Weibo

One of the sources we need to scrap was the Weibo website. We retrieve the publications and the comments about Airlines companies and fly experience. In this report, we're going to explain in a first time which data we scrap and the format of these data. In a second time, we're going to develop our scraping method and how we automate the code with a robot and we're going to finish with some statistics.

I Format of scraped data :

The table below summary all data we scraped and give their type and a description of what is the column.

Variable name	Type	Description
Data_Source	string	Source
Date_publication	date	Date of publication
Description	string	Description of publication
Nb_sharing	int	Number of sharing
Likes	int	Number of likes
Nb_comments	int	Number of comments

II Scraping method:

We scrape our data in several steps :

- Creation of search equations to find the informations we need
- For each equation we get the source code of the page.
- For each publication of each page, we will retrieve and save the different variables in a dictionary if the publication is recent. Then we translate the publication the english
- The dictionnary values feed the json file.

To load website and get the data we use the packages requests and BeautifulSoup. For the translation we use TextBlob, it's a package using the google translate API.

We made also a Robot which collects last data. To use it, you need just to give the number of days you want to have (fixed to 7 days) and if you execute the code, you're going to retrieve only the one week old comments. With a command, you can automate the process on OSIRIM (read the READ.txt files to know how).

III Statistics:

All the statistics below are made on all publications without date limitation (we don't take only one week old publications).

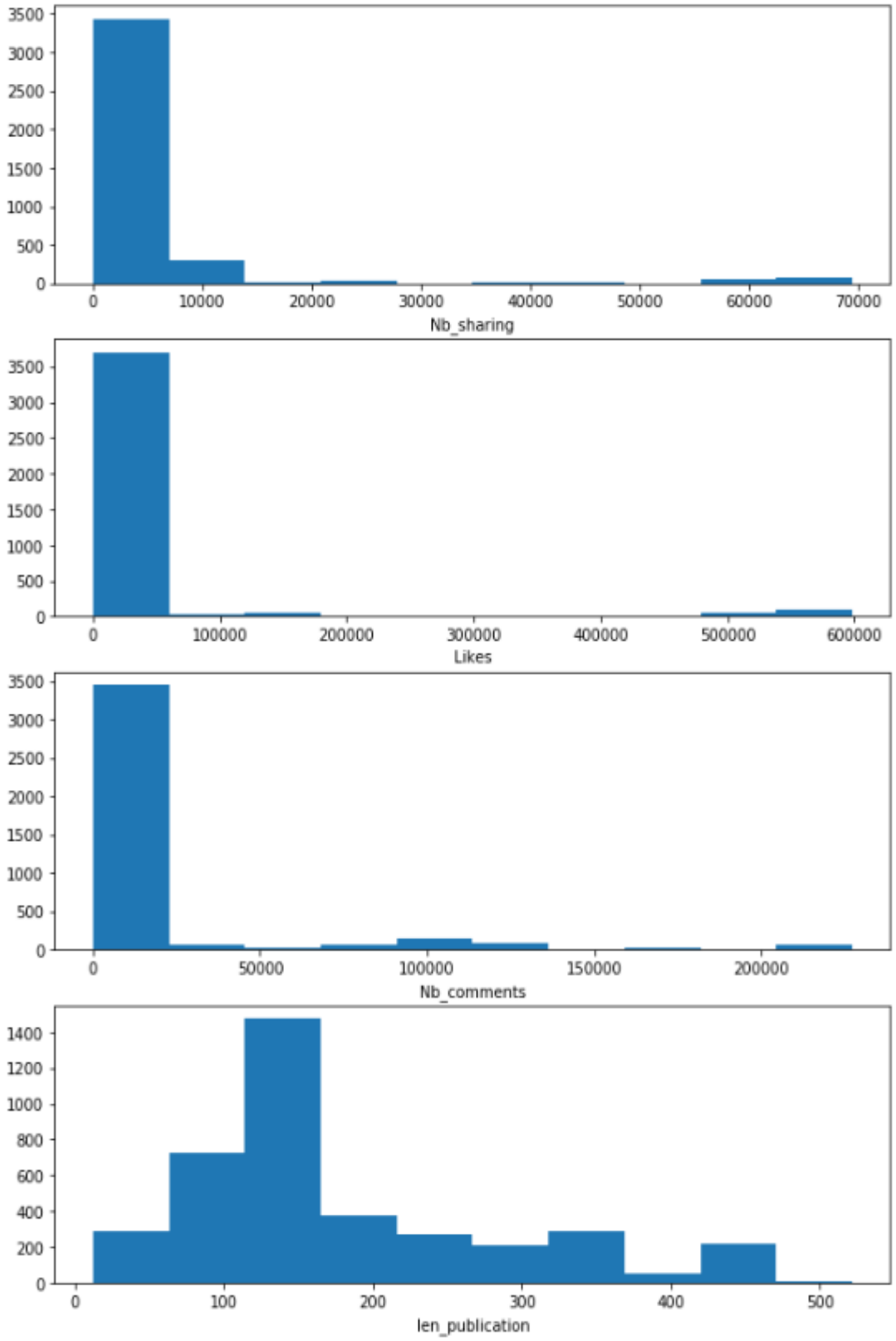
The final dataset counts 3918 rows X 72 columns and weigh 6,4 Mo.

To get the search equations we use 3 lists: One contain the air companies names, another the list of Boeing models and the last one contain the list of Airbus models.
We combine the Aircompanies list with the models lists. Then we combine the final list with the root of the website. We finally have **3431** url.

We got no null value on the variables that we get with the scraping

len_publication refers to the length of the publication's description

	Nb_sharing	Likes	Nb_comments	len_publication
count	3918.000000	3918.000000	3918.000000	3918.000000
mean	3666.550026	30384.798367	14250.313936	180.419602
std	11746.373572	101538.994302	41703.858468	112.072176
min	0.000000	0.000000	0.000000	12.000000
25%	68.000000	2169.000000	20.000000	112.000000
50%	349.000000	7038.000000	212.000000	136.000000
75%	1178.750000	9785.000000	934.000000	220.000000
max	69454.000000	598502.000000	227321.000000	522.000000



Histogram for each numeric column (length of text publication included):