Group 9

- Andrea Iommi - 578212
- Irene Pisani - 560104

Project presentation

Data Mining 2022/23
M.Sc. Computer Science, Artificial Intelligence
University of Pisa

TASK 1 – Data U&P

- Improving data quality of *User* and *Tweet* datasets
- Extract new indicator for describing Twitter users behavior

Data U&P

Correctly cast attributes types

➤ USERS and TWEETS DATASETs

Feature	Type	Correct type
user_id	int	int
name	object	string
lang	object	string
real	int	bool
created_at	object	datetime64
statuses_count	float	float

tweets.csv (casting after cleaning phase)

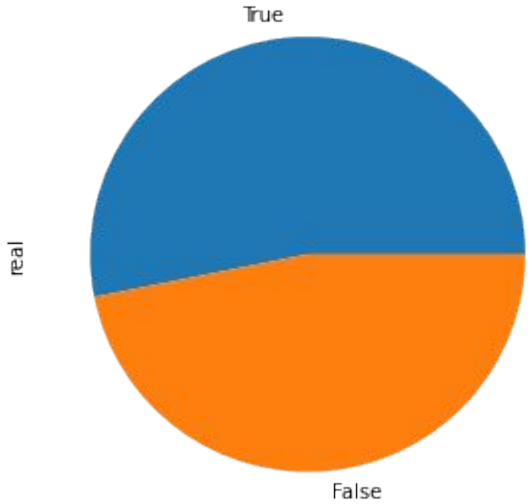
Feature	Type	Correct type
id	int	int
user_id	object	int
retweet_count	object	float
reply_count	object	float
favorite_count	object	float
num_hashtags	object	float
num_mentions	object	float
created_at	object	float
text	object	string

Data U&P

Features insights for *Users* dataset

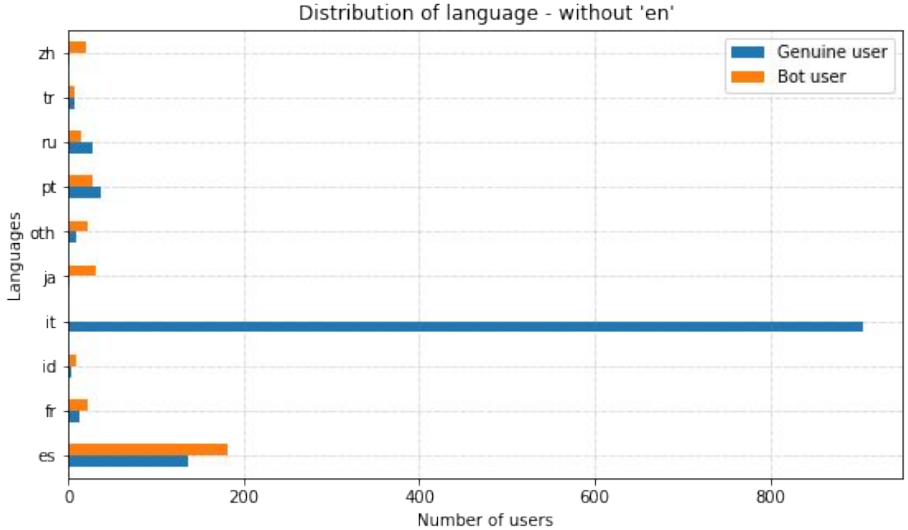
➤ FEATURE: `real`

Genuine 0.53% Bot 0.47%



➤ FEATURE: `lang`

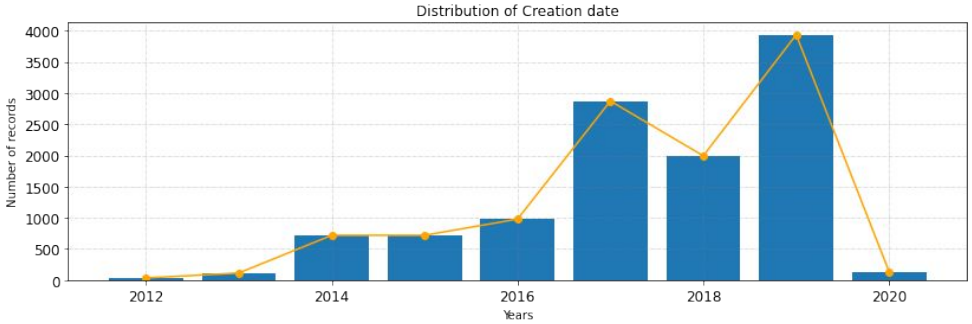
Distribution of languages without “en”

