Security related articles in N12 (Mako) site

Data Science - Final Project

Insecure Time and Its Effect on the Economy

Research Query 1

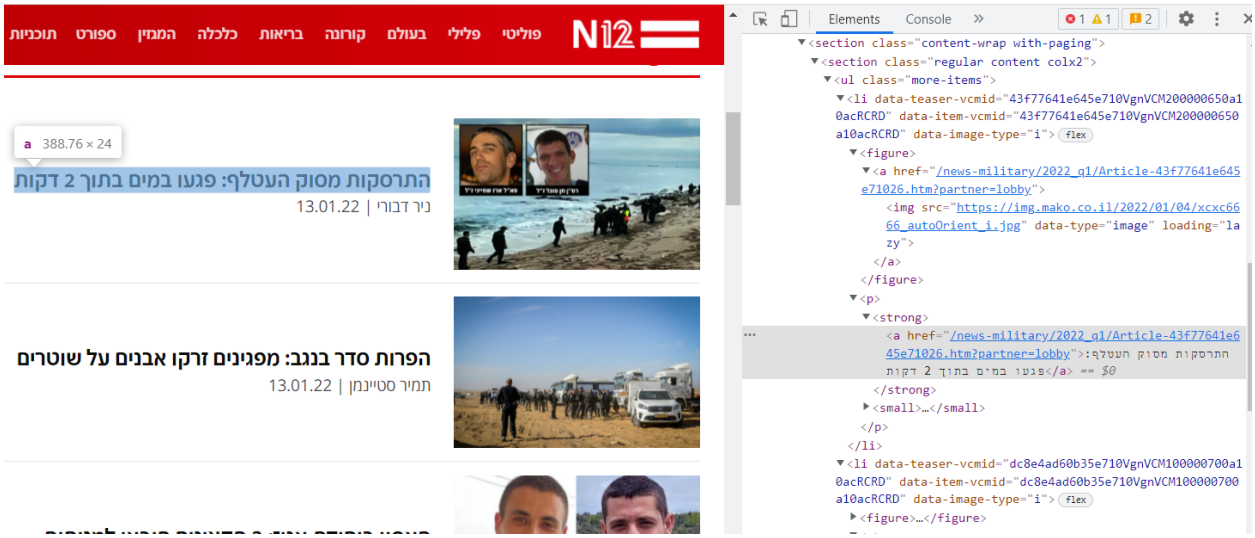
We are exploring the connection between times with significantly more security-related articles and a drop in main economy parameters.
Unfortunately in Israel, a high number of security-related articles indicate an insecure time (military operations etc), which may affect the economy.

For this research query:
The instances are days or weeks;
The features are:

- Number of security related articles
  And the economy indicators:
  - Shekel/Dollar exchange rate;
  - TLV125 stock exchange index.

Data Acquisition

Code: *N12_Crawling*
Crawled the site `https://www.mako.co.il/news-military:`



Scrolled over various pages; For each page:

Looped over ul class “more items” / li (list of articles):
For each article, collected:

