Project Presentation

Data Mining AA 2022 / 23

Alessandro Bucci
Giacomo Cignoni
Alberto Roberto Marinelli

Datatype Casting

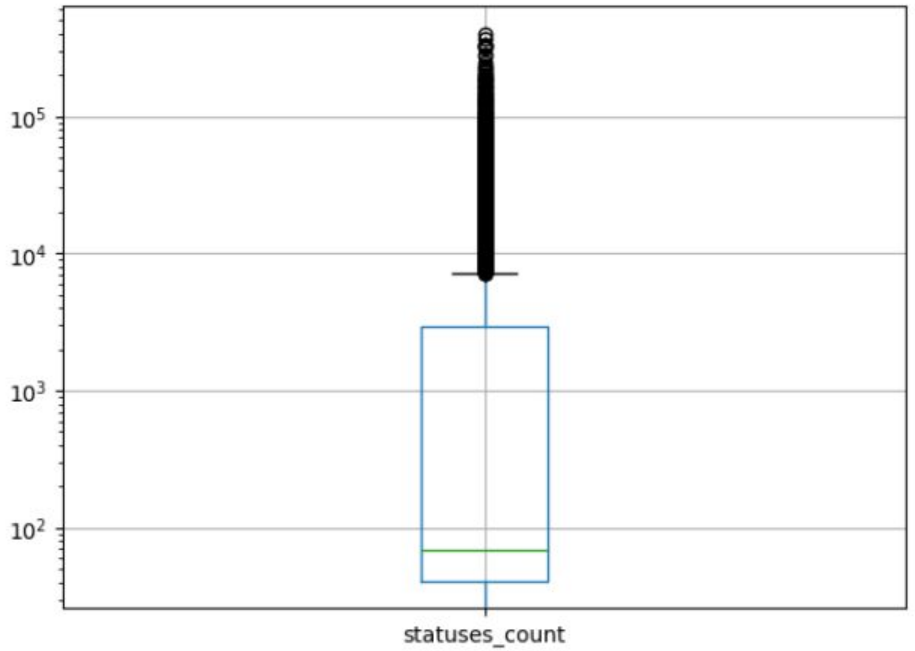
Both Users and Tweets dataframe features were cast to datatypes coherently with the **semantic** of the features.

#	Column	Dtype
0	id	object
1	user_id	object
2	retweet_count	object
3	reply_count	object
4	favorite_count	object
5	num_hashtags	object
6	num_urls	object
7	num_mentions	object
8	created_at	object
9	text	object

Dtype
Int64
Int64
Int64
Int64
Int64
Int64
Int64
Int64
datetime64[ns]
string

Outliers and attributes distributions

Analyzed features distributions with **boxplots**, most of the features have a “long tail”.



Substituted with NaN the **outlier features** having impossible values (e.g. more urls in a tweet than the number of characters or dates in the future).

Manage Duplicates

Idea

Initially all rows that were **exactly identical** were **removed**.

Then, assuming that **id** and **user_id** are **unique** in Tweets dataset, the idea was to extract the rows with the same values for these attributes in order to **recover the damaged values** by comparing duplicates.

	id	user_id	retweet_count	reply_count	favorite_count	num_hashtags	num_urls	num_mentions	created_at	text
80220	199269133416792064	270494010	0	0	0	0	1	1	2017-05-08 22:47:18	Ik heb een video toegevoegd aan een @YouTube-a...
80221	199269133416792064	270494010	0	0	0	0	1	1	2017-05-08 22:47:18	Ik heb een video toegevoegd aan een @YouTube-a...
53334	458097180310400960	385121466	1	0	0	0	0	1	2019-04-23 04:17:33	RT @jameon_heard: Having to beat around the bu...
53335	458097180310400960	385121466	1	0	0	0	0	1	2019-04-23 04:17:33	RT @jameon_heard: Either wayy
44223	473413979579351040	1395978913	0	0	0	0	0	0	2019-06-04 10:41:03	the universe always has a plan
...	...	...	...	...	...	...	...	...	...	...
27787	<NA>	3159993463	<NA>	<NA>	<NA>	<NA>	<NA>	0	2020-04-25 02:03:35	Renovation hell continues. Fifth day....

Manage Duplicates

No results but **new information discovered**

No value was corrected because, when analysing the result of this row-by-row matching operation, the **texts and related data of the Tweets were completely different**, so an error was assumed in the extraction of the Tweet id during crawling, and several **Tweets with the same id were found**.

	id	user_id	retweet_count	reply_count	favorite_count	num_hashtags	num_urls	num_mentions	created_at	text
80220	199269133416792064	270494010	0	0	0	0	1	1	2017-05-08 22:47:18	Ik heb een video toegevoegd aan een @YouTube-a...
80221	199269133416792064	270494010	0	0	0	0	1	1	2017-05-08 22:47:18	Ik heb een video toegevoegd aan een @YouTube-a...
53334	458097180310400960	385121466	1	0	0	0	0	1	2019-04-23 04:17:33	RT @jameon_heard: Having to beat around the bu...
53335	458097180310400960	385121466	1	0	0	0	0	1	2019-04-23 04:17:33	RT @jameon_heard: Either wayy
44223	473413979579351040	1395978913	0	0	0	0	0	0	2019-06-04 10:41:03	the universe always has a plan
...	...	...	...	...	...	...	...	...	...	...
27787	<NA>	3159993463	<NA>	<NA>	<NA>	<NA>	<NA>	0	2020-04-25 02:03:35	Renovation hell continues. Fifth day....