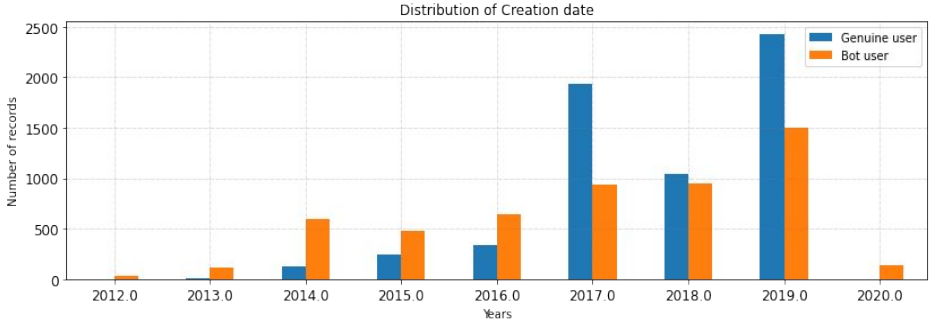


Data U&P

Features insights for *Users* dataset

➤ FEATURE: `created_at`



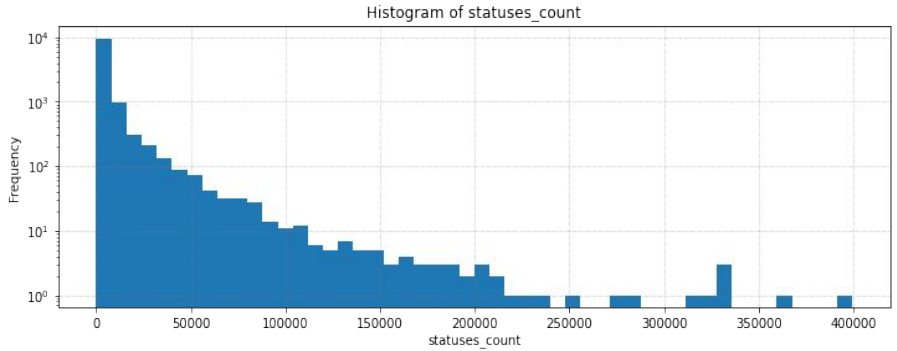


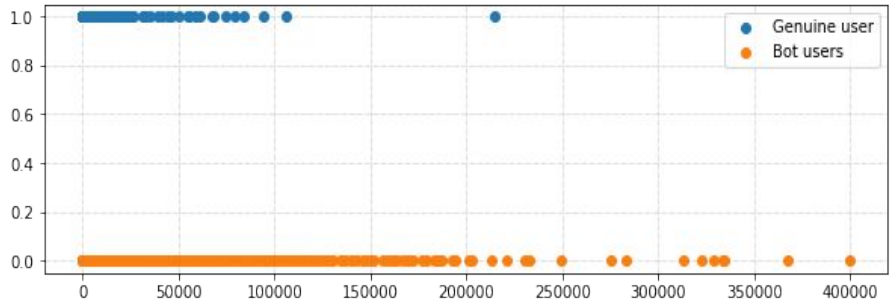
- On top there is the number of users enrolled to twitter through years, we discovered an i**ncreased tendency** (till 2020).
- On bottom is depicted the same chart but respect to genuine/bot users.
- It's important to point out that these results displayed are influenced by the crawler. E.g. in 2020 there are not genuine users enrolled, this could be explained by the fact that the random algorithm of the crawler didn't pick any genuine users.

Data U&P

Features insights for *Users* dataset

➤ FEATURE: `statuses_count`





- Values distribution highlight that statuses_count has a extremely **positive skewness**.
- Boxplot for genuine and bot users highlight that bot users typically have published **more tweets than genuine users**.

Data U&P

Cleaning procedure performed on *Tweets* dataset

1	Deleted rows with **Null id**, since there is no way to connect them to a user.	217283
2	Deleted **duplicated** tweets (same `user_id, created_at` and `text`) since don't add any further information.	2999450
3	Deleted **Not numeric id,** for the same reasoning at point one.	216730
4	Deleted Tweets with all **zeros values** and null text.	139985
5	Deleted tweets that **aren't associated** to a user in twitter's users.	0
6	Deleted **outliers** based on Sigma rule.	33
	TOTAL	3573481

Data preparation

Extract new features from *Tweets* dataset for improving *User* dataset

A = Average S = Standard dev. T = Total F = Favorite E = entropy

Achars	Sreply	Ayear	itr_sco
Schars	Ahash	Amonths	Echars
Alike	Shash	Adays	Elike
Aemoji	Thash	Fyear	Ereply
Aretwee	Aurls	Fmonth	Epop_sco
Sretwee	Aments	Fday	Eint_sco
Areply	Sments	pop_sco	Eintervarls

Discarded due to correlation with other features

- chars = # characters in text
- emoji = # emoji
- year = # tweets per year
- months = # tweets per month
- day = # tweets per days
- pop_sco = popularity score
- itr_sco = Interactivity score
- Eintervals = entropy of time elapsed among tweets published

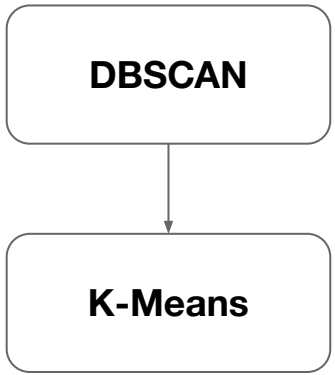
TASK 2 - Clustering

- 2-step clustering technique (DBSCAN + K-Means)
- Hierarchical clustering (agglomerative approaches)
- Additional clustering (X-Means and SOM)

2 step-clustering technique

DSCAN + K-Means: a combined application

➢ PROCEDURE DETAILS



1. Firstly, use DBSCAN clustering to **detect noise points/data** and to have a insight how much number of clusters are suitable.

2. Then we applied Kmeans to dataset where the noise points (detected by DBSCAN) have been removed to extract more meaningful clusters.

➢ MOTIVATIONS

- Early experiments shows that DBSCAN algorithm seems to not working well in this context.

- We were able to adopt this alternative approach since the **removal of outliers** in data preparation was performed with a very **gentle and minimalist approach**.