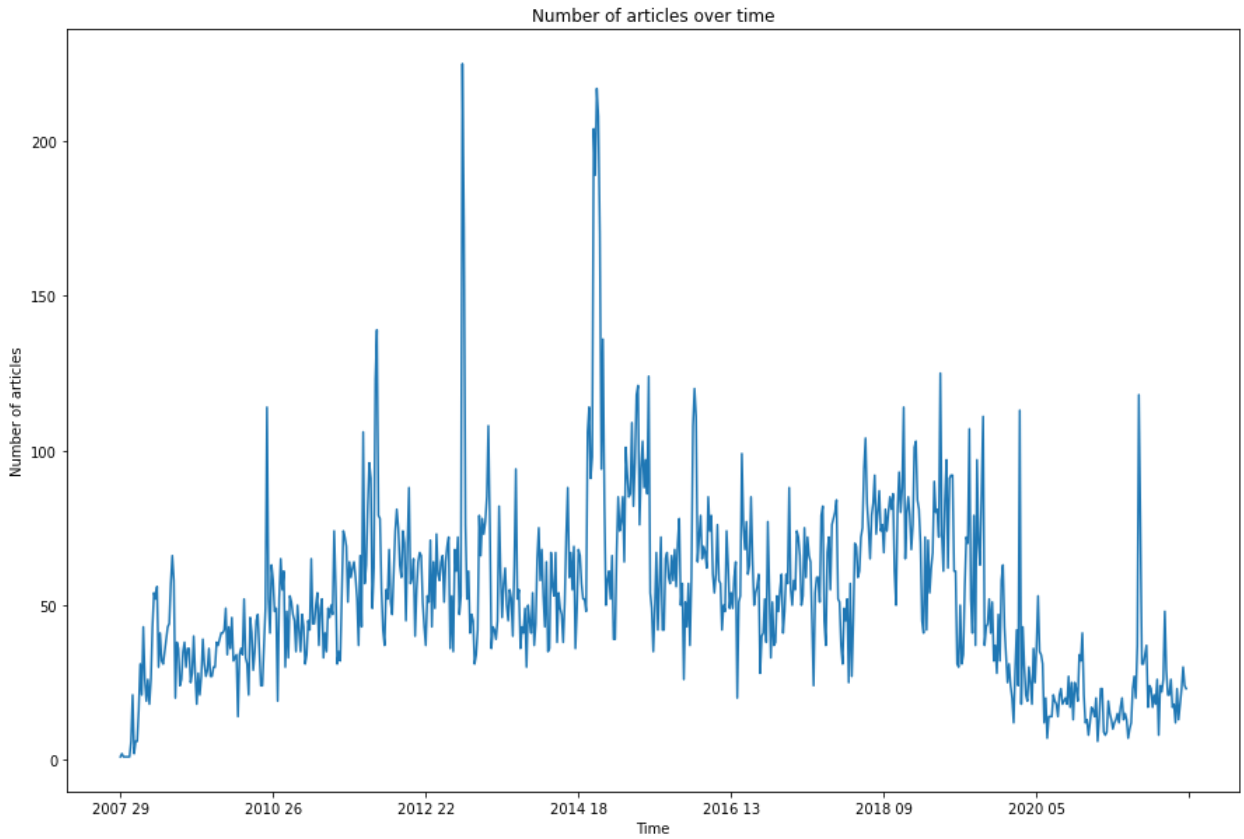
- Title text (p / <a href)
- Author (span[0])
- Date (span[1])

We crawled over 2000 pages, and collected articles from 13 years: 2008-2021 (36,804 articles)

Initial processing and data improvement:

Code: *N12-Data-Processing*
Improved the crawled data as follows:

- Date formatting - `pd.to_datetime(df['Time_Date'], dayfirst=True)`
- Week numbers - `df['Date'].dt.strftime("%G %V")`
- Number of articles per week - `df['Week_Year'].value_counts(sort=False)`
- Clean up - removed (`dropna`) lines with NaN year values

