Data Science
Final Project HIT

By:

Shani Elgamil

Daniel Dolberg

Prelude

Twitter היא מהרשתות החברתיות המובילות כיום, היא ידועה במערכת שלה המאפשרת פרסום מסרים קצרים של עד 280 תווים שבה כל אדם יכול להכנס ולהביע את הדעות ואו החוויות שלו.
הפלטפורמה לאט לאט הפכה ל"מרכז העיר הדיגיטלי שבו כולם באים לשמוע את החדשות האחרונות", ואפילו עם כל מיני מחלקות שהיו לאחרונה מספר המשמתמשים ממשיך לגדול.

בתוך טוויטר נמצאים אין ספור אמנים, אלפי פוליטיקאים ומליוני סלבריטאים.

התכונה שמושכת אנשים לפלטפורמה זה היכולת לפרסם לכל העולם את הדעות והמחשבות שלהם בעזרת מה שנקרא tweets ואו "ציוצים".

tweet הוא פשוט הגרסא של twitter לפוסט. זה יכול להיות טקסט, תמונה, סרטון ואו אפילו הצבעה.

לכל tweet אפשר לעשות מה שנקרא Retweets, זה כאשר משתמש רוצה לפרסם מהפרופיל שלו tweet של משתמש אחר ולהביא קרדיט למפרסם המקורי

דוגמא:

פה משתמש בשם ND Stevenson **עשה** retweet **לציוץ שהמשתמש** Pamela Ribon **פרסמה**

בנוסף קיימת גם התכונה של Quote Retweet שבה משמש יכול לעשות retweet לציוץ ולהוסיף ציטוט משלהם לתוכו

דוגמא:

פה משתמש בשם ND Stevenson עשה Quote Retweet לציוץ שהמשתמש Pamela Ribon פרסמה והוא הוסיף את התגובה שלו:

"Look at the babies"

המטרה שלנו

אנחנו רוצים לדעת אם דרך נתונים אחרים על tweet אפשר לחזות כמה retweets יהיו לו

הכלים שלנו

על מנת להגיע למטרה שלנו נשמש ב:
Selenium
הAPI של twitter
ו tweepy ל-Python

בנוסף נשתמש במודולים הבאים:

selenium
tweepy
Numpy
pandas
json
BeautifulSoup
IPython.display import clear_output #clears output
datetime
Time
seaborn
scipy
sklearn
matplotlib

איך מתחילים?

נתחיל בכך שנבקש מAPI של טוויטר 10,000 ציוצים ונוציא מהם את כל מה שאנחנו יכולים.

```
tweets_list = tweepy.Cursor(api.search_tweets, tweet_mode="extended",include_entities=True,count = 100).items(10000)
```

<u>בעיה:</u> הרבה מאוד מהציוצים שקיבלנו הם בשפות שונות, הרבה מהם הם תגובות לציוצים אחרים ואפילו יותר הם בעצם retweets שטוויטר מחשיב בציוצים בעצמם. מה עושים?

<u>פתרון:</u> נצמצם את הבעיה!

אנחנו בסוף החלטנו לצמצם את האיזורים שמהם אנחנו לוקחים את הציוצים ל: ארצות הברית, הממלכה המאוחדת, קנדה, ני-זילנד ואוסטרליה. זה יבטיח שכמעט כל הציוצים יהיו באנגלית. ניקח 2000 מכל מדינה.

בנוסף אנחנו נוסיף לquery הגדרות שאומרות לAPI לא לקחת ציוצים שהם תגובות ואו ציוצים שהם כבר retweets.

```
tweets_list += tweepy.Cursor(api.search_tweets, q="place:"+ usa + ' -filter:replies -filter:retweets',
                             tweet_mode="extended",
                             include_entities=True,count = 100).items(max_count_4_each_country)
```

הוצאת נתונים מהציוצים שרכשנו

כעט אנחנו נוציא כל מה שאנחנו צריכים מציוצים שרכשנו.

מתוך הציוצים אפשר להוציא:

טקסט, תאריך הוצאת הציוץ, כמה תגובות יש לו, כמה לייקים יש לציוץ וכו'.

אפשר גם דרך הציוץ לראות מי צייץ אותו ובכך לרכוש גם מידע על מי שצייץ אותו.

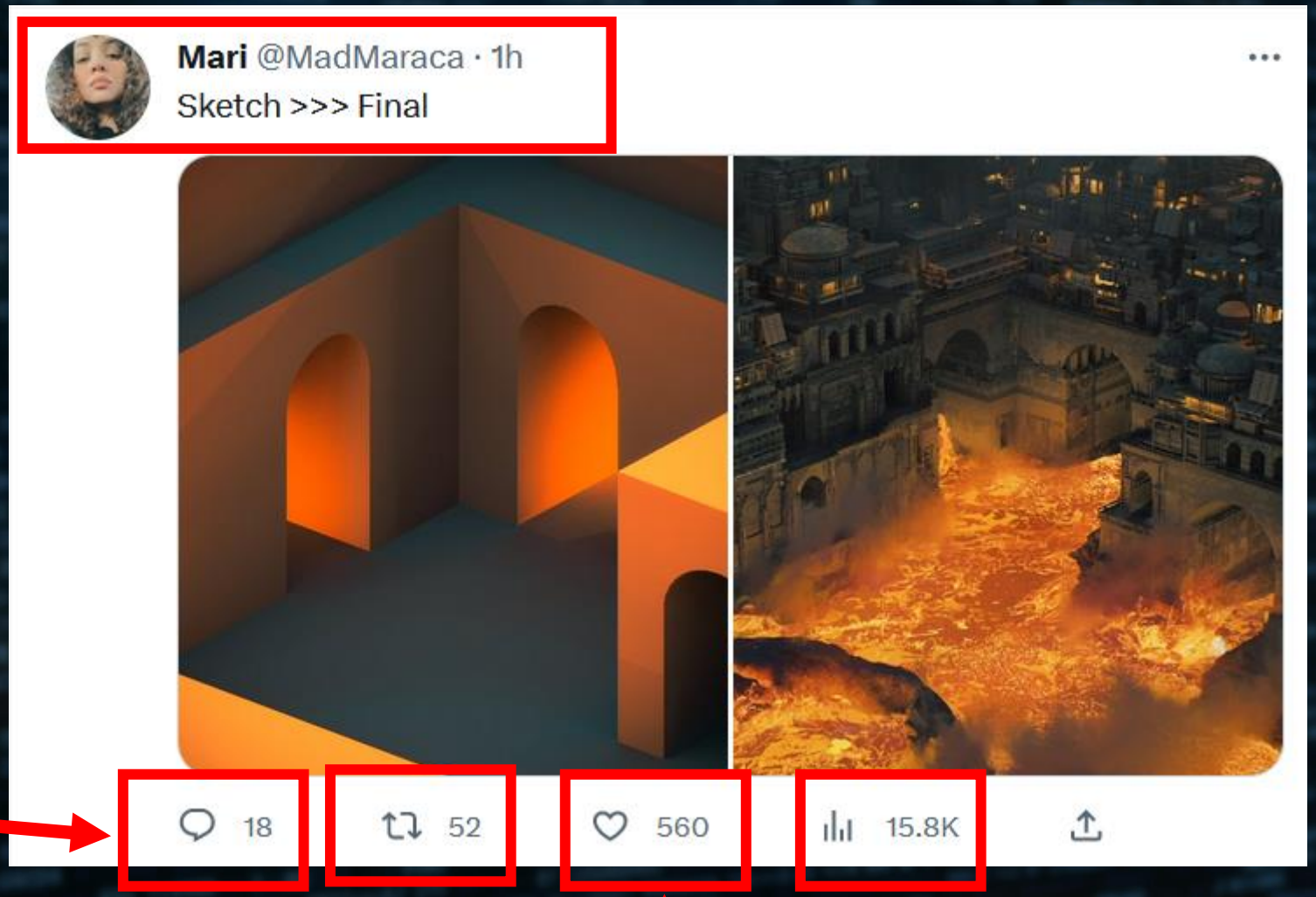
<u>בעיה</u>: הAPI של טוויטר רוצה להסתיר מאיתנו כמה דברים ואו ממש קשה להוציא אותם ממנו, מה עושים?