Manage Null

Users DataFrame

For the users dataframe, the **median** was used to replace null values, in particular distinguishing between **bots** and **non-bots**.

Tweets DataFrame

In this dataframe, for each user with *more than 20 Tweets*, the **median** was calculated **specifically**. Otherwise, due to the limited data available to make a good estimate of the median, this value was calculated on **bots** and **non-bots** as in the previous dataframe.

Indicator extraction

For each user:

retweet count
reply count
favorite count
number of hashtags
number of urls
number of mentions

- Average and Sum
- Total success score

For each tweet of the user

- Count
- Count with text
- Tweets in a year
- Tweets outside of plausible years

- Average text length
- Creation year
- Entropy

Indicator extraction

Entropy is obtained by using the unique **timedeltas** (the time passed between the publishing of each tweet) as categories.

```
2022-01-01 02:00:00 - 2022-01-01 01:00:00 = Timedelta('0 days 01:00:00')
```

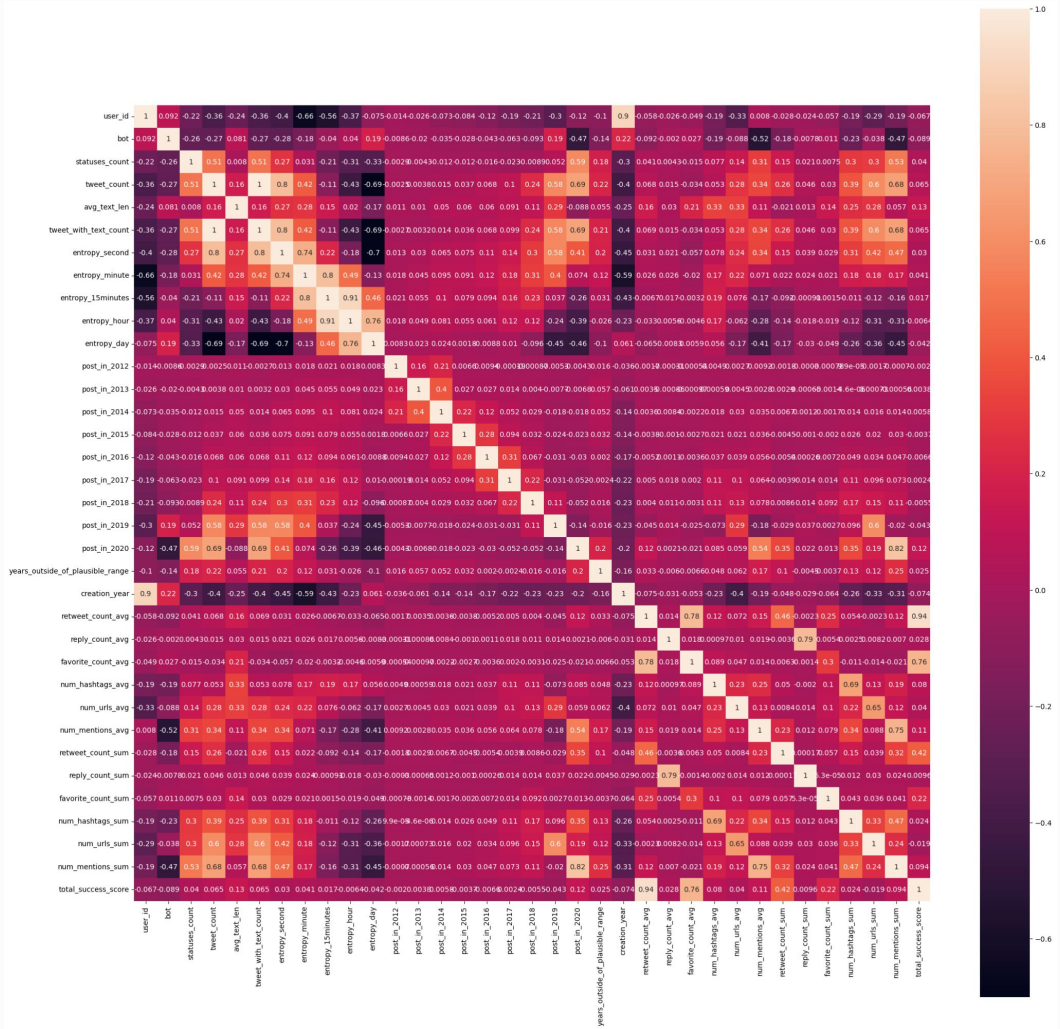
The precision used are **second**, **minute**, **quarter of an hour**, **hour**, **day**.