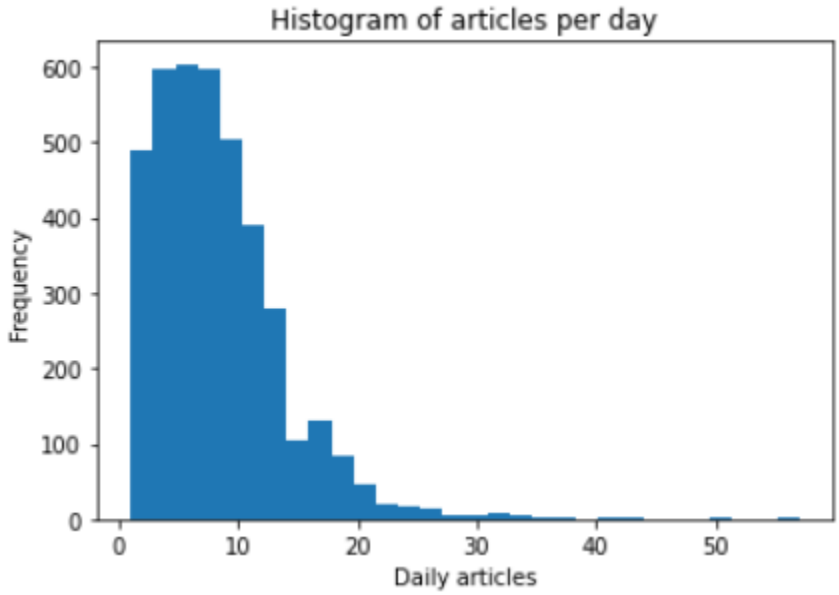


It seems that data before 2011 is missing, so we eliminated it:
`df.drop(df.index[df['Year'] <= 2010], inplace=True)`

Initially we accumulated articles per week, because we thought this would be more stable (without daily fluctuations). Later this proved to be less appropriate for reasons detailed below.

EDA: Definition of Insecure Time

Code: *EDA - USD rate*
We analyzed how many security articles are released per day.



Statistical parameters (using `Describe`):

```
count    3913.000000
mean        8.223102
std         5.555222
min         1.000000
25%         4.000000
50%         7.000000
75%        11.000000
max        57.000000
```

Based on the statistical parameters, we define “Insecure Time” as a day with > 20 articles (2*std above the average amount)
(There are 130 days with >20 articles)

ILS - USD rate and correlation to Insecure time

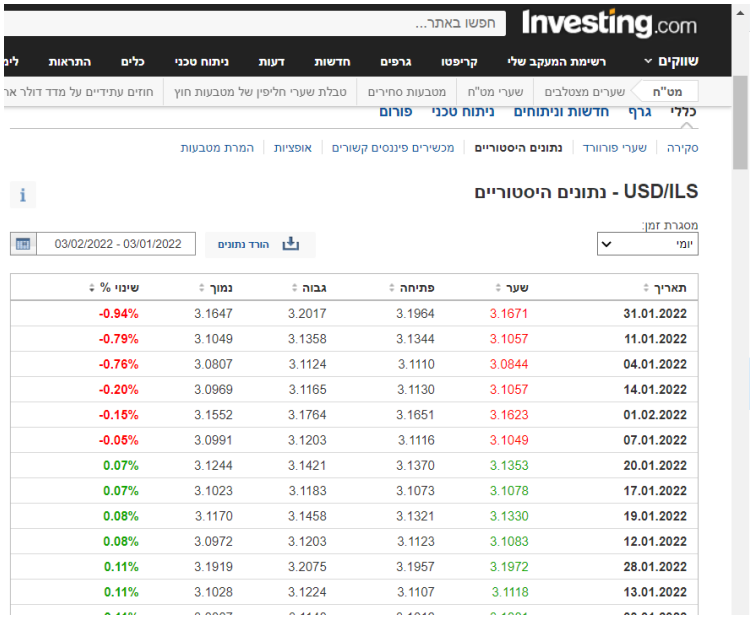
Hypothesis / Research question: During “Insecure Time” the ILS value drops.