<u>פתרון</u>: נוציא את מה שדרוש לנו בכוח בעזרת crawling!

קודם כל צריך להבין איך ציוץ ופרופיל בנויים

איך tweet בנוי

User



Comment Count

Comment Count

retweet Count

likes Count

Views Count

איך פרופיל בנוי

name

bio

creation date

how many other users is this user following

birthday

how many other users are Following this user

בעזרת סלניום הוצאנו:

גיל המשתמש – בהרבה פרופילים המשתמשים בחרו להראות מתי הם נולדו, זה לא מופיע בAPI

מיקום – יש פרופילים שמראים מאיפה הם הגיעו

תאריך יצירת חשבון – מתי החשבון נוצר.

<u>בעיה</u>: טוויטר מרשה לאנשים לכתוב מאיפה הם מגיעים, לכן הרבה אנשים בעצם רושמים מקומות פיקטיביים כבדיחה איפה שאמור להיות רשום המיקום שלהם, בנוסף בטוויטר אין באמת מקום שבו המשתמש יכול להזין את המין שלו.

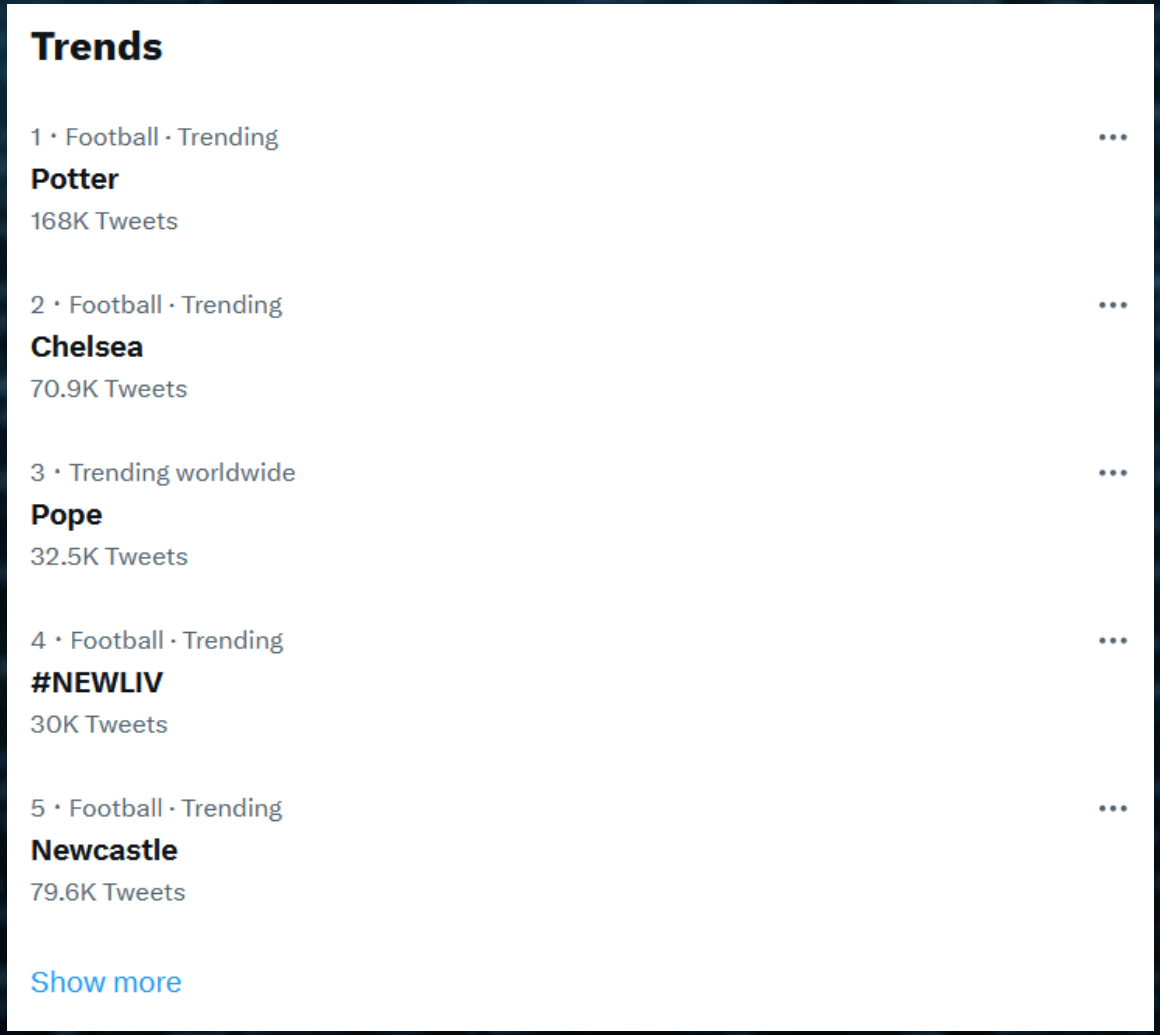
<u>פתרון</u>: יצרנו אלגוריתם שבודק אם השם של המיקום הוא אכן שם של מקום אמיתי בעזרת מודול geonamescache אשר מביא לנו רשימה של כל השמות של המדינות בעולם.

המין של המשתמש הייתה בעיה מעניינת. בתרבות של טוויטר מאוד מקובל שמשתמשים רושמים את הלשון פנייה שלהם בbio, בשם שלהם ואו אפילו המיקום שלהם בצורה של she/her, he/him או they/them

<u>לדוגמא:</u>

בנוסף:

בטוויטר אפשר לראות מה הטרנדים שכולם מדברים עליהם באותו רגע, לכן אנחנו מעוניינים לדעת אם הציוץ משתמש במילות טרנדיות



בטוויטר גם מקובל להדגיש את הנושא שמדברים עליו בעזרת סולמית, זה נקרא גם HashTag.

כאשר מישהוא מחפש בטוויטר את הנושא שמופיע בHashTag יש יותר סיכויים שטוויטר יראה את הציוץ שמשתמש בו.

לכן נרצה לדעת גם אם הציוץ משתמש בHashTag

בעיה: הAPI מביא לנו לבקש טרנדים של מקום מסויים רק 10 פעמים כל רבע שעה,מה שגורם לאי נוחות והארכת הזמן של תהליך הרכשת הנתונים

פתרון: מצאנו אתר שמראה לנו את כל הטרנדים העכשויים בטוויטר בצורה נוחה וברורה יותר ממה שהAPI סיפק.