Reference

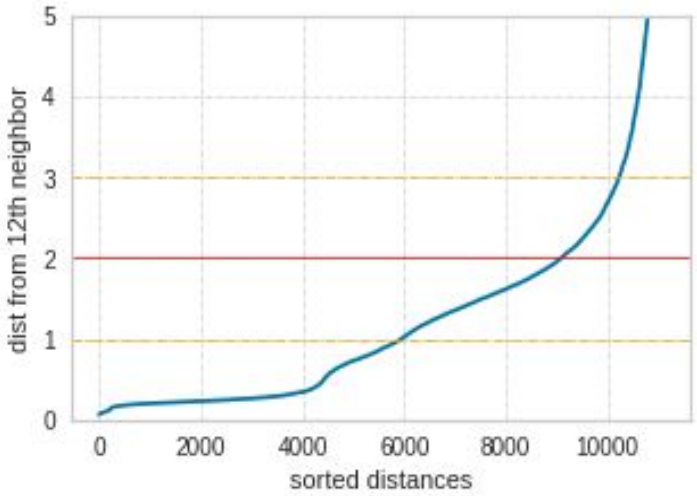
Xue Liu, Yong Ding, Hao Tang, and Feng Xiao. *A data mining-based framework for the identification of daily electricity usage patterns and anomaly detection in building electricity consumption data*. Energy and Buildings, 231:110601, 2021.

DBSCAN clustering

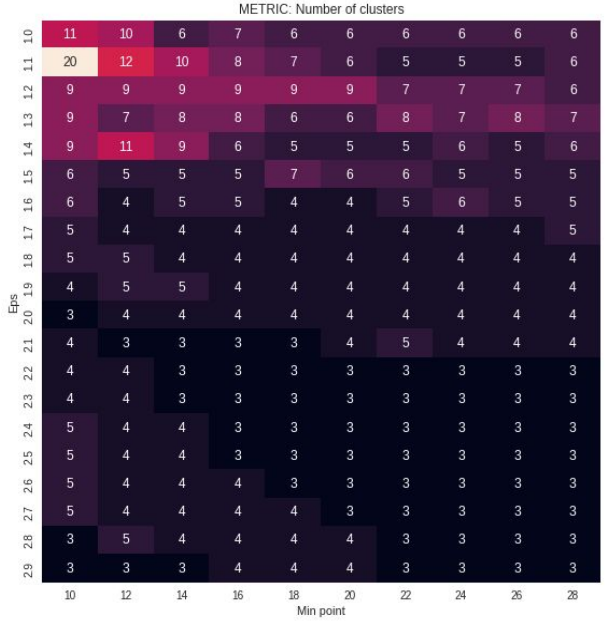
Explore hyper-parameter space to identify optimal values

➤ KNEE METHOD

Detect the optimal **eps** value.



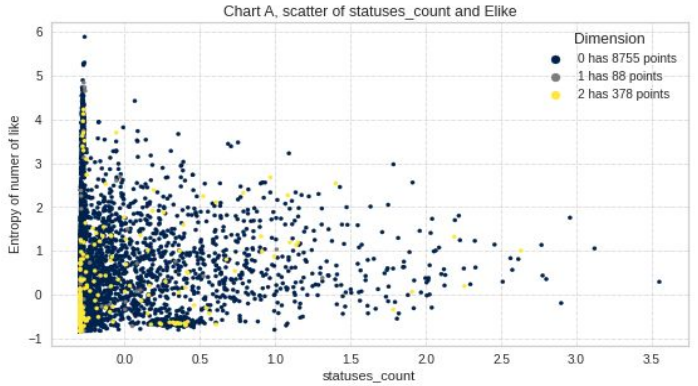
➤ CONFIGURATION HEATMAP

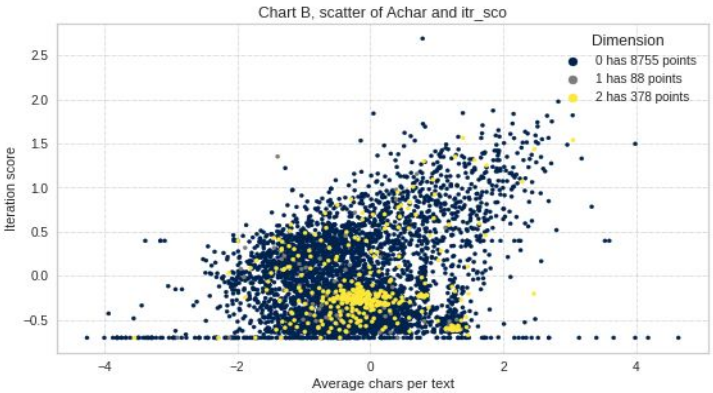


DBSCAN clustering

Clustering analysis and visual interpretation

- Noise cluster is omitted to get a cleaner view.
- In the chart A, the cluster 2 has a low number of posts published and the entropy of like received covers the range of values. We could associate the cluster 2 to a group of user that publish **few but relevant tweets**.
- In the chart B, we can observe that the cluster 2 is concentrated at **60 chars per tweets** and has l**ow interactivity score**. This situation could be plausible typically for the genuine users.
- In the cluster 0 and 1 we aren't able to identify a particular behaviour/pattern among all charts generated. In Cluster 2 we observe the behavior is typically associated with a classical user in twitter platform.