Total success score is obtained by the formula:

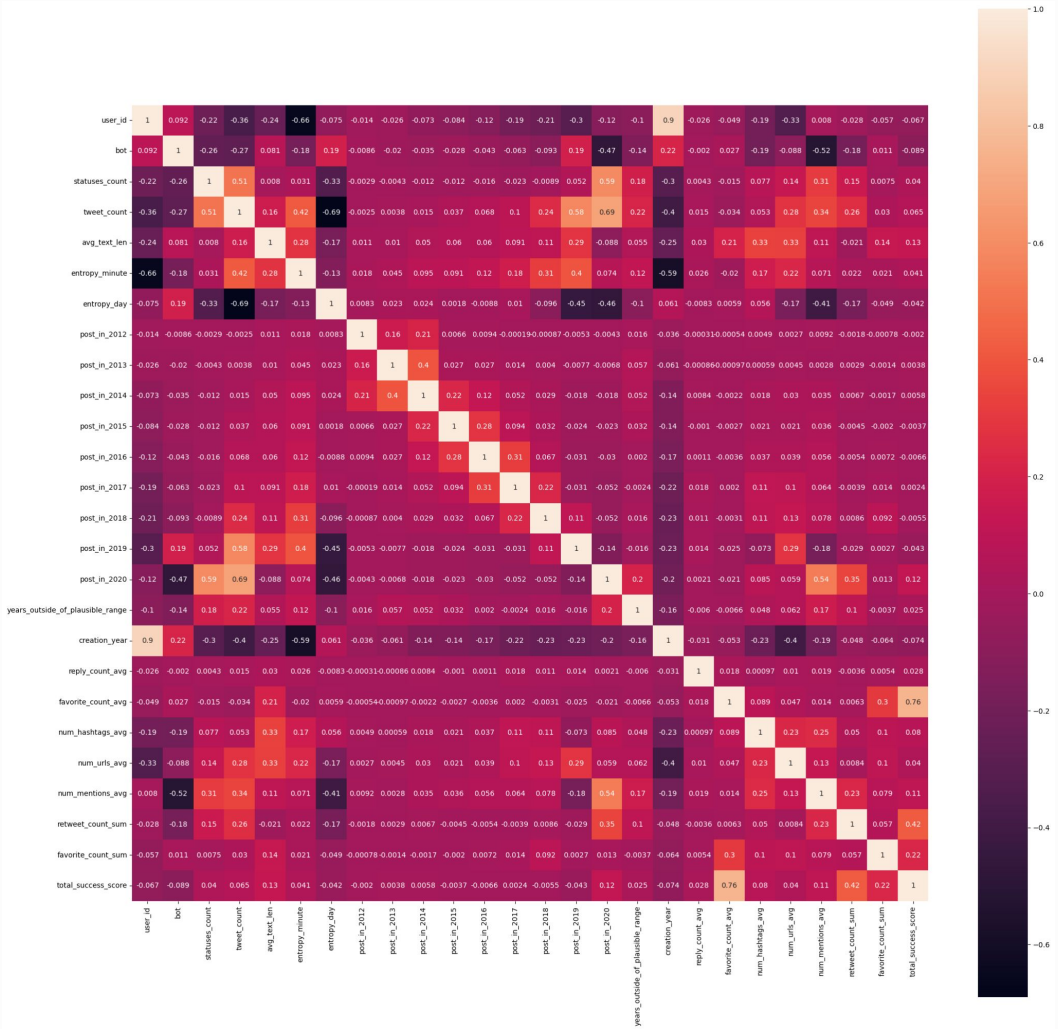
$$\text{Total Success Score} = \frac{\text{Total Acceptance Score}}{\text{Total Diffusion Score} + 0.1}$$

Where the **Total Acceptance Score** is the sum of **retweet_count_sum**, **reply_count_sum** and **favorite_count_sum**
and **Total Diffusion Score** is the sum of **num_hashtags_sum**, **num_mentions_sum** and **num_urls_sum**.