כל מה שהיינו צריכים לעשות הייתה ליצור פונקציה getAllTrends() שהשתמשה בסלניום כדי לעשות crawling לאתר וליצור בעזרתו מילון של כל הטרנדים לפי עיר ומדינה.

```python
def getAllTrends():#returns a dictionary of all the trends
    trends = {}
    us_link = 'https://trends24.in/united-states/'
    nz_link = 'https://trends24.in/new-zealand/'
    uk_link = 'https://trends24.in/united-kingdom/'
    au_link = 'https://trends24.in/australia/'
    ca_link = 'https://trends24.in/canada/'

    main_links = [us_link,nz_link,uk_link,uk_link,au_link,ca_link]

    driver = webdriver.Chrome()
    wait = WebDriverWait(driver, 1)

    for l in main_links:
        driver.get(l)
        city_elements = wait.until(EC.presence_of_element_located((By.CSS_SELECTOR, ".suggested-locations__list")))
        country_name = driver.find_element(By.CSS_SELECTOR,'#app-bar-toggle > span:nth-child(1)').text #get the area name
        trends[country_name] = []
        tr = driver.find_element(By.CSS_SELECTOR, "div.trend-card:nth-child(1) > ol:nth-child(2)")
        tr = tr.find_elements(By.TAG_NAME,'a')
        for y in tr:
            trends[country_name].append(y.text)

        tmp_links = [x.get_attribute('href') for x in city_elements.find_elements(By.TAG_NAME,'a')]

        for x in tmp_links:
            driver.get(x)
            tr = wait.until(EC.presence_of_element_located((By.CSS_SELECTOR, "div.trend-card:nth-child(1) > ol:nth-child(2)")))
            name = driver.find_element(By.CSS_SELECTOR,'#app-bar-toggle > span:nth-child(1)').text #get the area name
            name = name[:-(2+len(country_name))] #remove the name at the end, for example ', United States' at the end
            trends[name] = []
            tr = tr.find_elements(By.TAG_NAME,'a')
            for y in tr:
                trends[name].append(y.text)

    driver.quit()
    return trends
```

בסוף קיבלנו DataFrame כזה:

	name	age	city	country	gender	acount age	total tweets	followers	following	respectability	...	word count	is quote	hashtags	hashtag count	trending	using trends?	metaData	join_date	date	tweet id
0	giathxo	NaN	Bethpage	United States	NaN	11.3	3867	216	445	2.060185	...	5	True	NaN	0	[Saka, #AVLARS, Zinchenko, Xhaka, #SaturdayMor...	False	--------	2013-12-16	2023-02-18	1626942985463013376
1	chenguanxi7979	NaN	Los Angeles	United States	NaN	2.5	3	1	3	3.000000	...	2	False	NaN	0	[#gokingsgo, #AVLARS, Saka, Zinchenko, Mings, ...	False	--------	2022-09-18	2023-02-18	1626942982774525954
2	HolaArizona	NaN	Phoenix	United States	NaN	11.7	3929	68	163	2.397059	...	57	False	NaN	0	[#SmackDown, Saka, #AVLARS, Zinchenko, Xhaka, ...	False	--------	2013-12-20	2023-02-18	1626942982397267969
3	420PandaNation	NaN	Michigan	United States	NaN	3.1	5502	2712	2478	0.913717	...	19	False	NaN	0	[Saka, #AVLARS, Zinchenko, Xhaka, #SaturdayMor...	False	--------	2022-09-25	2023-02-18	1626942978760613889
4	JeffreyLuscombe	NaN	Fort Lauderdale	United States	NaN	15.5	342636	13792	11222	0.813660	...	16	False	NaN	0	[Saka, #AVLARS, Zinchenko, Xhaka, #SaturdayMor...	False	--------	2008-11-06	2023-02-18	1626942973282770945
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
9995	LuisaLongone	NaN	Auckland	New Zealand	NaN	10.9	1518	487	824	1.691992	...	8	True	[MIEExpertNZ, MIEE, MIEEFellow]	3	[#NZvENG, MySpace, napier, SMS 2FA, Chantelle,...	False	--------	2014-06-23	2023-02-16	1626295677088329728
9996	moanaduffy25	NaN	Auckland	New Zealand	NaN	3.4	127	2	12	6.000000	...	23	False	NaN	0	[#NZvENG, MySpace, napier, SMS 2FA, Chantelle,...	False	--------	2022-07-29	2023-02-16	1626295570699780096
9997	everylotchc	NaN	Christchurch City	New Zealand	NaN	4.4	33456	564	59	0.104610	...	7	False	NaN	0	[#NZvENG, MySpace, napier, SMS 2FA, Chantelle,...	False	--------	2020-01-17	2023-02-16	1626295394639683584
9998	everylotakl	NaN	Auckland	New Zealand	NaN	4.3	38084	828	55	0.066425	...	5	False	NaN	0	[#NZvENG, MySpace, napier, SMS 2FA, Chantelle,...	False	--------	2020-01-16	2023-02-16	1626295393003933697
9999	everylotwlg	NaN	Wellington City	New Zealand	NaN	4.4	33587	1105	111	0.100452	...	13	False	NaN	0	[#NZvENG, MySpace, napier, SMS 2FA, Chantelle,...	False	--------	2020-01-17	2023-02-16	1626295391296827395

10000 rows × 24 columns

3 DAYS LATER...

לאחר שהבאנו לציוצים להתבשל במשך 3 ימים, חזרנו אליהם ויצרנו DataFrame חדש עם כל הנתונים המעודכנים שלהם.

	name	age	city	country	gender	acount age	total tweets	followers	following	respectability	...	retweets	quote retweets	comments	word count	is quote	hashtags	hashtag count	trending	using trends?	link to tweet
0	streetsforall	NaN	Desert Hot Springs	United States	NaN	13.5	4714	8223	137	0.016661	...	11	0	3	34	True	NaN	0	['#TabooToken', 'Chargers', 'Herbert', 'Staley...	False	https://www.twitter.com/streetsforall/status/1...
1	Gatitaconestres	NaN	New Hampshire	United States	NaN	5.0	12600	333	189	0.567568	...	0	0	0	4	True	NaN	0	['#TabooToken', 'Chargers', 'Herbert', 'Staley...	False	https://www.twitter.com/Gatitaconestres/status...
2	Cheli_Smith	NaN	Upstate New York	United States	NaN	12.8	13567	542	2121	3.913284	...	0	0	0	17	False	['shortfilmmaking', 'onset', 'cinema', 'tvfilm...	7	['#TabooToken', 'Chargers', 'Herbert', 'Staley...	False	https://www.twitter.com/Cheli_Smith/status/161...
3	IgawaPastor	NaN	Mission Viejo	United States	NaN	10.2	5113	150	398	2.653333	...	0	0	0	9	False	NaN	0	['#TabooToken', 'Chargers', 'Herbert', 'Staley...	False	https://www.twitter.com/IgawaPastor/status/161...
4	WalshJesuit	NaN	Walsh Jesuit High School	United States	NaN	16.1	1960	3934	484	0.123030	...	7	2	1	41	False	['PartnersinEducation']	1	['#TabooToken', 'Chargers', 'Herbert', 'Staley...	False	https://www.twitter.com/WalshJesuit/status/161...
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
9301	arshadzackeriya	NaN	Wellington City	New Zealand	NaN	14.0	1022	289	482	1.667820	...	0	0	0	26	False	['DevOps', 'DevOpswithZack']	2	['Paula', 'Chargers', 'Paul Henry', '#UFCVegas...	False	https://www.twitter.com/arshadzackeriya/status...
9302	Ncookie98	NaN	Auckland	New Zealand	NaN	11.3	133568	121	516	4.264463	...	0	0	0	2	False	NaN	0	['Paula', 'Chargers', 'Paul Henry', 'Perth', '...	False	https://www.twitter.com/Ncookie98/status/16138...
9303	erimedi	NaN	Christchurch City	New Zealand	NaN	8.0	10015	190	376	1.978947	...	0	0	0	11	False	NaN	0	['Paula', 'Chargers', 'Paul Henry', '#UFCVegas...	True	https://www.twitter.com/erimedi/status/1613849...
9304	auralorgasm	NaN	Kapiti Coast District	New Zealand	NaN	9.8	56050	490	412	0.840816	...	0	0	0	4	True	NaN	0	['Paula', 'Chargers', 'Paul Henry', '#UFCVegas...	False	https://www.twitter.com/auralorgasm/status/161...
9305	Wiki_Pita	NaN	Auckland	New Zealand	NaN	9.3	21218	1970	3482	1.767513	...	0	0	0	11	True	NaN	0	['Paula', 'Chargers', 'Paul Henry', 'Perth', '...	False	https://www.twitter.com/Wiki_Pita/status/16138...