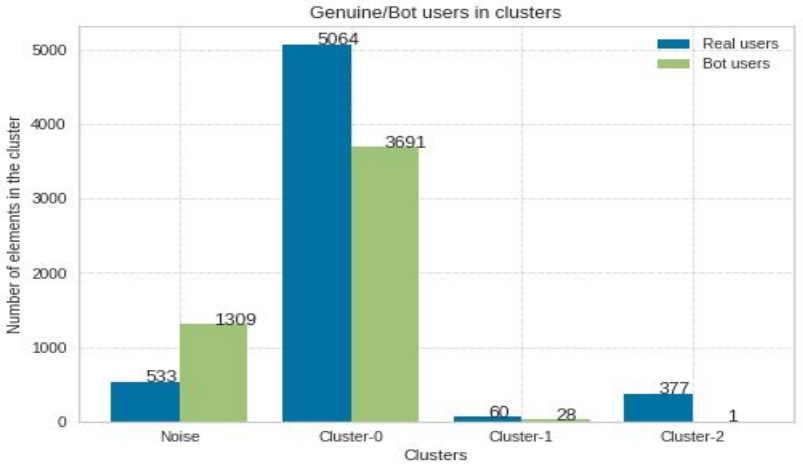


DBSCAN clustering

Clustering characterization and evaluation

➢ GENUINE/BOT INTERPRETATION

- The **cluster 2 is composed only by genuine users,** this confirm that we have found a typically behavior of genuine user.
- In all clusters except of Noise there is a **genuine users predominance**.
- The most of bot users are grouped into cluster 0.



➢ CLUSTERS VALIDITY WITH INTERNAL METRICS

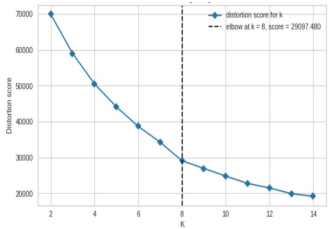
Davies Bouldini	2.087
Silhouette	0.2064
Calinski Harabasz	505.5334

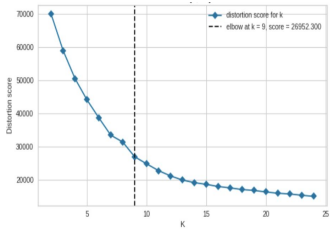
- Davies Bouldini affirm that the **clusters are not well separated**, Silhouette shows low cohesion among clusters, Calinski Harabasz confirm the **overlapping**.
- We can conclude that even if some pattern are revealed, DBSCAN is not the best solution to discriminate groups inside the dataset.

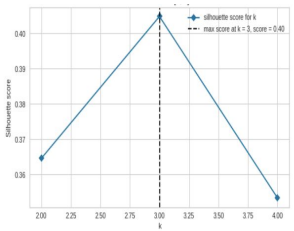
K-Means clustering

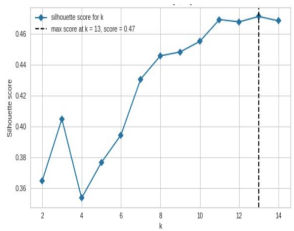
Identify optimal number of clusters K

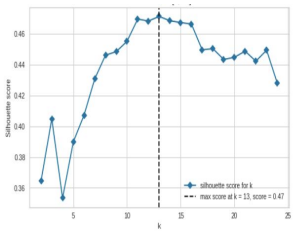
- **Elbow method with SSE** by considering incremental range of k values. Obtained candidate k values: **8,9**.
- **Silhouette method**. incremental range of k values. Obtained candidate k values: **3, 13**.
- **Ward linkage method**. Obtained candidate k value: **2**.











➤ ASSESS CLUSTERS VALIDITY WITH INTERNAL METRICS

K value	SSE	Davies-Bouldini	Silhouette	Calinski-Harabasz
2	69969.311	1.515	0.364	**3181.567**
3	59030.091	1.309	0.404	2739.499
8	29097.497	**1.013**	0.445	2940.946
9	26952.299	1.049	0.448	2869.494
13	**19922.963**	1.101	**0.471**	2857.561

- **Optimal k = 2** (following Calinski-Harabasz)
- When computing scores on the not normalized version of our dataset all 4 metrics agreed in indicating 2 as the optimal k value.

