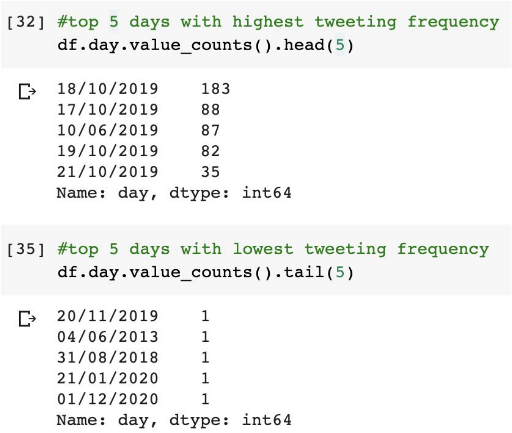
2. Moving to the popularity of the tweets.

<table>
<tr><th>Question</th><th>Code and result</th></tr>
<tr><td>Which tweet has got the <b>highest engagement</b>?</td><td>#Calculate total engagements then sort them desc.
df['Total engagements']=
df['retweets']+df['favorites']+df['replies']
df[['text','retweets','favorites','replies','Total engagements']].sort_values(by=['retweets'],ascending=False).head(5)


<table>
<tr><th>text</th><th>retweets</th><th>favorites</th><th>replies</th><th>Total engagements</th></tr>
<tr><td>هل هذه المناظر منصفة لأهالي حي حطين الكرام من خرج لم يستطع العودة لبيت</td><td>134</td><td>59</td><td>31</td><td>224</td></tr>
<tr><td>عبدالرحمن الحمدان مدرب فريق حطين يتحدث في بشجاعه عن معاناة تتطلب تك</td><td>120</td><td>95</td><td>10</td><td>225</td></tr>
<tr><td>نتمنى أن توزع بطاقات أو ستيكرات لسكان الحي حتى يتسنى لهم الدخول والخ</td><td>102</td><td>42</td><td>13</td><td>157</td></tr>
</table>
(Figure 4.3.3 Most popular tweets)</td></tr>
<tr><td><b>Comment</b></td><td>The tweet with the highest engagements number is:
"هل هذه المناظر منصفة لأهالي حي حطين الكرام من خرج لم يستطع العودة لبيته ومن يريد الخروج لموعد لا يستطيع حتى وإن كانت حالة حرجة أو طارئة"</td></tr>
</table>

3. Exploring it from the day and time perspectives.

3.1. Day:

<table>
<tr><th>Question</th><th>Code and result</th></tr>
<tr><td>In which <b>days more tweets were posted</b> regarding the issue? And those with low tweeting rates.</td><td>
[32] #top 5 days with highest tweeting frequency
df.day.value_counts().head(5)

18/10/2019   183
17/10/2019   88
10/06/2019   87
19/10/2019   82
21/10/2019   35
Name: day, dtype: int64

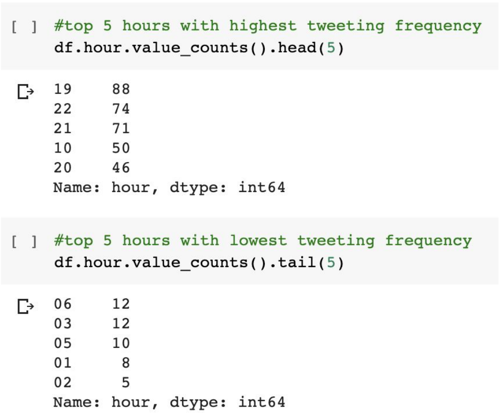
[35] #top 5 days with lowest tweeting frequency
df.day.value_counts().tail(5)

20/11/2019   1
04/06/2013   1
31/08/2018   1
21/01/2020   1
01/12/2020   1
Name: day, dtype: int64

(Figure 4.3.4 Tweeting rate/day)</td></tr>
<tr><td><b>Comment</b></td><td>- 18th of October (i.e. second day of Riyadh Season), scored the highest tweeting frequency, when the residents noticed the unusual traffic in their district.
- None of the days with low tweeting rates were in October 2019 which means that the topic has been trending throughout the month.</td></tr>
</table>