כעט נתחיל להתבונן בנתונים, להסיר דברים ולהסיק מסקנות

ראשית אנחנו רואים שכמעט כל העמודה של "גיל" ו"מין" היא NaN, כלומר העמודות האלה כבר לא שמישות כי הן לא אומרות לנו כלום, נוריד אותן

```
[4]: df.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 48770 entries, 0 to 9352
Data columns (total 25 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Unnamed: 0      48770 non-null  int64
 1   name            48770 non-null  object
 2   age             624 non-null    float64
 3   city            48770 non-null  object
 4   country         48770 non-null  object
 5   gender          0 non-null      float64
 6   acount age      48770 non-null  float64
 7   total tweets    48770 non-null  int64
 8   followers       48770 non-null  int64
 9   following       48770 non-null  int64
 10  respectability  48770 non-null  float64
 11  verified        48770 non-null  bool
 12  text            48770 non-null  object
 13  views           48770 non-null  int64
 14  likes           48770 non-null  int64
 15  retweets        48770 non-null  int64
 16  quote retweets  48770 non-null  int64
 17  comments        48770 non-null  int64
 18  word count      48770 non-null  int64
 19  is quote        48770 non-null  bool
 20  hashtags        11594 non-null  object
 21  hashtag count   48770 non-null  int64
 22  trending        48770 non-null  object
 23  using trends?   48770 non-null  bool
 24  link to tweet   48770 non-null  object
dtypes: bool(3), float64(4), int64(11), object(7)
memory usage: 8.7+ MB

[5]: df=df.drop(["age", "gender",'Unnamed: 0'], axis=1)
```

כעט צריך לטפל בבעיה: בוטים

בוטים הם משתמשים במופעלים על ידי מחשב ולא בן אדם, לרוב כדי לפרסם מוצר ואו להפיץ חדשות.
שמנו לב שיש הרבה שמות שמופיעים יותר מפעם אחת בטבלאות שלנו. הסיבה שזה מעלה חשד לבוטים היא מכיוון שכל טבלה נוצרה על ידי ציוצים שצויצו בטווח של 2 עד 5 דקות לפני הפעלת הקוד שמצא אותם. זה לא נפוץ שבני אדם מצייצים יותר מפעם אחת בזמן הזה.

לנו יש 6 טבלאות, כל אחת בערך בגודל 10,000 שורות. אנחנו בדקנו עבור כל טבלה בנפרד אם יש שמות שמופיעים יותר מפעם אחת, אם כן אז נוריד את כל השורות עם השמות שלהם.

```
for i in range(0,6):
    df1[i].drop_duplicates(subset= 'name' ,keep = False, inplace=True)
```

לאחר מכן, נחבר את כל הטבלאות לטבלה אחת גדולה.

בסוף נשארנו עם טבלה אחת גדולה בגודל של 26 אלף שורות.

אמנם עדיין יש כפילויות של שמות, אבל אנחנו נניח שאלו בני אדם מכיוון שרוב הטבלאות נוצרו שעות ואו אפילו ימים אחד מהשני ומאוד יכול להיות שאלו אנשים שפשוט רשמו במקרה באותם זמנים שהפעלנו את הקוד.

כעט נתחיל את ניתוח המידע שלנו.

נוריד קודם כל דבר שהוא רק טקסט

```
df = dfn.drop(['name','text','quote retweets','following','respectability','trending', 'link to tweet','hashtags'], axis = 1)
#df.drop('name',axis = 1,inplace = True)
```

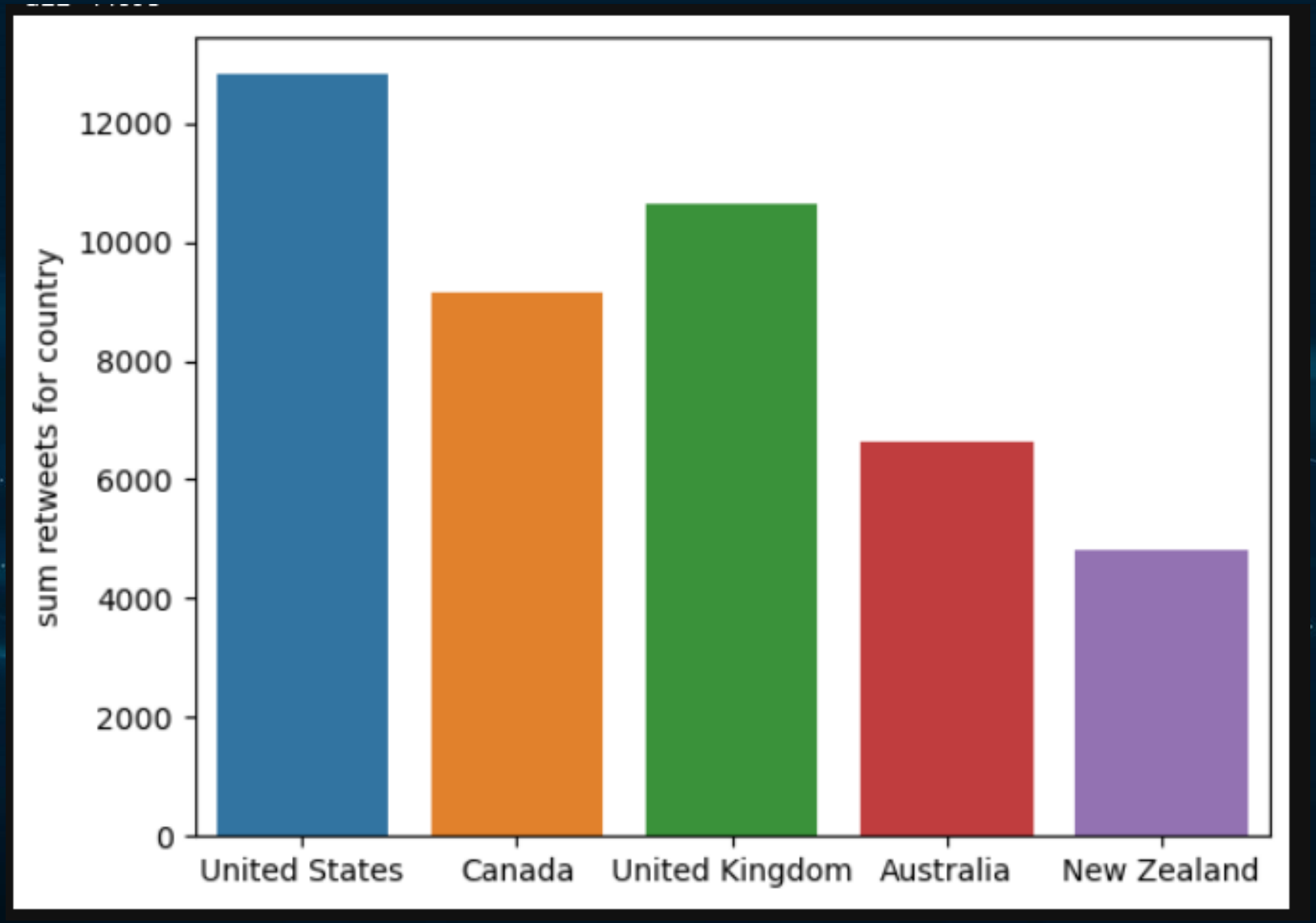
ונמיר את כל המשתנים הקטגוריאליים למספרים כדי שהלמידת מכונת תוכל לעבוד איתם אחר כך

```
df = df.copy()
for c in ['city','country','verified','is quote','using trends?']:
    df[c] = df[c].astype('category')
    df[c] = df[c].cat.codes
```