ILS-USD rate Data Acquisition

Code: *ILS-USD Selenium , ILS-Data-Processing*

To get the ILS / USD currency rate we crawled:

`https://il.investing.com/currencies/usd-ils-historical-data`



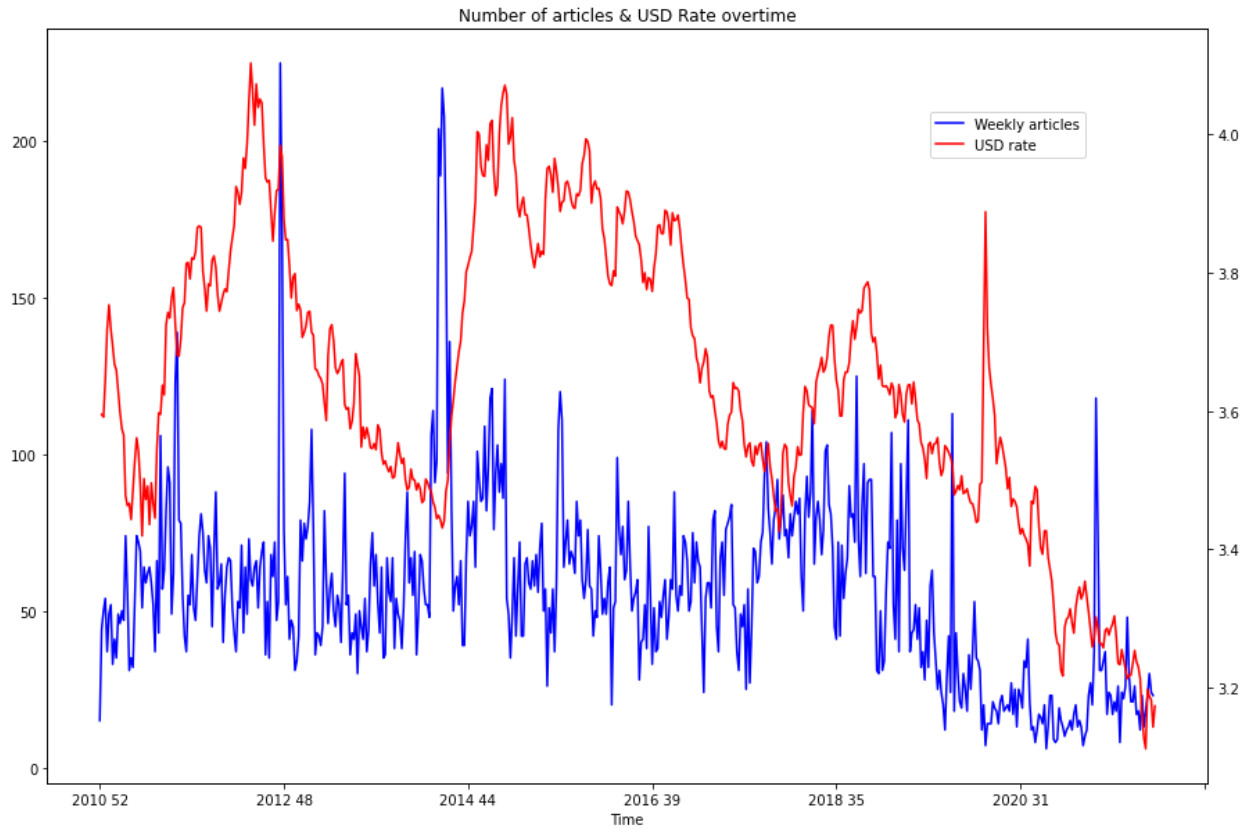
We used Selenium for the crawling:

- Set the dates in the `widgetFieldDateRange` field
- Read the ILS-USD rate into a dataframe and saved to file ILS-USD.csv .

ILS-USD rate EDA and analysis

Code: *EDA - USD rate*

Merged the daily article number and the ILS-USD dataframes:



It is possible but somewhat difficult to see the “jumps” in USD value during “Insecure Time”.
However this can clearly be seen when we compare averages:

Overall average daily change - (mean of “Change %”) : **-0.0027%** (USD value decreases) Average daily change during “Insecure Time” : **0.0292%** (USD value **increases** over **x10** of daily rate change!)

TLV125 index and correlation to Insecure time

Hypothesis / Research question: During “Insecure Time” the TLV125 index drops.

