Capstone 2 Milestone Report
Correlating web traffic to external events

Jonas Cuadrado

1. **Problem to solve:**

Knowing how different event affect the behavior of a company's client can allow a company to improve its service, wither by offering specific products or by adapting to the clients needs, as well as allowing the company to predict trends based on seasonal trends.

In this example, the goal is to correlate external events to Wikipedia searches spikes. The external events will be tracked from news articles written by real journalists from different media. I have chosen Wikipedia due to its publicly available data, its popularity, and its ability to attract plenty of users from diverse backgrounds.

2. **Client:**

In this specific project, Wikipedia could use that information to predict higher demand of traffic and open more servers to reduce delays. This can be generalized to any company.

In general, every company should be interested in how their users behave. Being able to track their interests and observe correlations between events is of upmost importance to determine how to improve and where to invest in the corporate model.

3. **Datasets:**

The Wikipedia project has readily available traffic data per article straight from the API. The description is available on https://dumps.wikimedia.org/other/pageviews/readme.html. There are two immediate options: directly accessing the tar.gz datafiles for each hour of the day, that contain the number of visits to every article per hour, and per day, or using aggregate counts straight to the API to obtain daily values.

I have opted for the daily approach to reduce the amount of data to handle, and to avoid hourly peaks that may be affected by the geographical location of the search. For instance, in the morning in Europe we may find searches on events related to what happened there rather than in the US, since in America it'd be the middle of the night.

There was a change in the data management, so we will only use data from after May 2015.

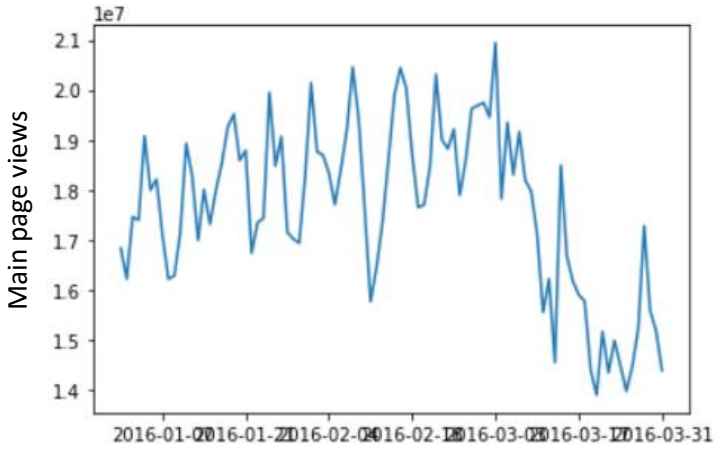
For the news, Kaggle has a public dataset containing articles from different sources for a period of time of over 1 year: https://www.kaggle.com/snapcrack/all-the-news

To better extract the data, some NLP will be required to obtain the topic, at least, for most of the articles. The date is immediately available, which facilitates the time correlation between events.

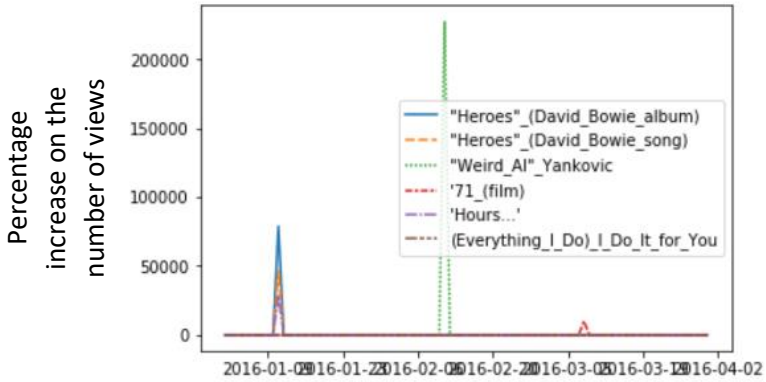
4. **Initial findings: Wikipedia dataset**

The Wikipedia dataset is well structured, essentially clean, and shows some of the expected behavior. Instead of accessing the daily views of all the articles and selecting the most relevant ones, it is possible to collect the 1000 articles most seen on each day. We can check if there are meaningful peaks in them, and if they are meaningful.