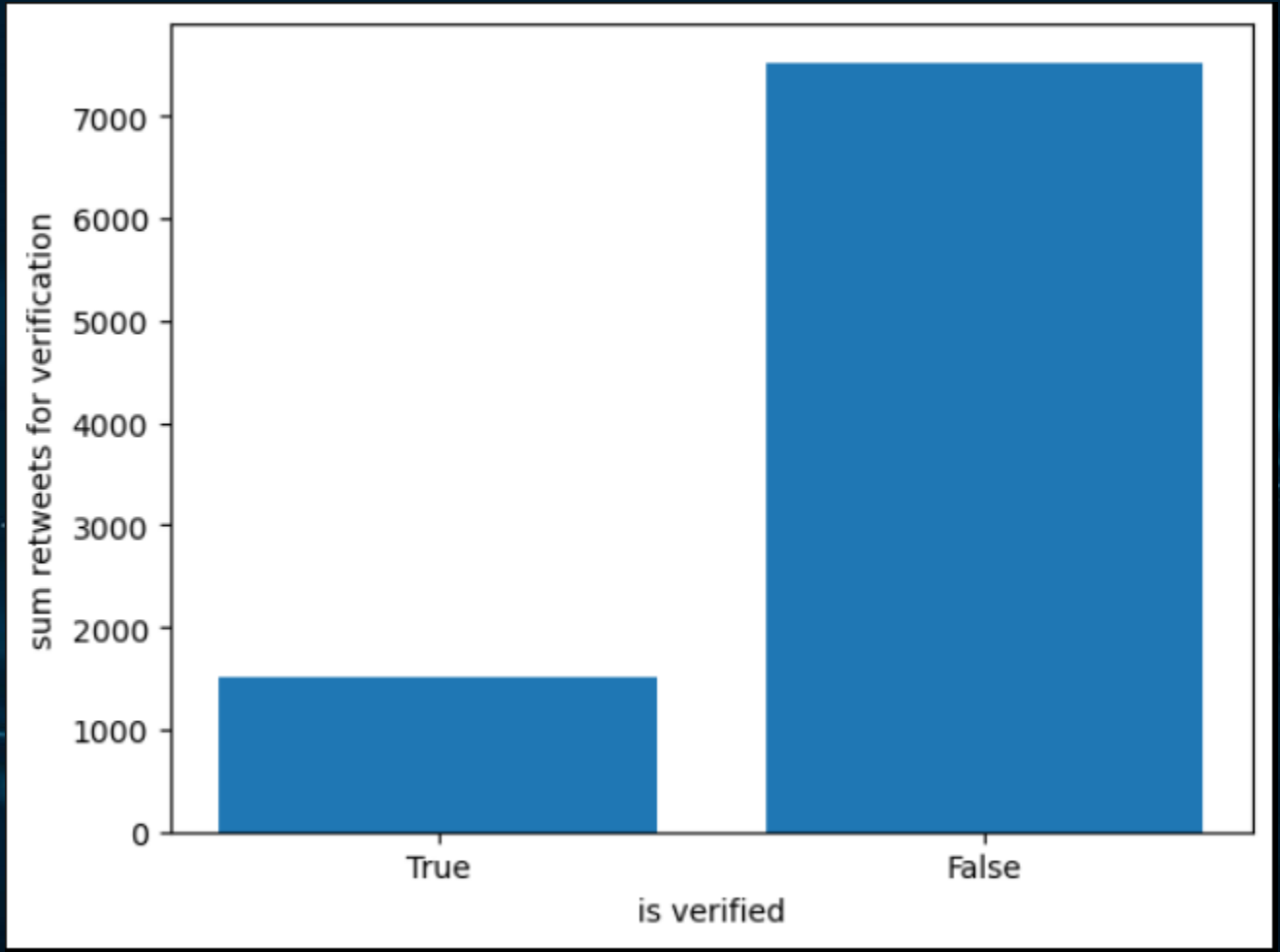
ונבדוק את הקורלציות בין משתנים

	city	country	acount age	total tweets	followers	verified	views	likes	comments	word count	is quote	hashtag count	using trends?	retweets
city	1.000000	-0.105194	0.012671	0.006928	0.007075	0.003349	0.001116	-0.000662	-0.003025	0.015921	0.005329	0.005889	-0.000252	0.001341
country	-0.105194	1.000000	0.008859	0.023344	0.001770	-0.007616	0.006275	0.006052	0.009536	-0.004431	0.048715	-0.056077	-0.018405	-0.000383
acount age	0.012671	0.008859	1.000000	0.016038	0.036189	0.110349	0.005416	-0.005934	-0.001248	0.014624	0.046967	-0.034932	0.030520	-0.007979
total tweets	0.006928	0.023344	0.016038	1.000000	0.001211	0.000918	-0.000727	-0.000844	-0.000400	-0.017371	-0.003578	-0.008226	-0.002864	-0.001138
followers	0.007075	0.001770	0.036189	0.001211	1.000000	0.147239	0.104543	0.056642	0.038925	0.002560	-0.007460	-0.002604	-0.004427	0.036118
verified	0.003349	-0.007616	0.110349	0.000918	0.147239	1.000000	0.111875	0.083314	0.065401	0.046278	0.027852	-0.000516	-0.009389	0.064890
views	0.001116	0.006275	0.005416	-0.000727	0.104543	0.111875	1.000000	0.863751	0.647494	0.021025	-0.000062	-0.009122	-0.007361	0.772564
likes	-0.000662	0.006052	-0.005934	-0.000844	0.056642	0.083314	0.863751	1.000000	0.750846	0.018823	-0.012747	-0.004286	-0.008747	0.822642
comments	-0.003025	0.009536	-0.001248	-0.000400	0.038925	0.065401	0.647494	0.750846	1.000000	0.026422	-0.014861	-0.004099	-0.005062	0.594196
word count	0.015921	-0.004431	0.014624	-0.017371	0.002560	0.046278	0.021025	0.018823	0.026422	1.000000	-0.109375	0.167023	0.037181	0.044075
is quote	0.005329	0.048715	0.046967	-0.003578	-0.007460	0.027852	-0.000062	-0.012747	-0.014861	-0.109375	1.000000	-0.131021	-0.059044	-0.013883
hashtag count	0.005889	-0.056077	-0.034932	-0.008226	-0.002604	-0.000516	-0.009122	-0.004286	-0.004099	0.167023	-0.131021	1.000000	0.063991	0.012341
using trends?	-0.000252	-0.018405	0.030520	-0.002864	-0.004427	-0.009389	-0.007361	-0.008747	-0.005062	0.037181	-0.059044	0.063991	1.000000	0.004504
retweets	0.001341	-0.000383	-0.007979	-0.001138	0.036118	0.064890	0.772564	0.822642	0.594196	0.044075	-0.013883	0.012341	0.004504	1.000000