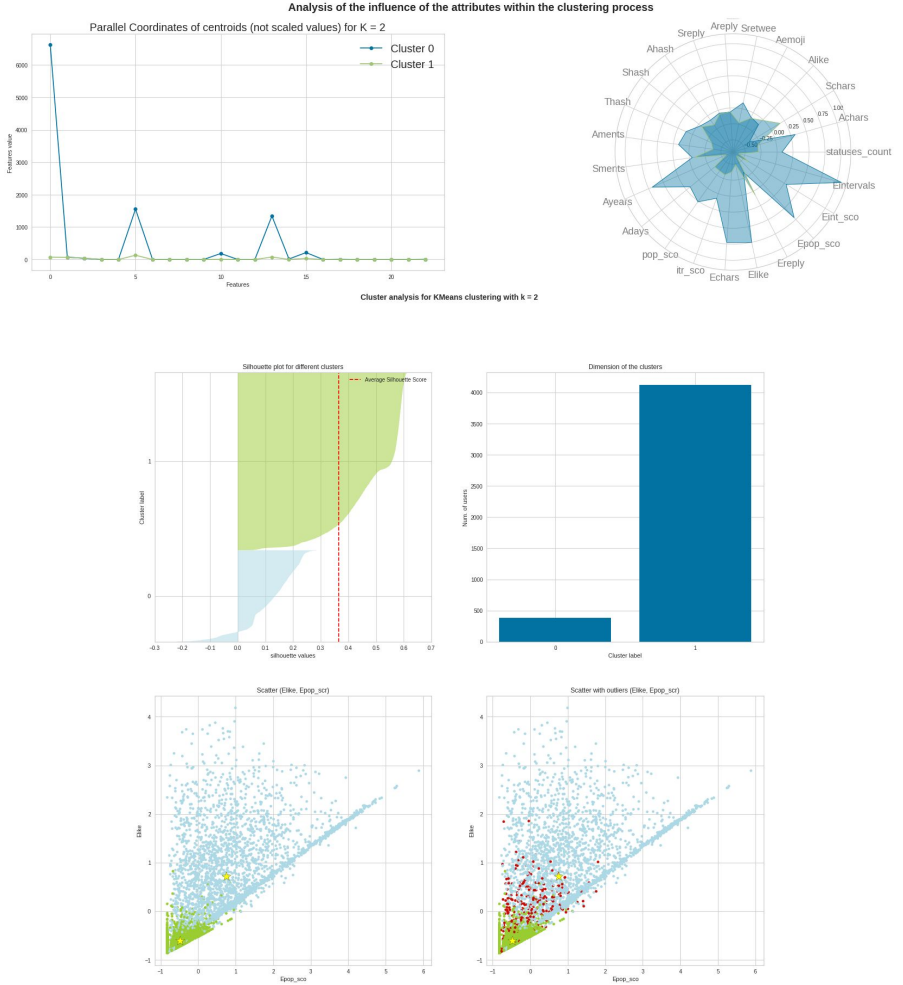
K-Means clustering

Clusters composition

- 2 cluster differs in dimensions as well as in cohesion and sparsity degree. Cluster 1 seems to be a dense cluster where cluster 0 seems to be more sparse. The cluster are overlapping each other (thus leading to low degree of separation). Points with negative value of silhouette can be treated as noisy points.

- By analyzing the coordinates of cluster centroids we identify the most influential attributes: `statuses_count`, `Sretwee`, `Ayears`. Observed frequent trend: for most features the coordinate of centroid 0 has significant higher value then coordinates of centroid 1.

- A characterization step has been carried out w.r.t. the categorical attribute (`real` and `lang` `Fyear`, `Fday`, `Fmonth`) but the clustering process seem to follow anyone of this partitions given by.

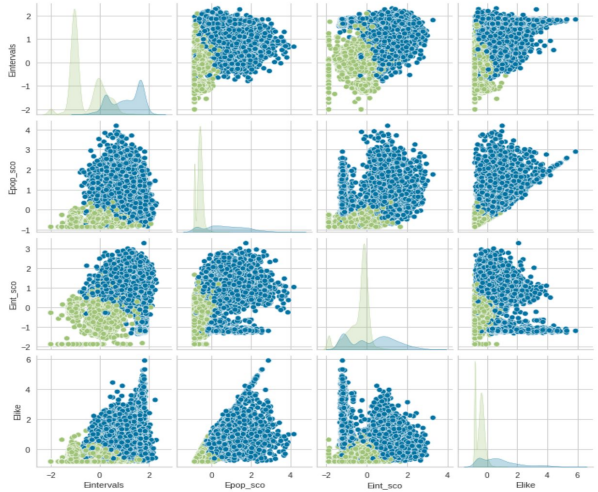


K-Means clustering

Clusters characterization

➤ TRENDS DETECTION WITHIN CLUSTERS

- The cluster 1, the more **dense and cohesive**, contains those points described by l**ow feature values**: low popularity score, low interactivity score, low number of tweets posted in certain period etc.
- The cluster 0, the **most scattered and least dense**, contains those points described by **high feature values**: high popularity score, high interactivity score, high number of tweets posted in certain period etc.



Stats	statuses_count	Epop_sco	Eint_sco	Eintervals
Mean	2773.364	2.041	1.216	6.755
Min	509.959	-0.044	0.351	2.557
50%	1462.371	1.929	1.299	7.032
Max	23710.039	6.689	2.770	9.526

(a) Statistic for cluster 0

Stats	statuses_count	Epop_sco	Eint_sco	Eintervals
Mean	531.769	0.273	0.856	2.994
Min	506.541	-0.065	0.087	-0.152
50%	521.924	0.282	0.953	2.203
Max	8298.814	2.188	1.930	9.033

(b) Statistics for cluster 1

2-step clustering

Overall results, interpretation and advantages

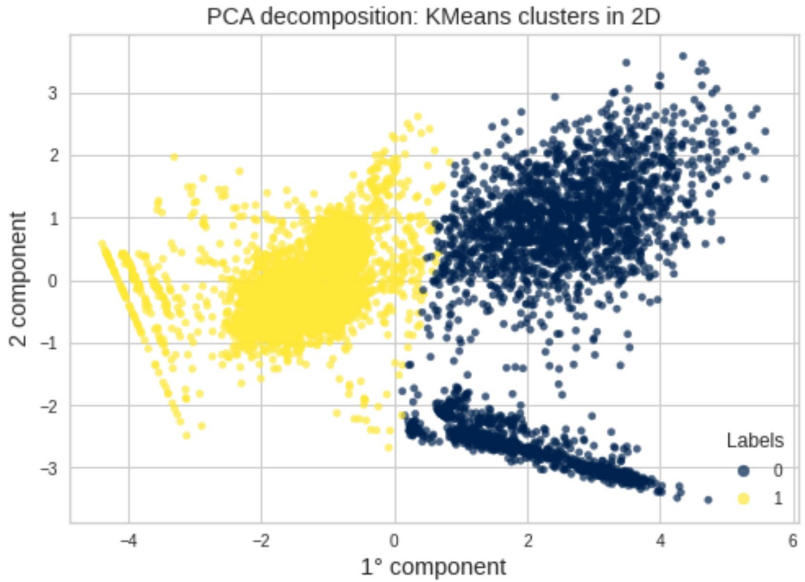
➢ A NEW CLUSTERING PRINCIPLE

- A social score depends on the user's behavior: the more active, popular and interactive the user is the higher its social score will be.
- A larger portion of users are assigned to the cluster that identifies a lower social score.