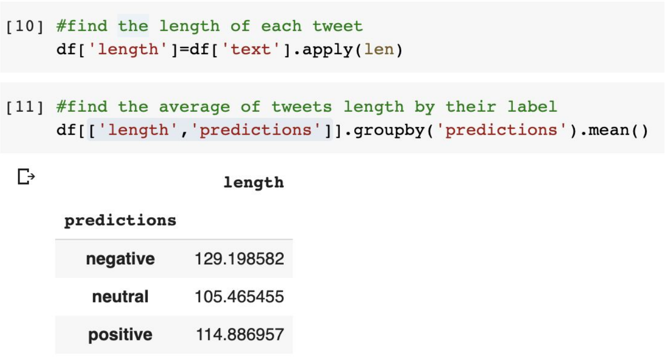
3.2. Time:

Question	Code and result

At which **hours more tweets were posted** regarding the issue? And those with low tweeting rates.	
	[ ] #top 5 hours with highest tweeting frequency
	df.hour.value_counts().head(5)
	19 88
	22 74
	21 71
	10 50
	20 46
	Name: hour, dtype: int64
	[ ] #top 5 hours with lowest tweeting frequency
	df.hour.value_counts().tail(5)
	06 12
	03 12
	05 10
	01 8
	02 5
	Name: hour, dtype: int64
	(Figure 4.3.5 Tweeting rate/day)

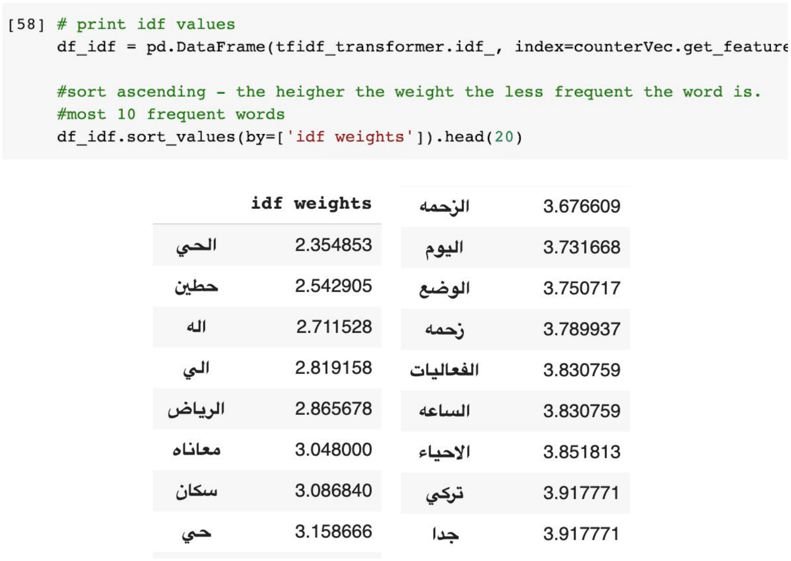
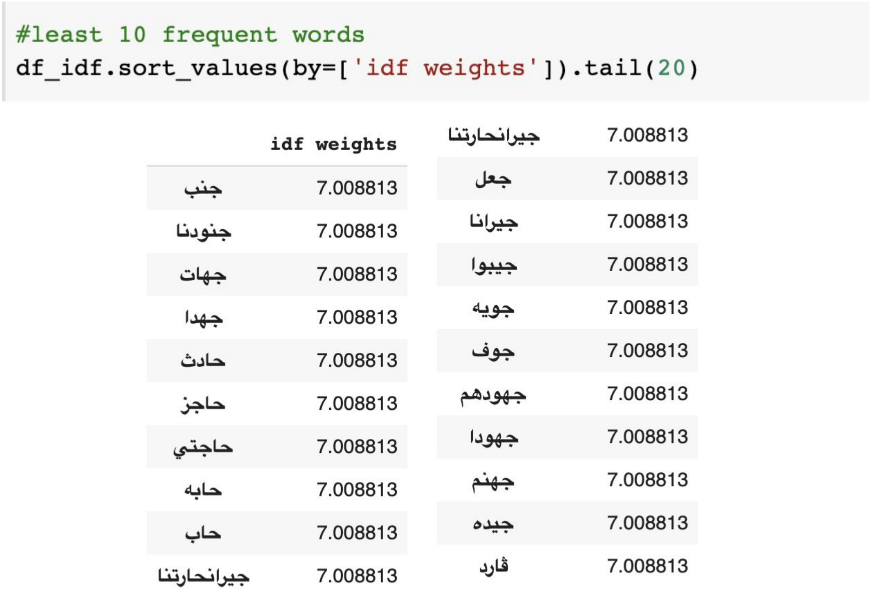
Comment	- Riyadh season's first event starts at 5 PM and finishes at 5 AM, the residents start tweeting more at 7 PM when the district possibly gets more crowded. - Residents tweet less after midnight.