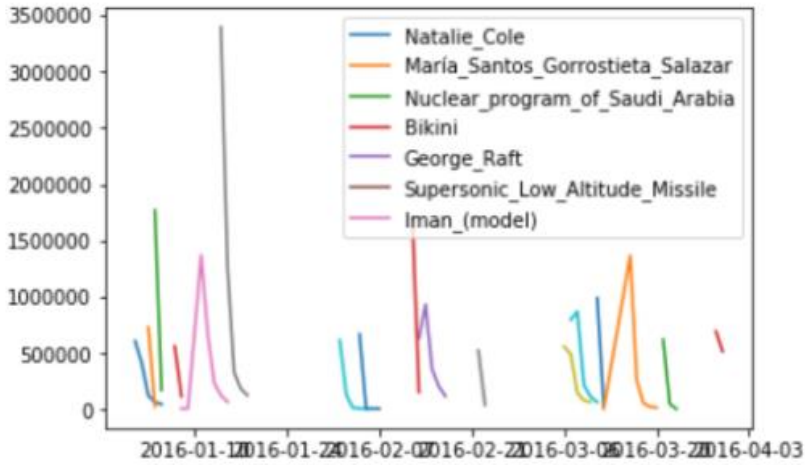
To perform an initial analysis, I have selected the months of January, February and March, 2016. The first noticeable property is that the main page is always the page with the most views. It has peaks, but it never grows or descends very importantly in percentage.



However, some specific events, like deaths or the release of a movie, can create a huge percentage increase in the traffic despite having a smaller total number of views. Indeed, the percentage change can go to over 10.000%



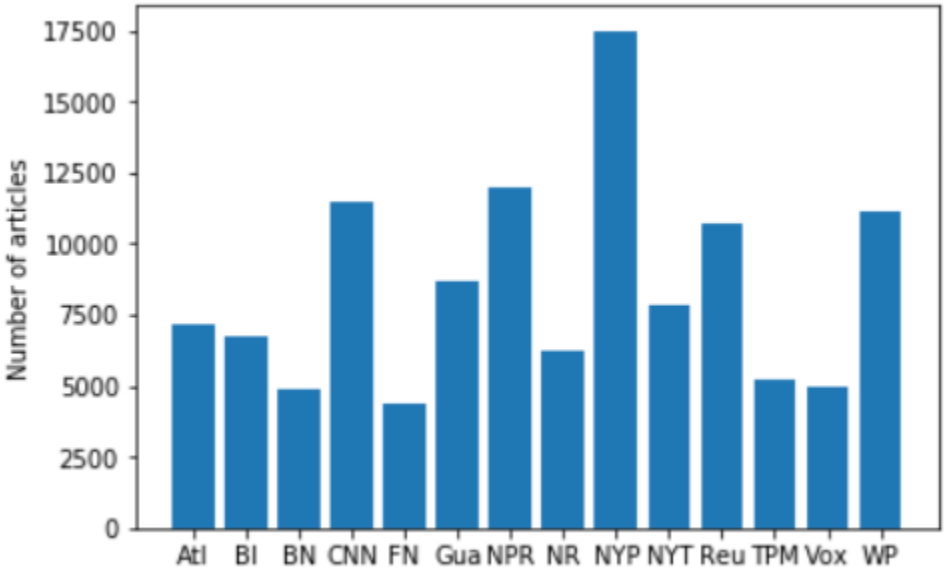
Interestingly, the highest peak changes every day: typically the page is not even on the top 100 most viewed ones until the spike is at its maximum, and they can die quite soon. In some cases it's a single day event, in other cases it can last for days. Filtering the pages with over 500.000 views at the peak we can observe well defined peaks that correlate to some news events, like the case of Alan Rickman or Supreme Court Justice Antonin Scalia.