➢ ADVANTAGES OF 2-STEP CLUSTERING

- **Good visualization** of points partitions in the reduced features space with PCA.
- Allow us to extract 2 significant clusters which are almost **linearly separable** in 2 dimensions.
- This outcomes also gives us **significant and meaningful information** about the social score/social power of the user on the Twitter.

Clustering process assigns a certain point to a cluster based on the magnitude (high or low) of the social score obtained.



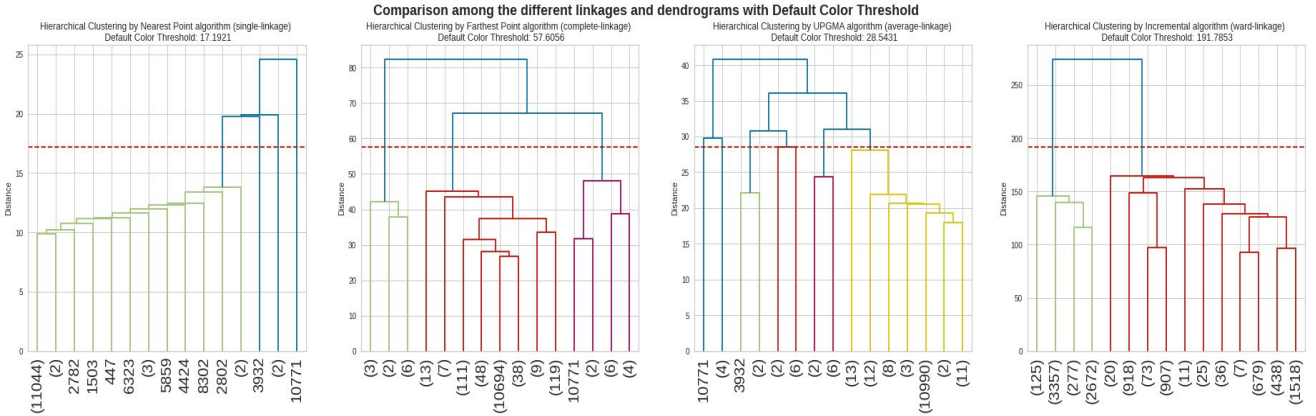
Hierarchical clustering

Compare different linkage method using scipy default cut-threshold

➤ DENDROGRAMS VISUALIZATION AND EVALUATION

The **single, complete, average** linkage method, ending up grouping most of the observations in a single huge cluster.

Only the **2-clusters partition** obtained with the **ward methods** seems to generate a more well-balance and **meaningful points partitions.**

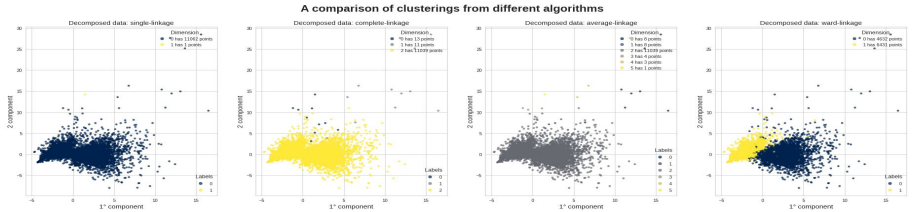


Linkage	cut-threshold	N. clusters	Davies-Bouldin	Silhouette	Calinski-Harabasz	Cophentic
single	17.192	2	0.064	0.905647	146.295	0.889
complete	57.605	3	0.775	0.845	585.918	0.841
average	28.543	6	0.615	0.827	289.504	0.925
ward	191.785	2	1.830	0.281	1913.841	0.373368

Hierarchical clustering

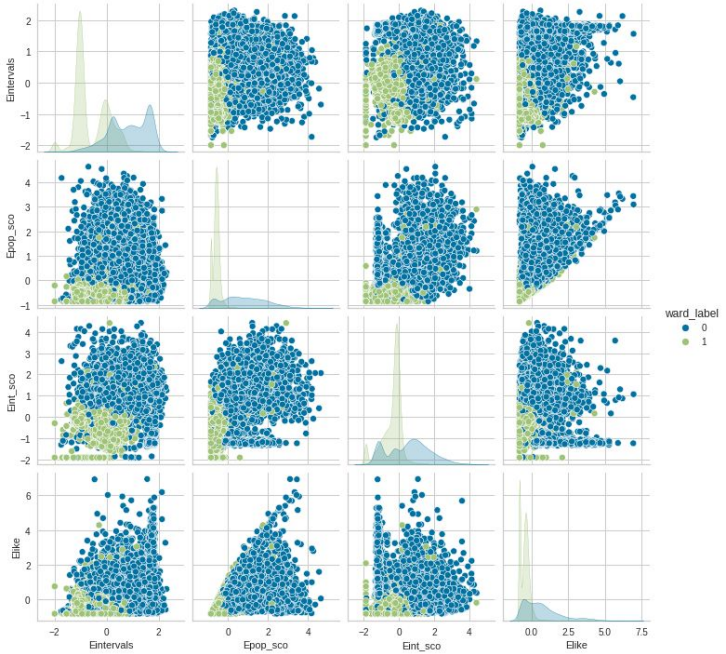
WARD linkage clustering characterization and interpretation

WARD linkage: the only worthwhile solution for performing a results characterization step.



WARD linkage outcomes remarks those one already reached with K-Means:

- Clusters show **more balanced populations** wrt K-Means.
- **Clustering** still seems to partition the users **based**, more or less, **on a social score**: the smaller cluster contains user described by higher features values (thus, **high social score**) and the other one contains users characterized by lower features values (thus, l**ow social score**).
- This trend is not as evident as in K-Means but is still clearly observable.