4. Sentiment vs tweet length.

Question	Code and result
Does the **sentiment of the tweet affect the length** of it?	 [10] #find the length of each tweet df['length']=df['text'].apply(len) [11] #find the average of tweets length by their label df[['length','predictions']].groupby('predictions').mean() length predictions negative 129.198582 neutral 105.465455 positive 114.886957 (Figure 4.3.6 Tweets sentiment along with average sentiment)
Comment	Negative tweets are more likely to be longer than the other two types. However, the lengths mean values are close to each other.

5. Having the tweets tokenized, and using the IDF concept we can score word occurrences.

Using SciKit-learn library, the IDF of each word can be calculated using `.CountVectorizer` class that counts the words, and then `.TfidfTransformer` class to measure each word's IDF score. The higher the IDF the less frequent the words is.

Question	**Code and result**
What are the **most frequent words**? And those with **least frequency**.	  (Figure 4.3.6 Tweets sentiment along with average sentiment)
Comment	The most frequent words are strongly related to Hittin district traffic issues, while the least frequent ones are, obviously, not that related.

```
[58] # print idf values
     df_idf = pd.DataFrame(tfidf_transformer.idf_, index=counterVec.get_feature

     #sort ascending - the heigher the weight the less frequent the word is.
     #most 10 frequent words
     df_idf.sort_values(by=['idf weights']).head(20)
```

	idf weights
الحي	2.354853
حطين	2.542905
اله	2.711528
الي	2.819158
الرياض	2.865678
معاناه	3.048000
سكان	3.086840
حي	3.158666
الزحمة	3.676609
اليوم	3.731668
الوضع	3.750717
زحمه	3.789937
الفعاليات	3.830759
الساعه	3.830759
الاحياء	3.851813
تركي	3.917771
جدا	3.917771

```
#least 10 frequent words
df_idf.sort_values(by=['idf weights']).tail(20)
```

	idf weights
جنب	7.008813
جنودنا	7.008813
جهات	7.008813
جهدا	7.008813
حادث	7.008813
حاجز	7.008813
حاجتي	7.008813
حابه	7.008813
حاب	7.008813
جيرانحارتنا	7.008813
جيرانحارتنا	7.008813
جعل	7.008813
جيرانا	7.008813
جيبوا	7.008813
جويه	7.008813
جوف	7.008813
جهودهم	7.008813
جهودا	7.008813
جهنم	7.008813
جيده	7.008813
ڤارد	7.008813

After listing a number of descriptive analysis results, the full code can be found in `[SWE485] Phase 3 - Descriptive analysis.ipynb` file.

6. The relationship between the number of replies and retweets.

Linear Regression is a data mining method that focuses on discovering the relationship between an outcome and input, it outputs estimated input coefficients and finds how each one is relative to the outcome.

Although our next goal is predicting the tweets' sentiments, we wanted to try Linear Regression practically to understand the concept more. To apply it, we first started reshaping the number of retweets and replies using `array.reshape(-1, 1)` since our data has a single feature, and deleted neutral class.

We ran our model to find the relationship between the number of replies and retweets, splitting the data to 50% training and 50% testing. The result of the accuracy was **50%** with a coefficient of determination of **0.19**, which leads to a conclusion that there is none or a very weak relationship between the number of retweets and replies.

4.4 Predictive analysis:

After preprocessing the tweets, we have used Mazajak API to classify them under 3 classes: positive, negative and neutral. We are going to apply two different data mining methods to compare their accuracy and assess their performance based on different areas mentioned in 4.4.1 Evaluating the models, the steps in details are mentioned in the notebook for each model.

In order to answer the questions we had in the first phase, and meet our objectives (e.g analyze people’s sentiments), we will be applying Naive Bayes and Logistic Regression techniques to build a text classification model.

- **Naive Bayes:**

  Naive Bayes is a powerful data mining method used mostly for text classification, it is simpler than other techniques so we wanted to start with it and form an impression of the predictions for the data we have.

- **Logistic Regression:**

  Logistic regression is a machine learning algorithm that is usually conducted when the dependent variables are binary, implementing a predictive analysis. It is used to discover the relationship between an outcome and inputs.

To apply Logistic Regression and Naive Bayes, first, we had to convert the dependent variable data (Predictions column) to binary data (0 and 1), since Logistic Regression and Naive Bayes require the dependent variable to be binary. So, we first had to excluded the Neutral values,

```
data_df=data_df[data_df[‘predictions’]!= “neutral”]
```

Second, we have converted the data to numeric values (+ve = 1 and -ve =0).

```
data_df[‘predictions’]=data_df[‘predictions’].map({“positive”:1,“negative”:0})
```

Third, we started building the models by entering the training data on each model, we have used 0.33 of the data for testing and 0.67 for the training.