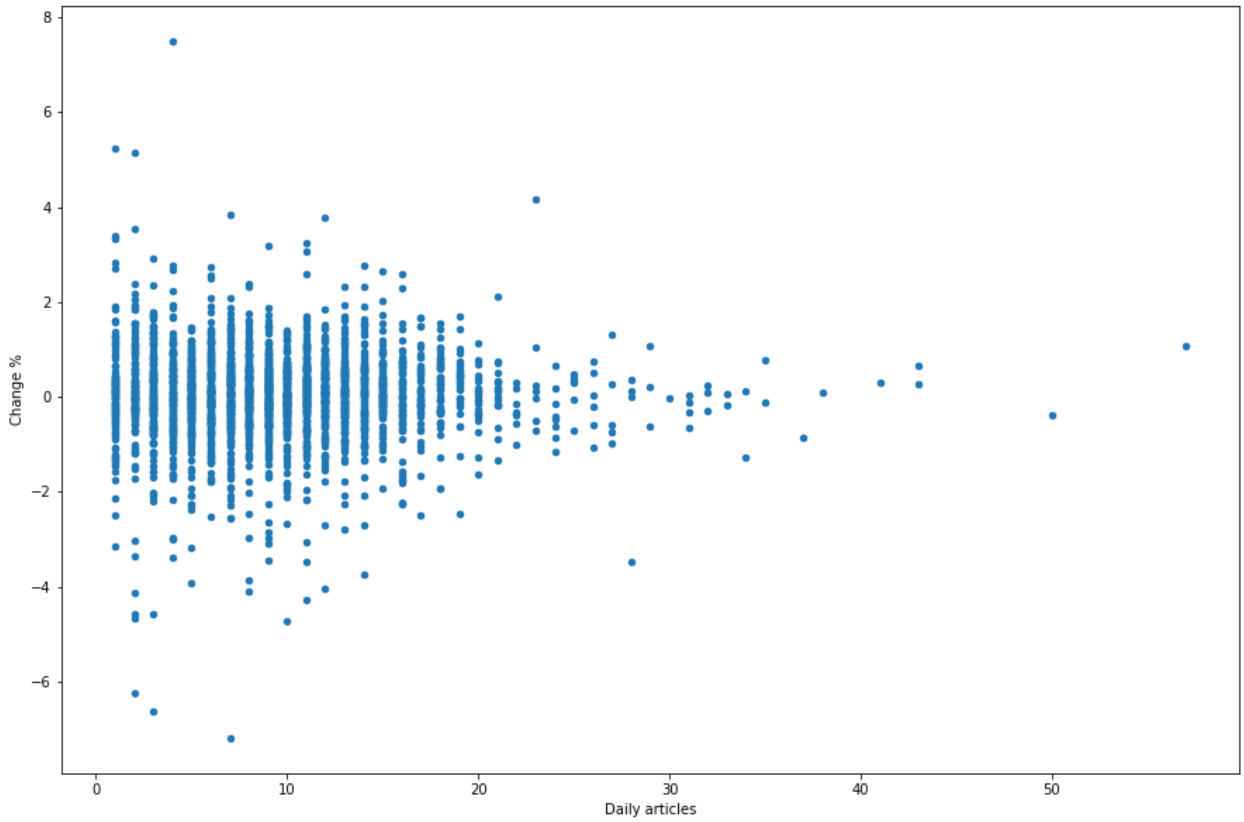
TLV125 index Data Acquisition

Code: *TA_125_Selenium*
To get the historical TLV125 values we crawled the site:
https://il.investing.com/indices/ta100-historical-data
In a similar way, using Selenium.

TLV125 index EDA and analysis

Code: *EDA - TLV125 rate*

We used a scatter graph to see the correlation between the number of security articles and TLV125 level.
In a general look it's difficult to see the effect:



But looking more closely on daily changes, the effect is clearly seen:

Overall average daily index change: **0.023%**
During "Insecure Time" : **-0.035%** - 50% higher, in the opposite direction!