Hierarchical clustering

Analyze clustering results by changing cut-threshold

- Change coefficient of Scipy standard rule in range [0.4, 0.9]

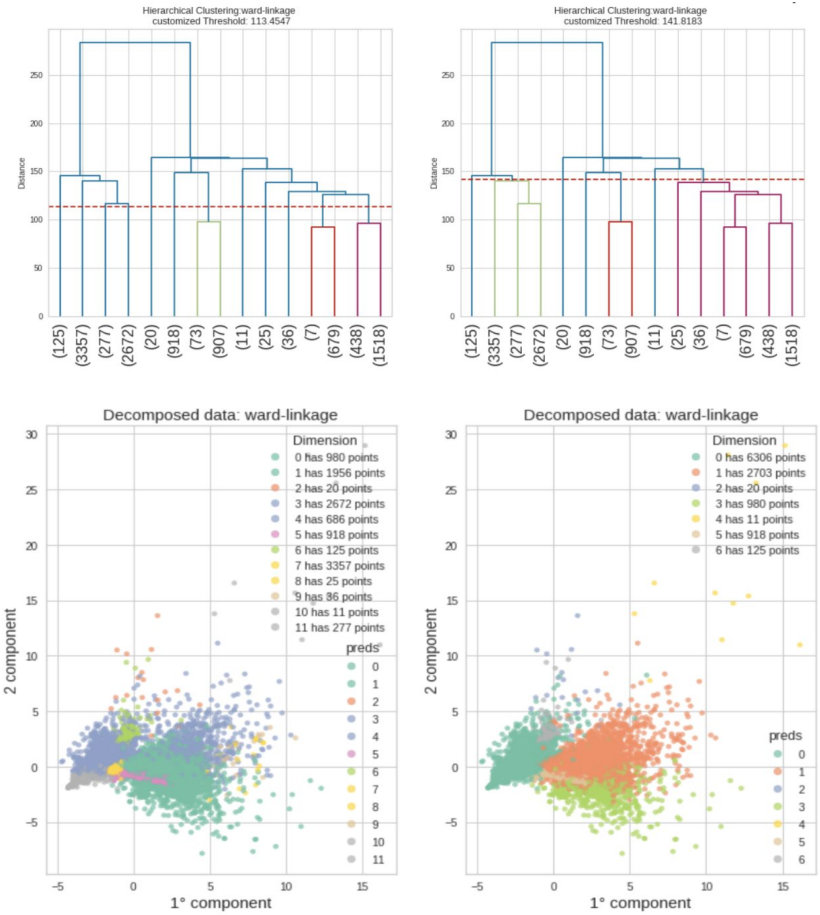
  *Cut_threshdold = coefficient * Linkage Matrix[:, 2]*

- Changing cut-threshold value single, average and complete methods still group in a bigger cluster the greater part of the sample, and only some single observations are grouped in some underpopulated noisy clusters.

- Some interesting results are visible with ward method using 0.5 and 0.4 as coefficient values with cut-threshold values equal to 141.818 and 113.454.

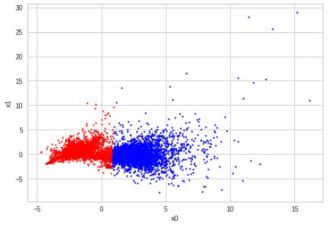
- In both case with obtain some more populated, well-defined and cohesive, while the remaining clusters seems to group together some marginal outliers, thus most of the clusters are significant. They are overlapping each other, thus leading to lower separation degree.

➤ WARD LINKAGE OUTCOMES

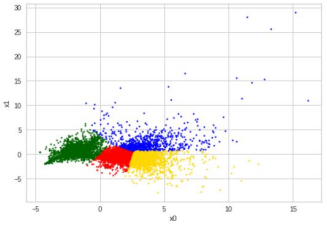


X-Means and SOM

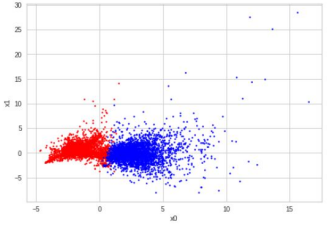
Visual comparison for different K values



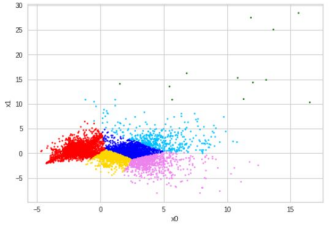
(a) Xmeans with max. $k = 2$.



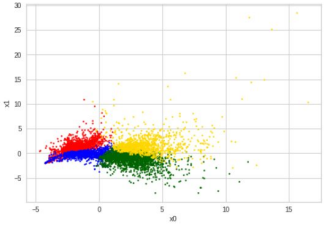
(b) Xmeans with max. $k = 4$.



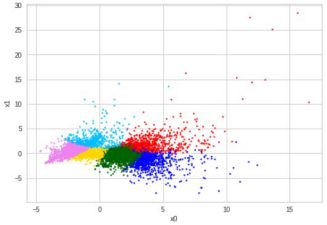
(c) SOM with $k = 2$.



(d) Xmeans with max. $k = 6$.



(e) SOM with $k = 4$.



(f) SOM with $k = 6$.

- **k = 2** he obtained results are very **similar** one to each other and they further **reminds WARD linkage results**; a bit more **overlappin**g in the case of SOMs.
- For **k = 4** and **k = 6** the identified clusters widely differ, suggesting that the **separation principle used is not the same**. More precise and **linear boundaries** are visible for Xmeans. With k = 6 the Xmeans identify a cluster containing only **marginal outliers**.