Logistic Regression:

```
classifier_log= LogisticRegression().fit(X_train,  y_train)
```

Naive Bayes:

```
classifier_nb= MultinomialNB().fit(X_train,  y_train)
```

After these steps, we have calculated the accuracy of the models using `score()` method and applying the Confusion Matrix concept on them by using `confusion_matrix()` method, which will be elaborated in the next section.

4.4.1 Evaluating the models:

We are going to choose our model comparing between Logistic Regression and Naive Bayes based on the confusion matrix (Number of true positive and true negative) with the accuracy, recall, precision, and the area under the curve.

Accuracy

Starting off with calculating the accuracy of both models using `score()` method, both Logistic Regression and Naive Bayes results were 77%. However, as we know, this is not an adequate metric to determine how efficient a model is, that is why we have chosen another criterion for comparison.

Precision, Recall and Confusion Matrix

Precision in both models was zero for positive. However, it was both 0.77 in Logistic Regression and Naive Bayes for negative class.

Given a high negative classification precision value is not enough since we do not find ALL relevant instances in the data, it only expresses the relevant data that was actually relevant. Another metric is needed in this case, which is Recall.

For both Logistic Regression and Naive Bayes, it is shown that Recall = 1 for the negative class, which means False Negative = 0, since 100% of the True Negative were discovered. While Recall = 0 for the positive class, which means there were no True Positive discovered as shown in the confusion matrix also in Figure 4.4.2.

The expected reason behind this problem is the fact that most of the tweets were complaints about the events in this neighborhood, and there was not enough positive data to train the model on.

Area Under the ROC Curve (AUC)

Another criteria of comparison is to measure the performance across all possible classification thresholds using the ROC Curve, by plotting True Positive Rate and False Positive Rate. We can then measure the entire 2-D area under the ROC curve to find the AUC value.

Both Logistic Regression AUC value is 0.78 which is considered excellent for both methods.

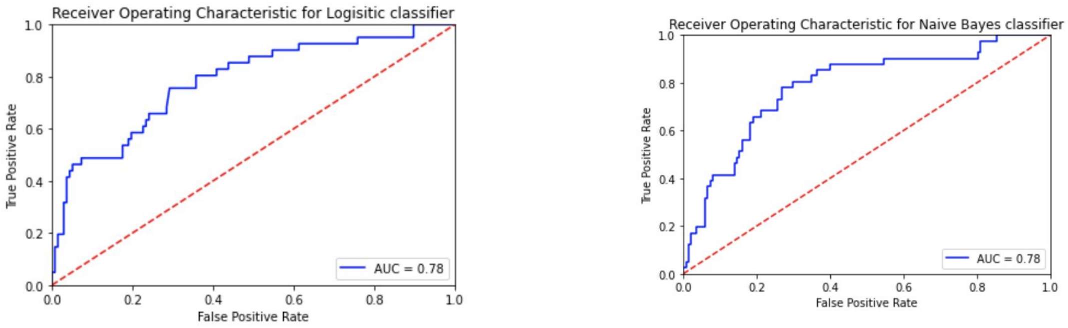


Figure A: Logistic Regression

Figure B: Naive Bayes

Figure 4.4.1: AUC Values

	Yes	No
Yes	137(TP)	0(FN)
No	41(FP)	0(TN)

Figure A: Logistic Regression

	Yes	No
Yes	137(TP)	0(FN)
No	41(FP)	0(TN)

Figure B: Naive Bayes

Figure 4.4.2: Confusion Matrix

The comparisons show that both Naive Bayes and Logistic Regression model have the same results based on the different aspects mentioned above. However, we decided to choose Naive Bayes since it is a much simpler data mining technique compared to Logistic Regression method, and we can build up other advanced techniques in the future based on it.

After our brief analysis above, we think our model might be a little biased, since most of the trained tweets were negative, this resulted in the model falling under **imbalanced classification problem**. Moreover, we have chosen a negative topic, so it is expected that there will be a lack of positive tweets.

5. Data Visualization and Findings

5.1 Introduction

We come to the end of this wonderful exploration journey, where all the interesting findings are translated into a visual context. Visual charts give the reader a quick and clear snapshot about the analysis phase, hence, the right choice of visualization type is a critical step.