All attributes Correlation map

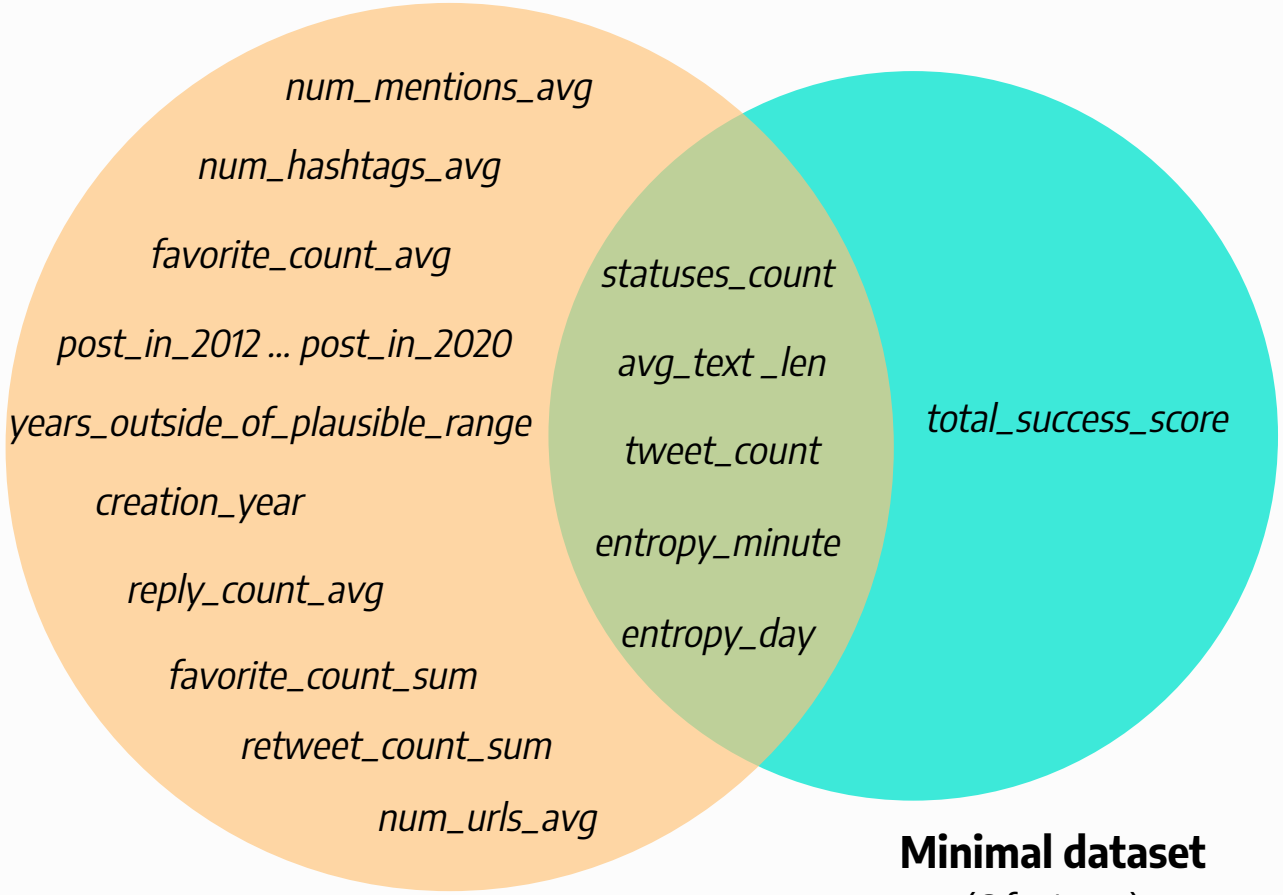


Uncorrelated attributes Correlation map



Clustering

Feature sets



Full Dataset
(23 features)

Minimal dataset
(6 features)

Preprocessing

1.

Log-scale reduction

Due to some attributes having some excessively **high values**, it was necessary to apply a reduction in **log-scale values**.

2.

Normalization

From each dataset (full and minimum), **two sub-datasets** of data were produced corresponding to the normalizations applied. This practice allow to **avoid the bias** given by the range of the different attribute.

- Z-Score
- Min-Max

DBSCAN

Best Result

- Dataset: **minimal**
- Normalization: **Min-Max**
- Clustering configuration: **0.03 eps** and **10 minpts**
- Cluster sizes:

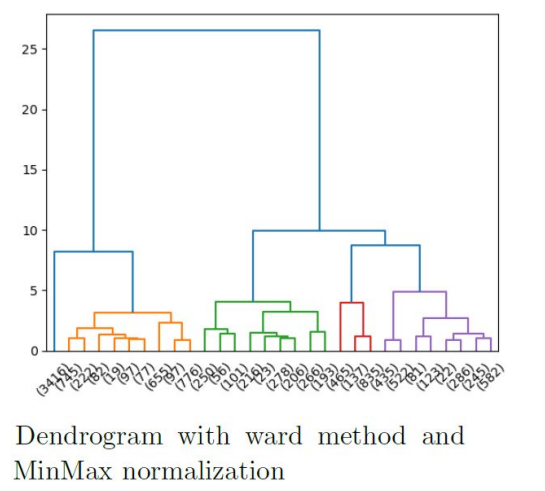
[**4398**, **3155**, 270, **822**, **2055**, 37, 392, 75, 200, 57, 24, 23]

Using the *'bot'* data, it was noted that clusters in this configuration almost always have a **well-defined distribution** of only bots or only users. Despite this, half of the **clusters are very small in size**.