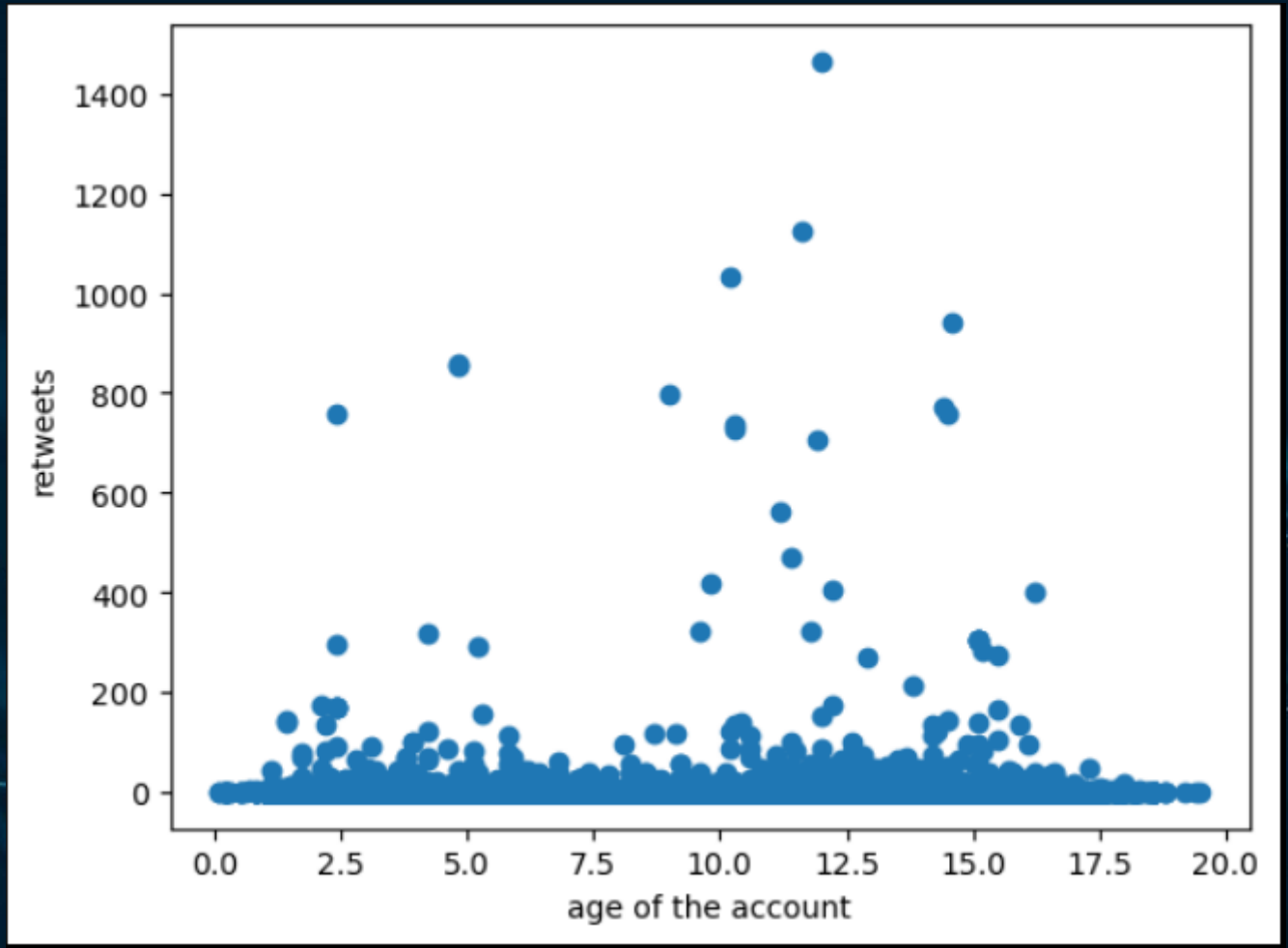
נסתכל על גרפים כדי לנסות להבין קצת יותר טוב מה קורה כאן