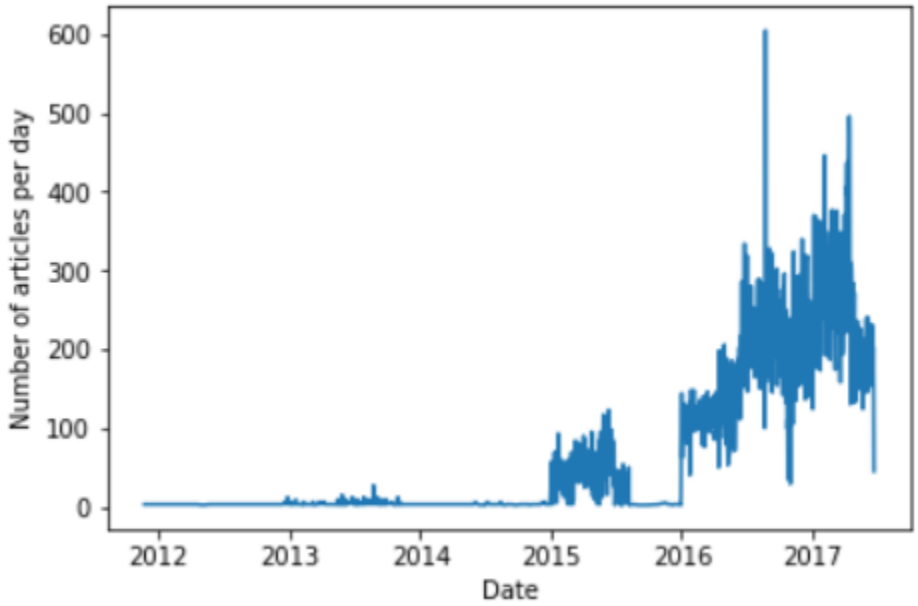

```
Peak order
['Natalie_Cole', 'María_Santos_Gorrostieta_Salazar', 'Nuclear_program
_of_Saudi_Arabia', 'Bikini', 'George_Raft', 'Supersonic_Low_Altitude_
Missile', 'Iman_(model)', 'Alan_Rickman', 'Xerostomia', 'Frederick_Do
uglass', 'Pride_and_Prejudice', 'Warzone_2100', 'Christopher_Paul_Nei
l', 'Omayra_Sánchez', 'Antonin_Scalia', 'Nicole,_Erica_and_Jaclyn_Dah
m', 'Anaximander', 'Pierre_Brassau', 'Nancy_Reagan', 'Meldonium', 'Lu
pe_Fuentes', 'Merrick_Garland', 'Bluetooth', 'Patty_Duke']
```

5. Initial findings: News dataset

The news dataset from Kaggle contains thousands of articles from different sources, including New York Times, NPR, Business Insider or CNN. The data is mostly clean, contains plenty of advertisements from Breitbart, so I have deleted all Breitbart articles since their validity is under scrutiny recently.



The times in which they were written span different periods, mostly after 2016. To perform an initial analysis I have selected the articles from 2016 and 2017 (over 108.000)



Using the approach described in the sklearn tutorial it is possible to run topic modelling on the data, using NMF (Non-negative Matrix Factorization) and LDA (Latent Dirichlet Allocation). The topics seem very diverse, including politics, the presidential election, international politics or business.

```
Topics in LDA model:
Topic #0: company said court federal government department executive case chief new
Topic #1: news media trump times russia people said president night political
Topic #2: said like time just team new school years make work
Topic #3: percent million year said 000 new years according company united
Topic #4: united states said mr trump american president order obama administration
Topic #5: said new north state police city people mr times york
Topic #6: mr trump said president house white new campaign people washington
Topic #7: health republicans said care law china republican mr house trump
Topic #8: people women like said just think don know going new
Topic #9: ms said mr new family years time york like people
```

Regarding the performance, I have implemented a topic coherence function that returns the average of the similarity between each pair of words that define the top N words in the topic. The total coherence is the mean of the coherence of all topics. Other measures like the log-likelihood are harder to quantify and don't agree with the human interpretation, which is what we are trying to imitate. In our case, LDA outperforms NMF.

6. Future work:

Now the data is ready to be analyzed in full and put together. I will use the API to find the spikes for the period January 2016 to June 2017. To correlate the two datasets I will use either the Wikipedia articles' title or the whole body to find the distance to the articles from the news. I will use the cosine similarity as the measure for correlation.

I will train the LDA and NMF algorithms with different parameters to see which one performs better, and use the best as default.