Hierarchical clustering

Best Result

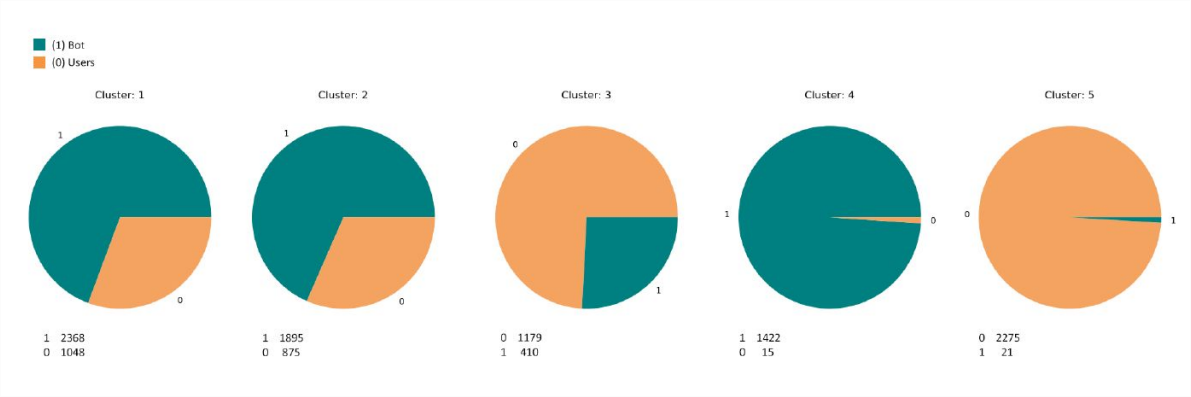
- Dataset: **full**
- Normalization: **Min-Max**
- Distance metric: **cosine**
- Method: **ward**
- Dendrogram:



Dendrogram with ward method and MinMax normalization

Hierarchical clustering

- Cluster sizes and distributions:



The analysis of each of these results using **radar plots** revealed some interesting features, such as:

Clusters 4 and *5* have **higher favorite sum**, but *Cluster 5* of humans have the **highest retweet sum on average**, indicating that humans gets more retweets.

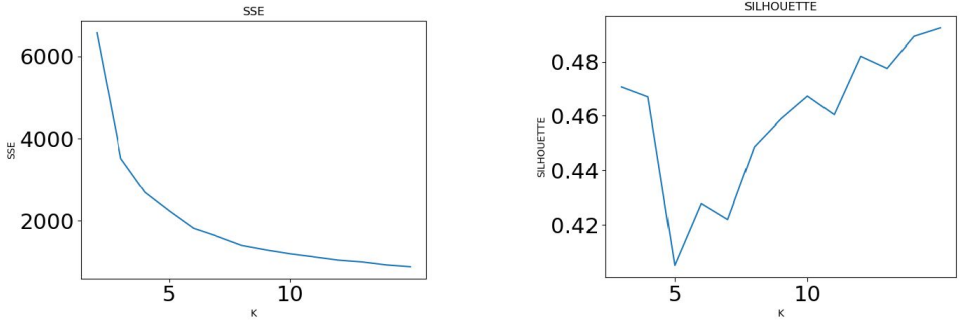
Mostly bots have a lot of tweets from *2019*, *2018* and *2017*, while clusters with mostly humans have tweets mostly in *2020*.

K-Means

Best Result

- Dataset: **full**
- Normalization: **Min-Max**
- K-Value: **7**
- Cluster sizes:

[1038, 3678, 987, 473, 1204, 2444, 1684]