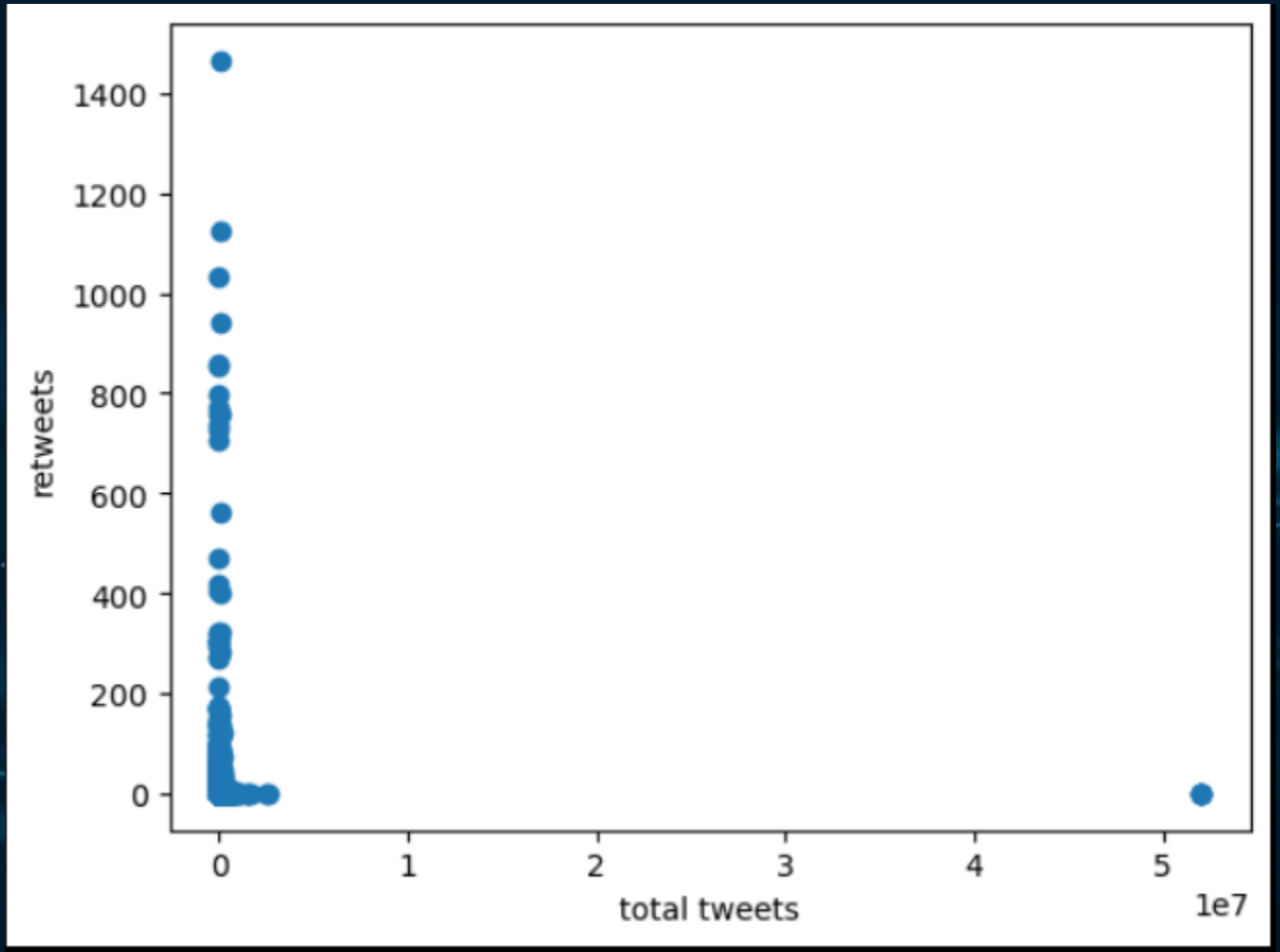
סכום כל הretweets מכל מדינה



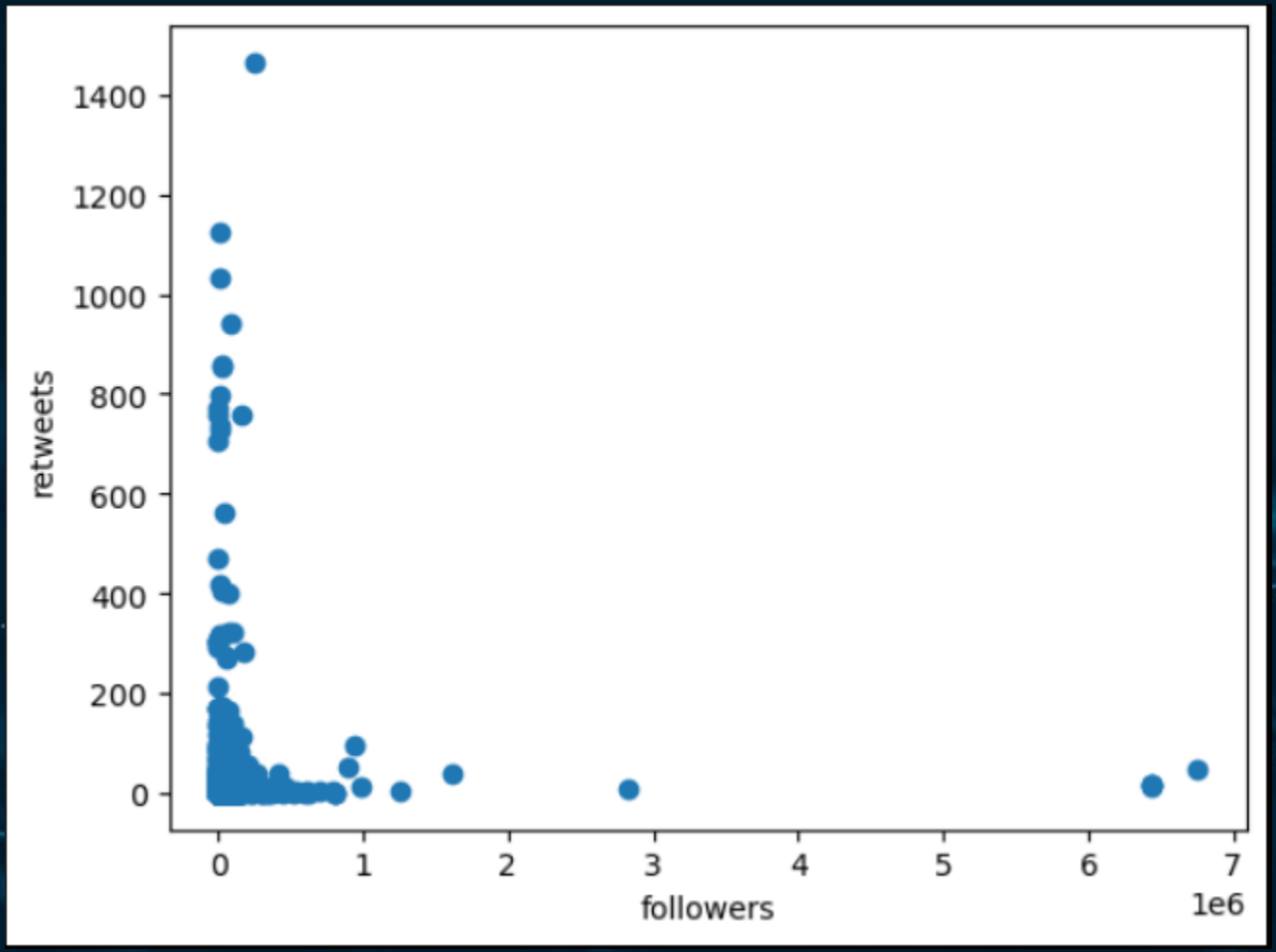
סכום כל הretweets של כל מי שverified ומי שלא