In this section, we ran a brainstorming about the key findings in the descriptive analysis that need to be visualized, to make it more powerful and easily understandable.

5.2 Engagements, length and sentiments

We have been wondering before whether the sentiment type has a relationship with the tweet length or not, and the total number of engagements. So, we plotted the three attributes along with the posting date in a scatter diagram (Figure 5.2.1 A-B).
The **color** = sentiment, **position** = engagement and posting date, and **size**=tweet length.

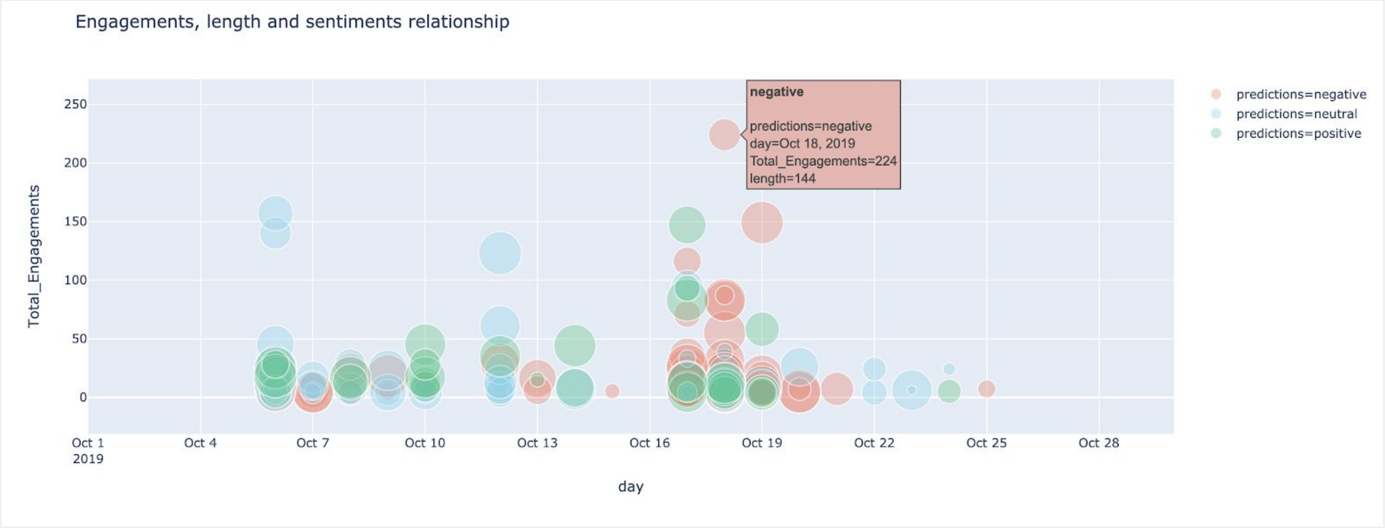


Figure 5.2.1 - A: Scatter diagram of the tweets length, engamanets, and sentiment over October.