(a) K-Means SSE graph **(b)** K-Means Silhouette graph

Figure 4: K-Means graphs for SSE and Silhouette

X-Means

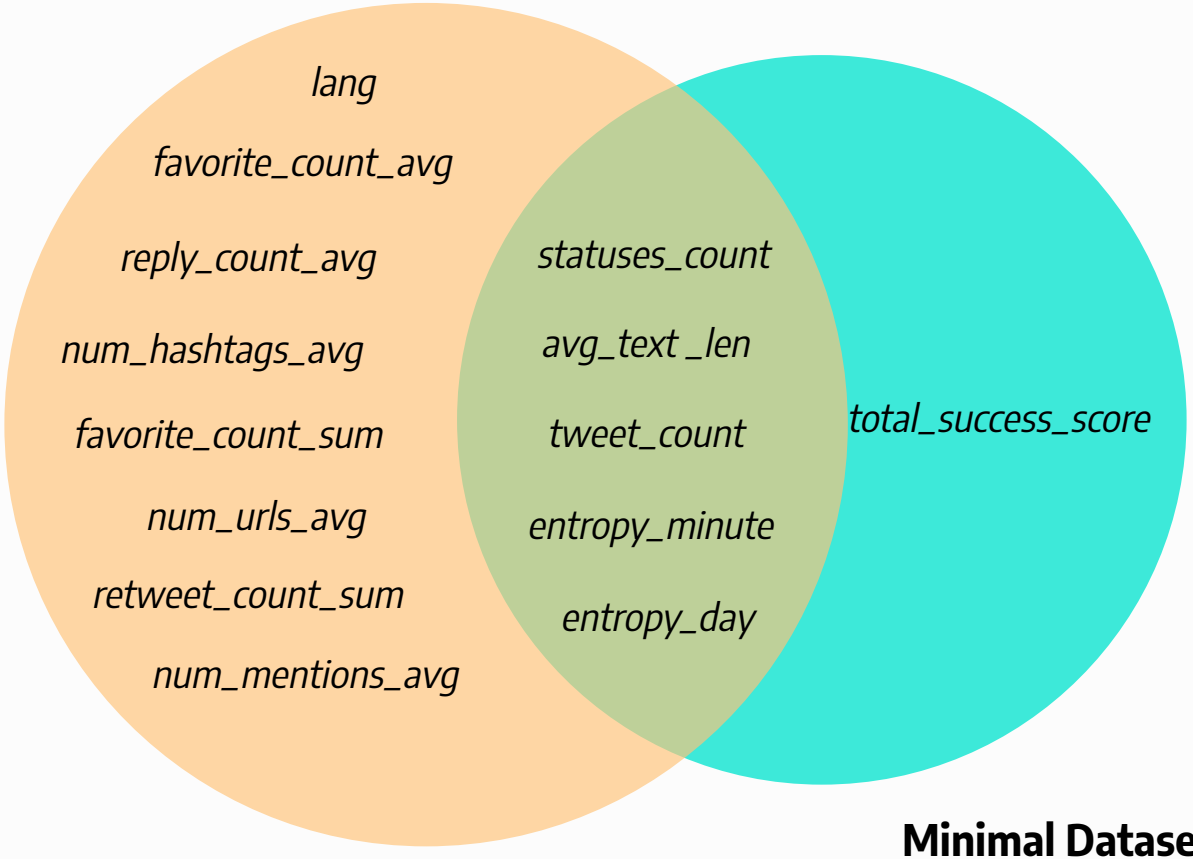
Results

We tried the **X-Means** implementation from Pyclustering library, with **Bayesian Information Criterion** (BIC) splitting criterion. We used *2* as the minimum number of clusters and a variable between *20* and *100* as the maximum. Centroids were initialized with K-Means++ heuristic.

With both **complete** and **minimal** datasets configurations, **X-Means fails** to achieve a good clustering, always selecting as the number of clusters the allowed maximum.

Classification

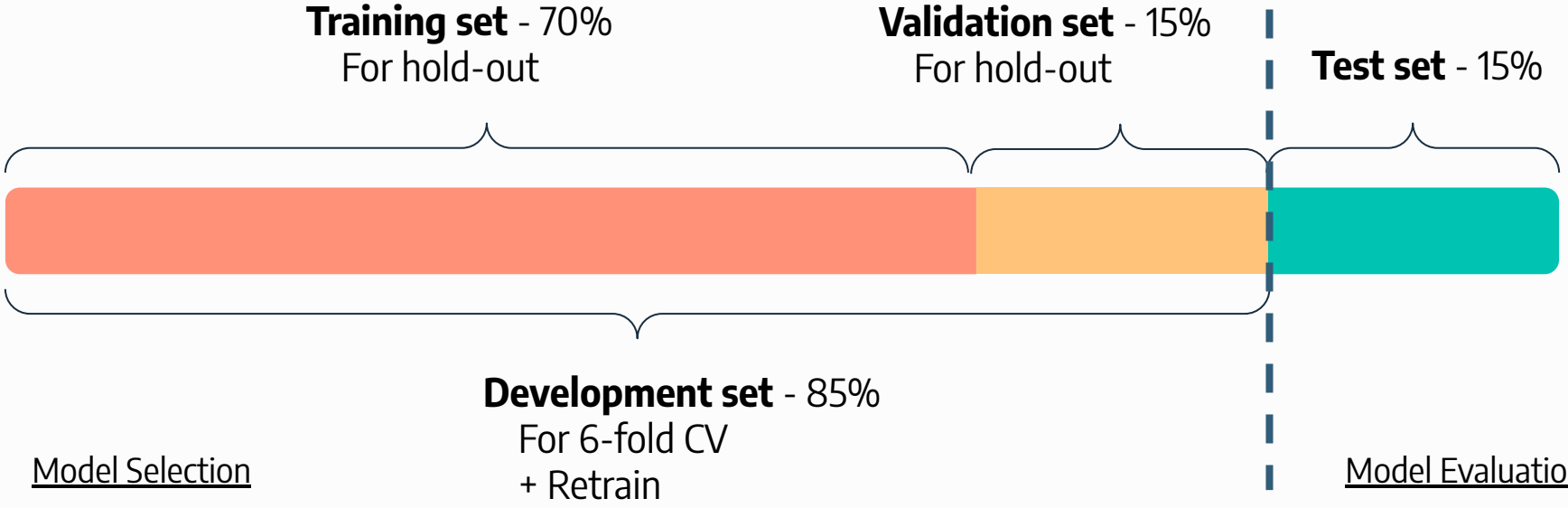
Feature sets



Full Dataset
(13 features)

Minimal Dataset
(6 features)

Train-test split



Models and Results

	Validation approach	Minimal Test accuracy	Full Test accuracy
KNN	hold-out	0.80	0.82
Bayesian classifier	hold-out	0.65	0.76
Decision Tree	6-fold CV	0.84	0.84
Random Forest	6-fold CV	**0.85**	**0.85**
SVM	6-fold CV	0.84	0.84
MLP	hold-out	0.84	**0.85**

- **Low recall** in all models for non-bots.
  That means significant number of bots classified as non-bots.