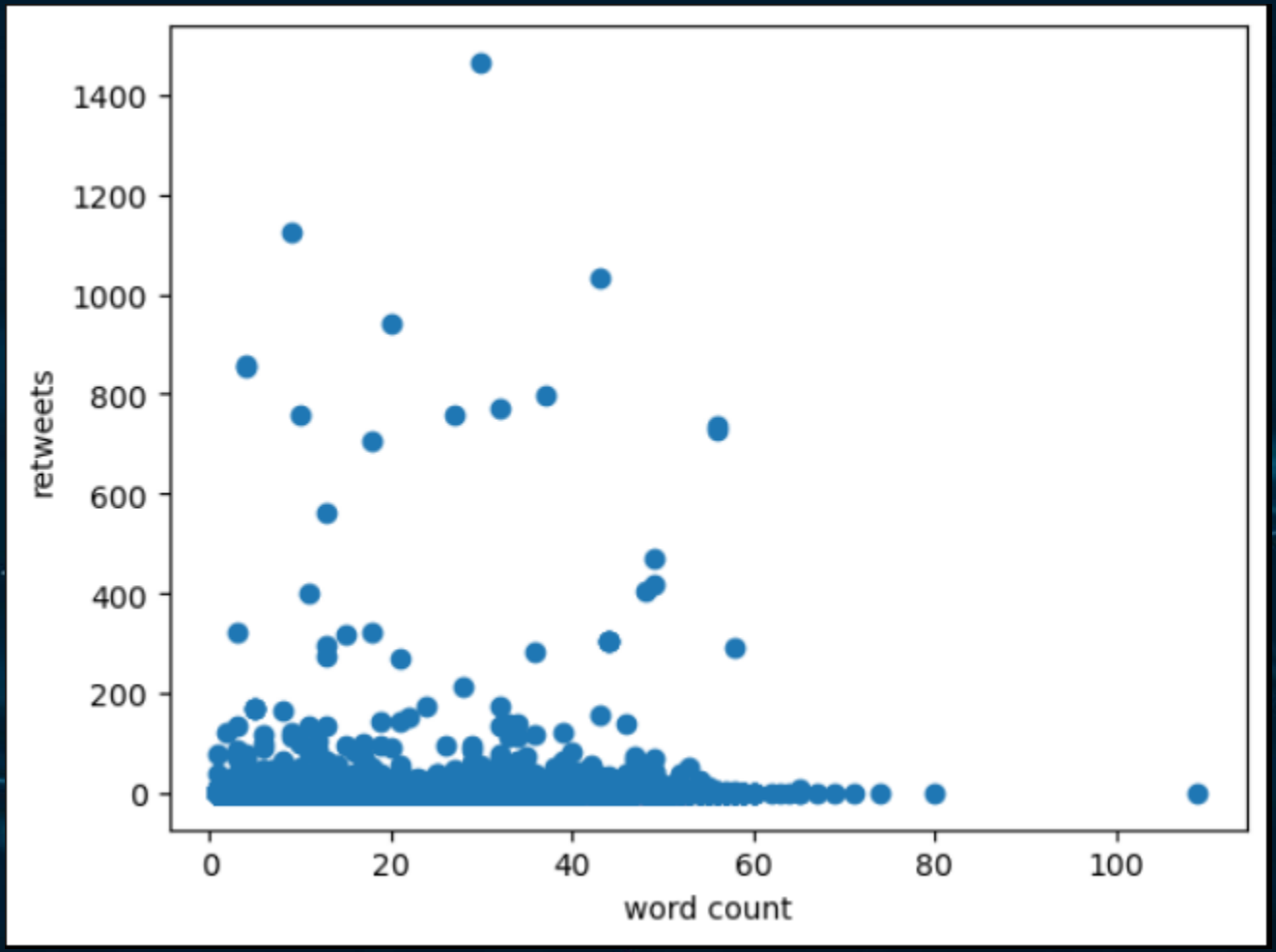
כמות הretweets לפי הגיל של האקאונט שפרסם את הציוץ



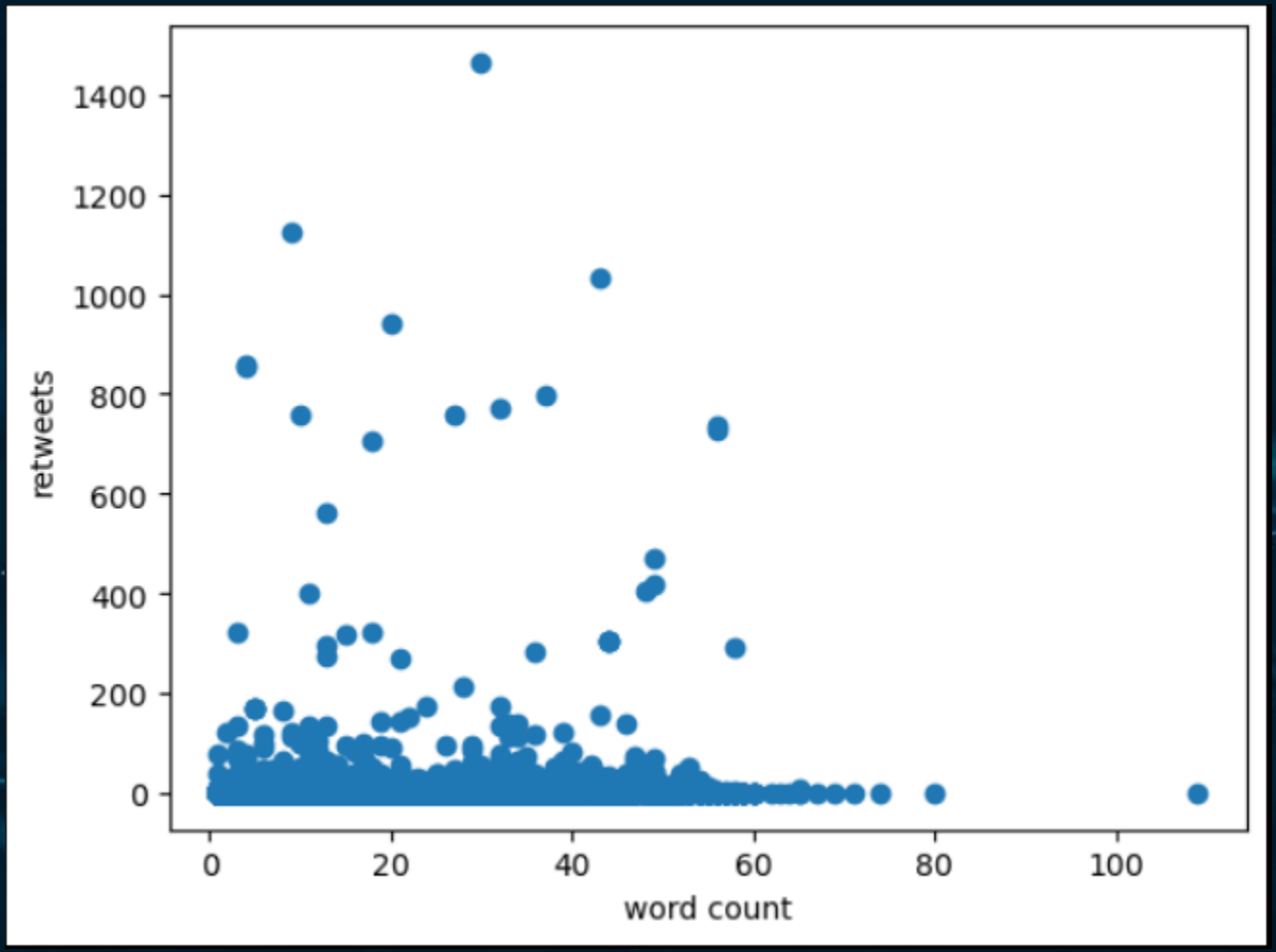
כמות הretweets לפי כמות הציוצים הכללית של האקאונט שפרסם את הציוץ



כמות הretweets לפי כמות העוקבים האקאונט שפרסם את הציוץ



כמות הretweets לפי כמות המילים שבציוץ



כמות הretweets לפי כמות המילים שבציוץ

החלטנו להביא ללמידת מכונה את: verified, comments,hashtag count,views, likes, using trends?

החלטנו להשתמש בmultiple linear regression

```
from sklearn import linear_model
from sklearn.model_selection import train_test_split

x = df.drop('retweets',axis = 1)
y = df['retweets']
```

```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.3 , random_state = 627)

lr = linear_model.LinearRegression() # create a linear regression object
lr.fit(x_train, y_train);

y_pred_train = lr.predict(x_train)
r2_score(y_train,y_pred_train)
```

```
0.8152665346132515
```

יצא שהמודל של המכונה היה כמעט קרוב למה שבאמת יוצא

```
plt.scatter(y_train,y_pred_train)
```

```
<matplotlib.collections.PathCollection at 0x12ad6bee490>
```