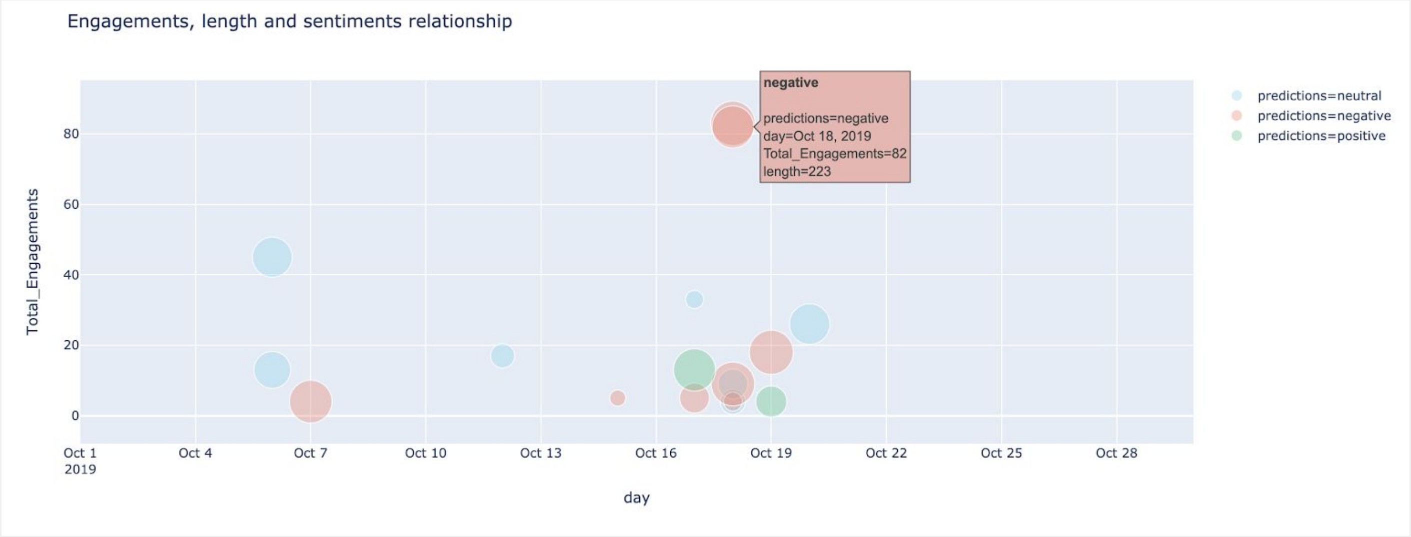


Figure 5.2.1 - B: 20 randomly sampled records are plotted.

Since the scatter graph answers the question **"Does the total engagement affect the length and the sentiment of the tweet?"** We can easily know that there is no clear relationship between the length and the engagements, as the circles are similar in size (=tweet length) and vary a lot with their engagements. However, it is noticeable that **negative tweets usually get more engagements** in our context (Riyadh season and Hittin residents tweets).

This graph was plotted using **Plotly** library.
Code:

```
df_engagements=df[df['Total_Engagements'] > 3]
df2=df_engagements.sample(20)
color_discrete_map = {'positive': 'rgba(49,196,133,0.5)', 'neutral':
'rgba(127, 211, 235,0.5)', 'negative': 'rgba(245, 126, 105,0.5)'}
data = px.scatter(df2,x='day', y='Total_Engagements',
range_x=['2019-10-01', '2019-10-30'], size='length',
color="predictions", hover_name="predictions", log_x=False,
size_max=30,color_discrete_map=color_discrete_map,
title="Engagements, length and sentiments relationship")data.show()
```

5.3 Time series

5.3.1 Tweeting frequency over time

The following chart answers the question “At which **hours more tweets were posted** regarding the issue? And those with low tweeting rates”. We can say that 7PM has the highest tweeting rate, which is actually true since it is the peak time of events. The lowest rate is at 2AM, considering that most of the events and restaurants are closed.

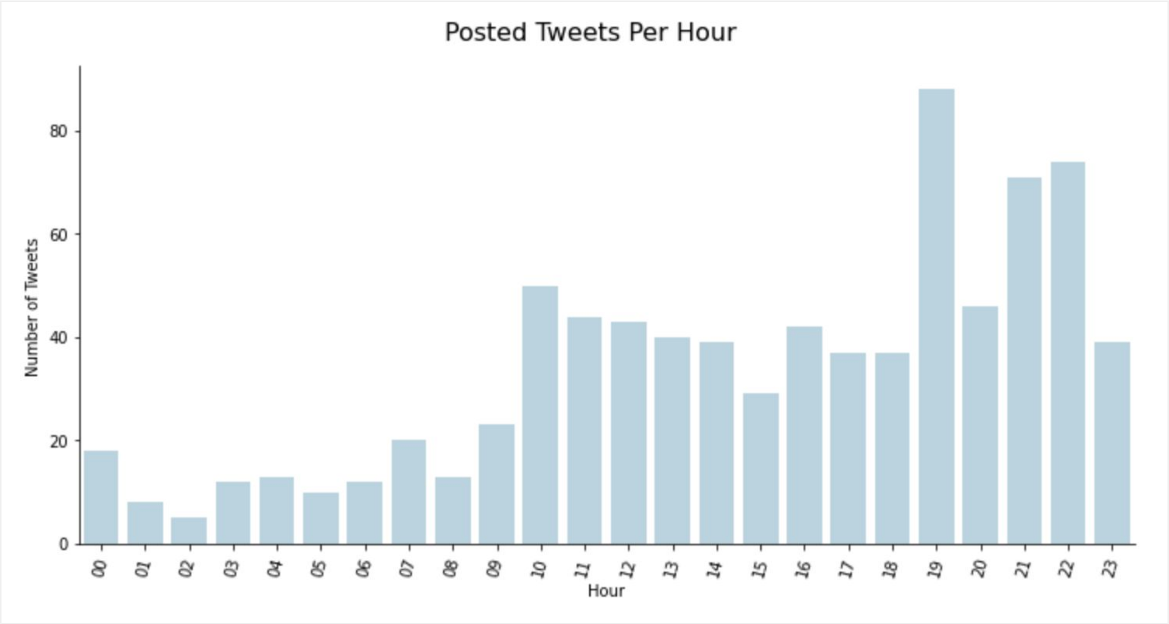


Figure 5.3.1: Tweeting frequency over time.