- **High precision** in all models for non-bots.
  That means most of non-bots are classified correctly.

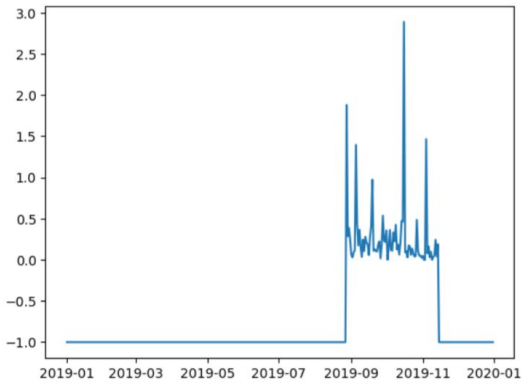
- In minimal feature set, we chose **Bernoulli instead of Gaussian** Bayesian classifier even if the latter had a higher validation accuracy, as Gaussian had a non-bot recall <0.5.

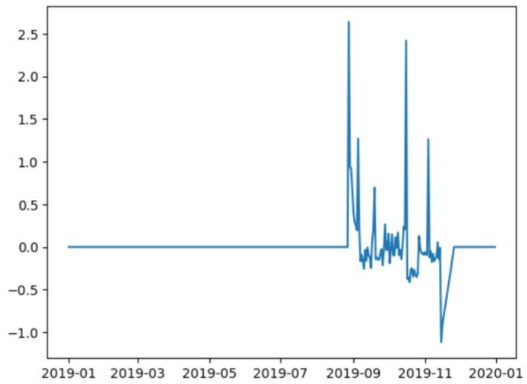
Extra considerations

Time Series

Preprocessing

Detrend



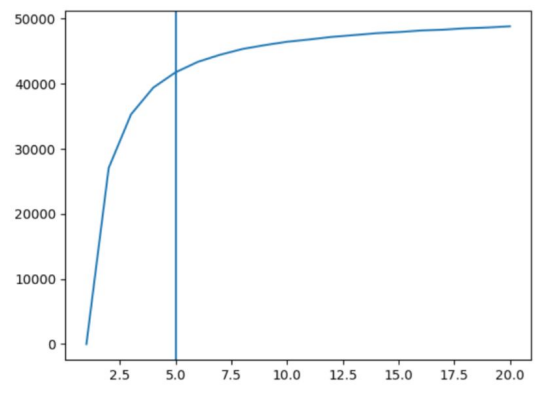


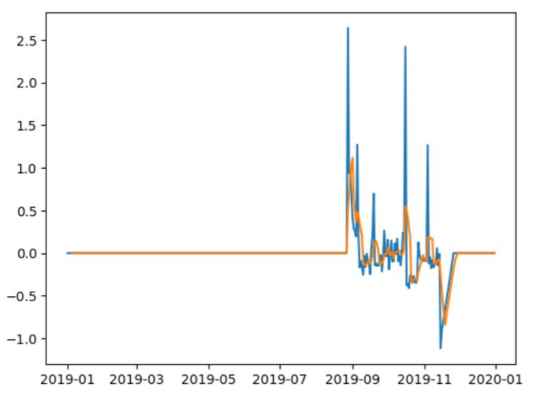
We used a Unit Root Test (**Augmented Dickey-Fuller**), a statistical test where we can say that a series is non-stationary if it does not have a unit root. That is expressed if the p-value is less than the significance level (0.05).

To correct them, we subtract a moving average of the series from the original series.

Preprocessing

Denoising

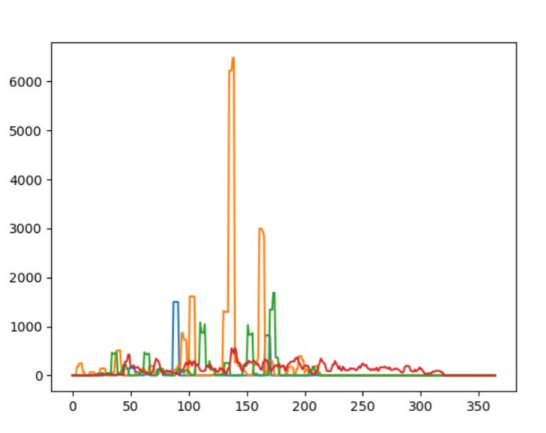


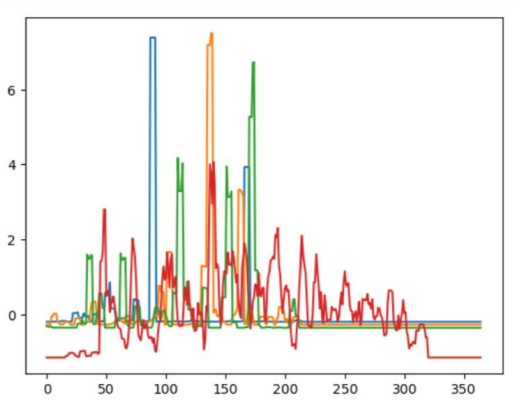


We used the **Knee method** on the mean of the **sums of absolute differences (SAD)** between the original time series and the smoothed one, choosing the best window that ”doesn’t smooth too much” the original time series.