The drop is significant but very transient:
1 Day following "insecure day" the mean daily change is 0.073% ! (3.2 times the average)
After 2 Days: 0.067% .

To see the daily change a day after Insecure day, we used the Shift function:

```
TA125_df['Change %'] = TA125_df['Change %'].shift(1)
```

Because of the transient effect of the TLV125 drop, the effect is hardly seen in weekly averages.

Also, we cannot use fillNA for days with no trade. If we do, we see a smaller daily index change - from **0.044%** to **0.033%**)

Text analysis

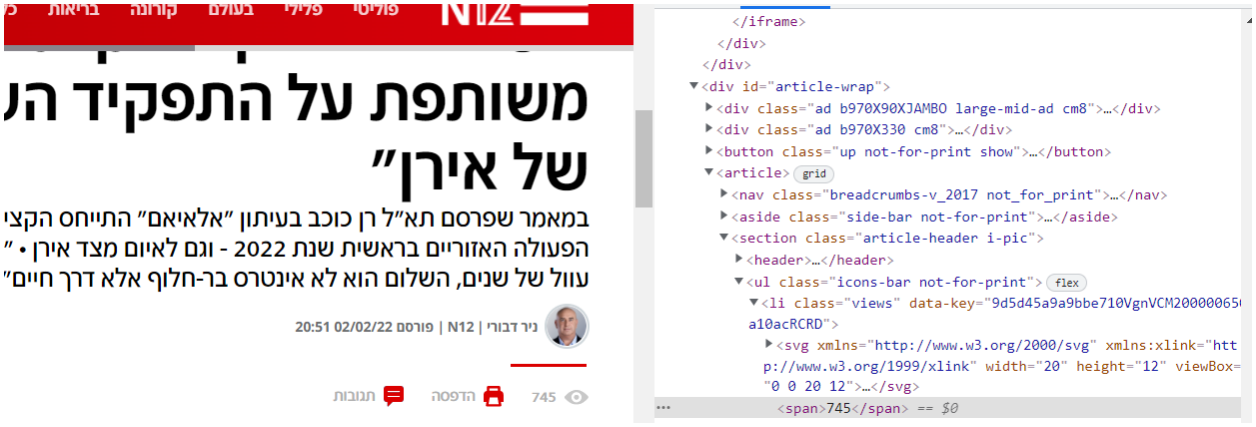
Research Query 2

For the N12 (Mako) site, we are trying to predict the success of specific titles, based on their text.

For this research query:
The instances are titles;
The features are word frequencies (extracted by `CountVectorizer`)
The supervised learning labels are Views (click through from the title to the article).

Data acquisition and cleaning

Code: *N12_Selenium_Article_Info*
To capture the views, we crawled the articles (using the URL extracted in the previous section). We used Selenium to extract the Views:

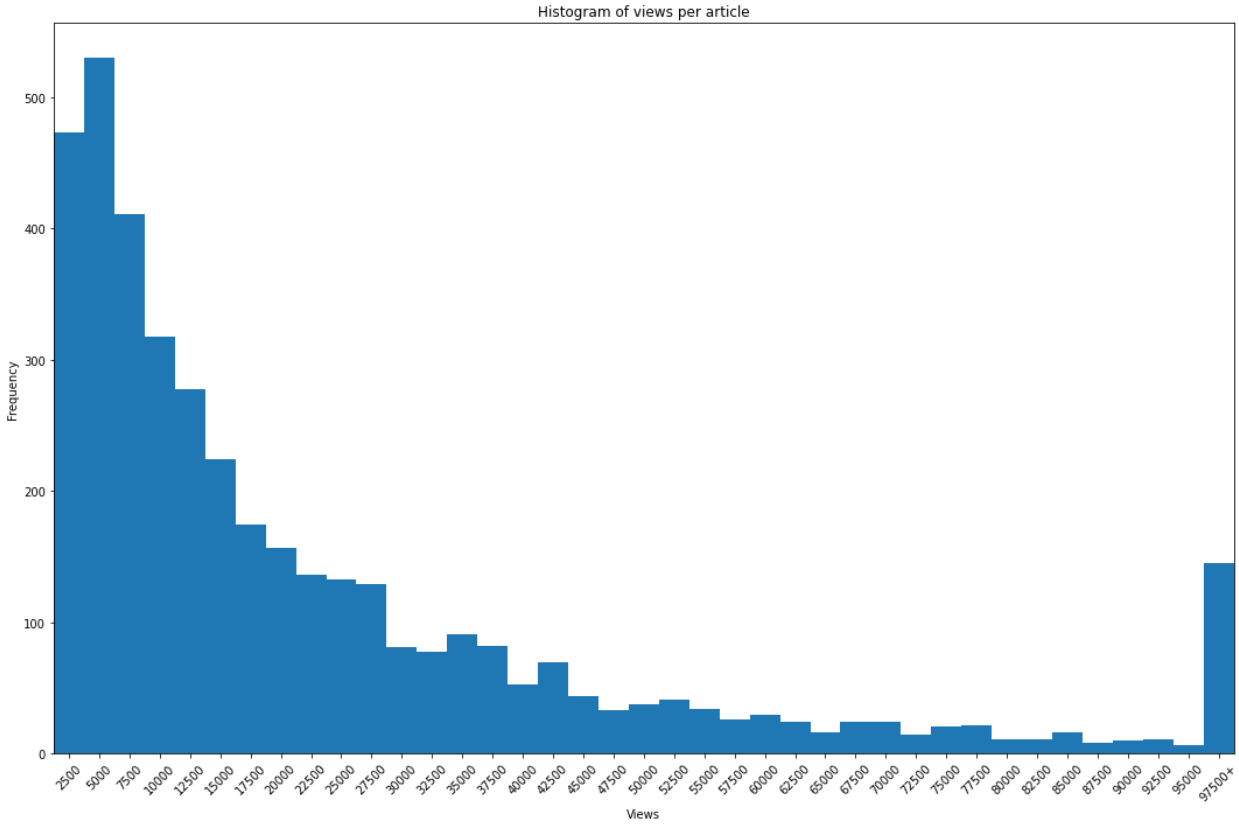



```
Views = WebDriverWait(driver, 5).until(
      EC.visibility_of_element_located((By.XPATH,
      '//*[@id="article-wrap"]/article/section[1]/ul/li[1]/span'))).g
      et_attribute("innerHTML")
```

For a small part of the articles, views were not captured. We used `dropna` to remove these entries.

Text EDA

First, let's find how many views are considered "successful".
Because the overall success of the N12 site changed over the years, we are considering titles from the last 3 years.



As can be seen, most of the articles get a few thousand views.
0.12 of the articles have more than 50,000 views - we define them as “successful titles”.
0.09 of the articles have less than 2,000 views - these are “unsuccessful titles”.

In order to see which words make a title successful, we compared word clouds of successful vs. unsuccessful articles:

Word cloud of successful titles

Word cloud of unsuccessful titles

(We also used word cloud to find stop words in Hebrew)

It can be seen that some words are common and popular in both populations (ישראל, עזה, נתניהו)
Some are typical to successful title (חמאס, חיזבאללה, סוריה, אירן)
And some typical for “uninteresting” titles (בעוטף, בדרום, בית, המבקר, שרפות, אזעקה)
(How unfortunate to see people are more interested in our enemies, and less in the effect on southern settlements - people are tired of it?)

Text machine learning

Next, we try to train a model for predicting a successful title. We used Naive Bayes (`MultinomialNB`) as a classifier.
We split the instances (titles) to 80% train and 20% test. Pipeline was used for vectorization, normalization and classification.

After training, we got **86%** accuracy when activating the model on the test set.