This graph was plotted using **Seaborn** library.

Code:

```
colors_blue = ["lightblue"]
graph = sb.catplot(data=df, x='hour', kind='count', sharey = False,
height=5, aspect=2, palette=sb.color_palette(colors_blue));
graph.set_axis_labels("Hour", "Number of Tweets")
graph.set_titles("{col_name}")
graph.fig.suptitle('Posted Tweets Per Hour', y=1.05, fontsize=16);
graph.set_xticklabels(rotation=75);
```

5.3.2 Tweeting frequency over time along with sentiments

Adding the sentiment attribute to the visualization in 5.3.2 gives it more depth:

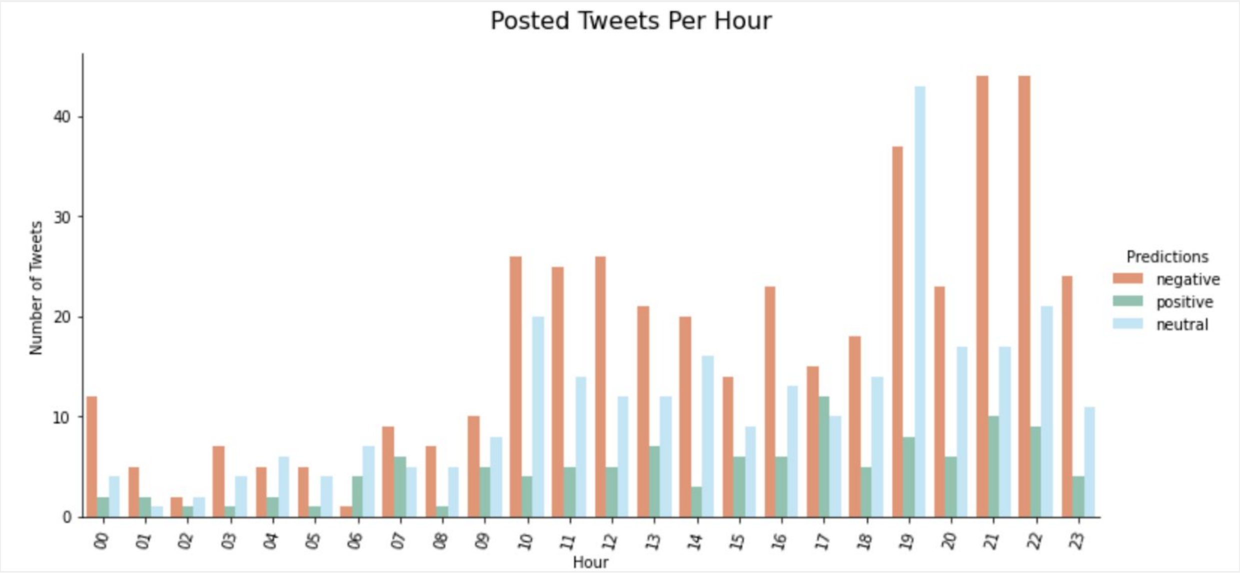


Figure 5.3.2: Tweeting frequency over time along with sentiments.

Although at 7 PM the tweeting frequency is at its highest, the majority of tweets were neutral ones. However, 9 PM and 10 PM have the highest negative tweeting frequency.

This graph was plotted using **Seaborn** library.

Code:

```
colors = ["#FF8A5C", "#7CCEB4", "#B3EBFF"]
df_predictions=df[df['predictions'] != 'neutral']
graph = sb.catplot(data=df, x='hour', hue="predictions", kind='count',
sharey = False, height=5, aspect=2, palette=sb.color_palette(colors));
graph.set_axis_labels("Hour", "Number of Tweets")
graph._legend.set_title('Predictions')
graph.set_titles("{col_name}")
graph.fig.suptitle('Posted Tweets Per Hour', y=1.05, fontsize=16,);
graph.set_xticklabels(rotation=75);
```

5.5 Tweeting frequency

We have also plotted the tweeper with **highest tweeting frequency**, it is better to distinguish the objective of the chart with different color than the rest of records:

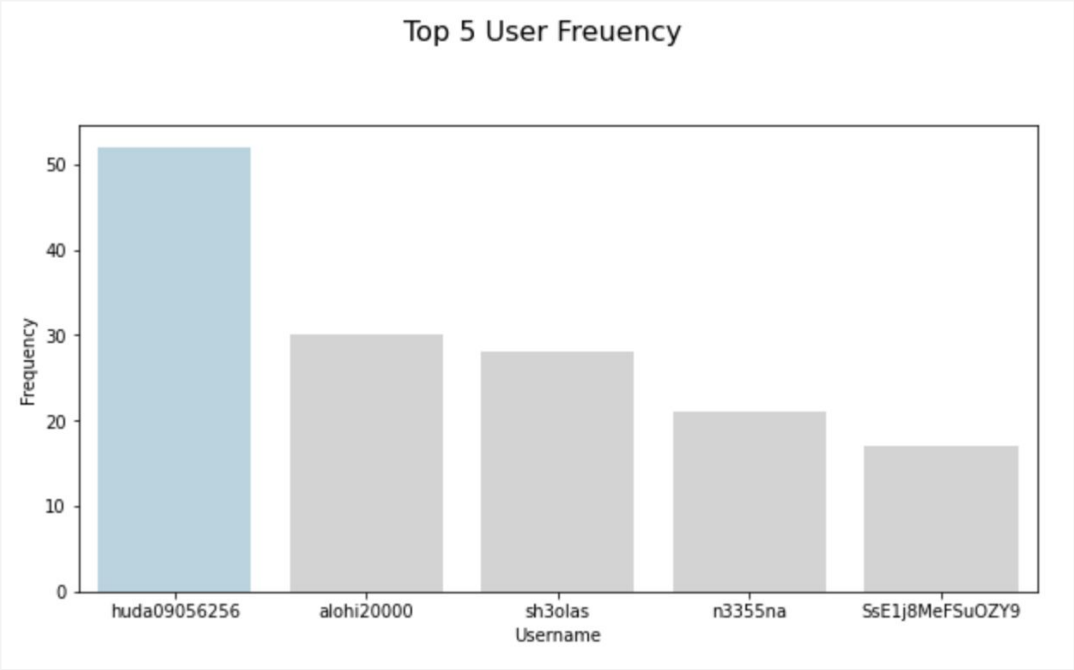


Figure 5.4: Top 5 tweepers with high tweeting frequency.

This graph was plotted using **Seaborn** library.

Code:

```
plt.figure(figsize = [10, 5])
clrs = ['lightgray' if (x < max(df.username.value_counts())) else
'lightblue' for x in df.username.value_counts() ]
graph = sb.countplot(data = df, x = "username", order =
df.username.value_counts().iloc[:5].index, palette=clrs);
graph.set_xlabel('Username');
graph.set_ylabel('Frequency');

plt.suptitle('Top 5 User Freuency', y=1.05, fontsize=16,
fontfamily='sans-serif');
```

4.5 Project files:

To explore the project clearly, the following table lists the file names along with their description and purpose.

	File name	Description
1	`HittenTweets.csv`	The retrieved records from `GetOldTweets3` before cleaning.
2	`[SWE485] Hitten Tweets - Phase1.ipynb`	The notebook contains the reading of the retrieved tweets from `GetOldTweets3` library and the initial exploring of data.
3	`[SWE485] Hitten Tweets - Phase2.ipynb`	The data preprocessing is documented in detail in this notebook.
4	`HittenTweetsCleaned.csv`	The cleaned records where any quality/tidiness issue was removed from.
5	`[SWE485] Phase 3 - Cleansing and tokenisation`	Tokenized and normalized records with sentiments from Mazajak API.
6	`[SWE485]Phase 3 - Descriptive analysis.ipynb`	The descriptive analysis performed is documented in this notebook
7	`[SWE485]Linear Regression.ipynb`	Linear Regression model is built and documented in this notebook
8	`[SWE485]Logistic Regression .ipynb`	Logistics Regression model is built and documented in this notebook
9	`[SWE485]Naive bayes.ipynb`	Naive bayes model is built and documented in this notebook
10	`HittenTweetsWithPredictions.csv`	The tokenized records with their predictions are stored in this file
11	`HittenTweetsPhase3WithoutPredictions.csv`	The tokenized records whiteout predictions are stored in this file.
12	`[SWE485] Phase 4- Descriptive analysis visualization.ipynb`	The visualization of the descriptive analysis is implemented in this notebook