Preprocessing

Normalization

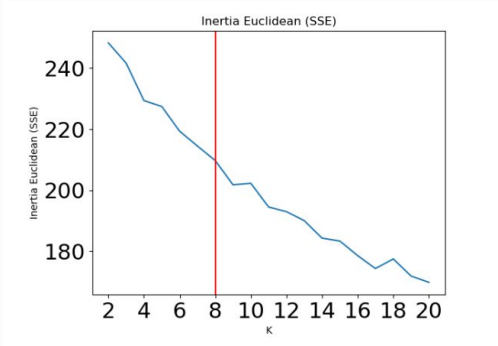




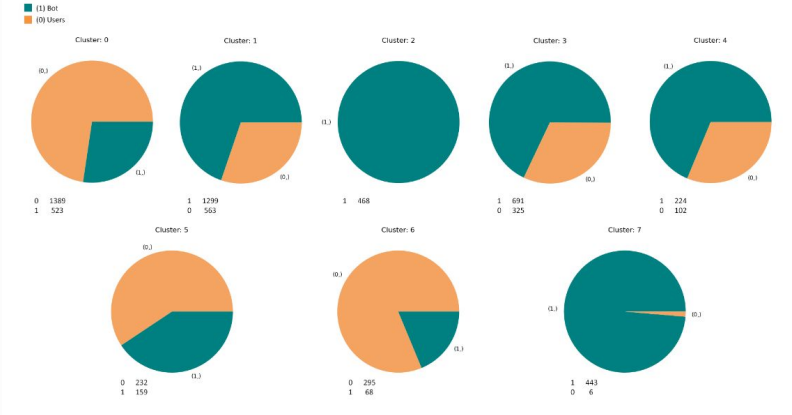
For the normalization we used the Mean-Variance scaler.

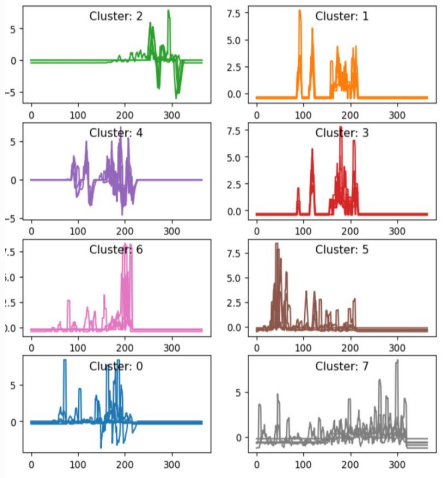
Chosen Clustering

Shape-based K-Means



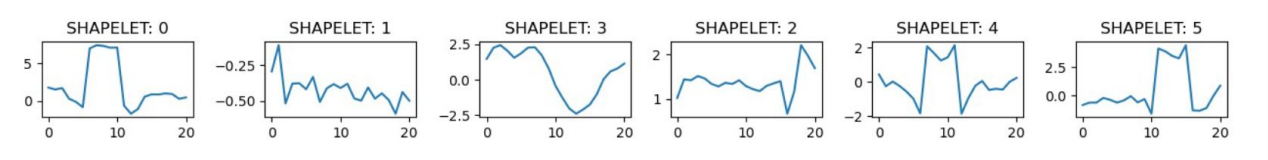
Although there is not any clear curve to apply the elbow method, **K=8** was the best as it had a balanced number of clusters in terms of SSE and cluster sizes.



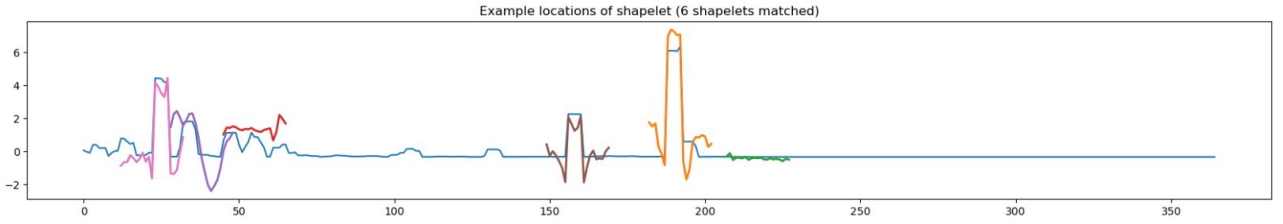


Shapelets

For extracting the shapelets we resorted to the Grabocka method, using as size of the shapelet **21** and found **6** shapelets.



Then we trained a Shapelet model consisting of a time series transformation and a logistic regression layer on top. Using this model, we are able to analyse the time series by **observing the number and position of shapelets matched**.



Thanks for your attention!

Alessandro Bucci
Giacomo Cignoni
Alberto Roberto Marinelli