TASK 2 – Predictive analysis

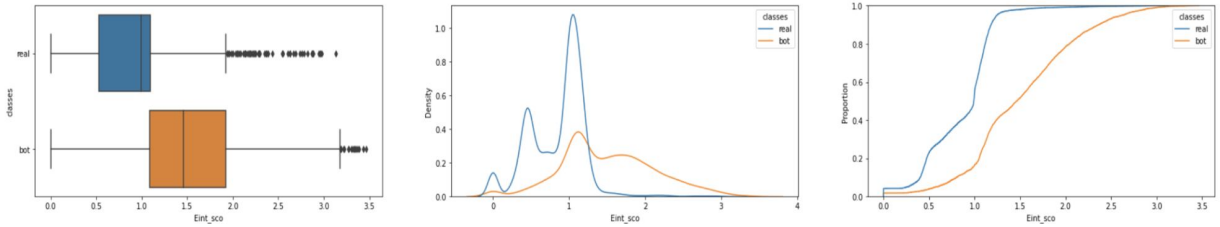
- Define a user profile suitable for this classification task
- Compare different estimators performance

  - KNN
  - GNB
  - DT
  - SVM
  - NN
  - MLP
  - RF
  - AB
  - BAG
  - VC

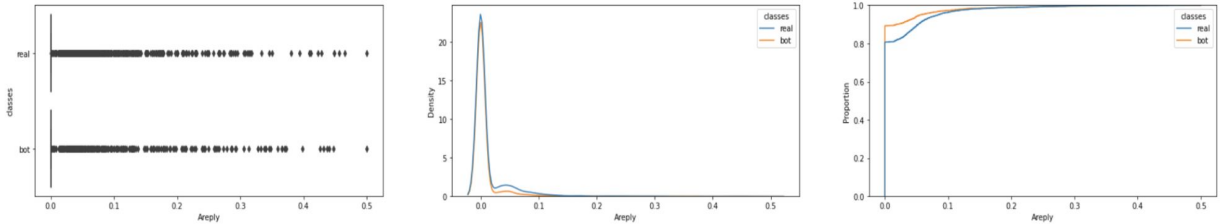
User profiling

Exploratory analysis: get insights about features discriminative capability

Assumption: a feature is more informative the more different are the KDE, ECDF and the boxplots computed for bot and genuine users.



(a) KDE, ECDF and boxplots for `Eint_sco` feature.



(b) KDE, ECDF and boxplots `Areply` feature.

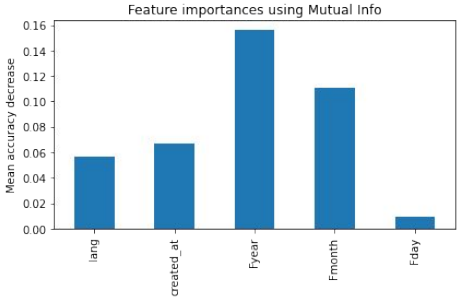
Exploring all other features behaviour, we came up expecting
`Epop_sco`, `Eint_sco`, `Turls`, `Aurls`, `Tments`, `Thash`, `Ahash`, `Aretweet`, `Sreweet`, `Tlike`, `Achar`, `statuses_count`, `Fyear`
to be indicative features.

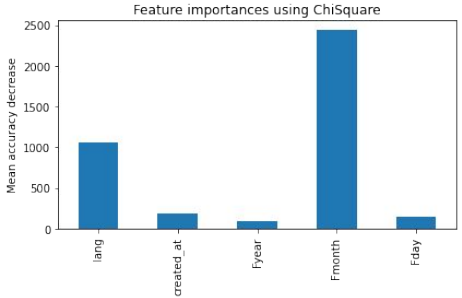
User profiling

Feature selection: Univariate method (CAT) and features importance score (NUM.)

Select the best **categorical features** based on **univariate statistical test** using both **chi-square** and **mutual information** metrics for measuring the dependency between features and target class.

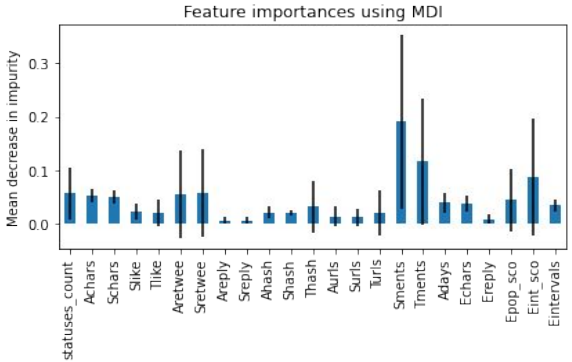
⇒ `Fmonth, Fyear, lang, created_at`

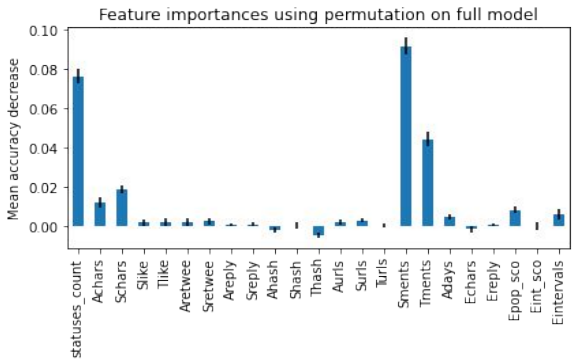




Numerical features importance is computed as the mean of accumulation of the impurity decrease within each tree. **High MDI = high importance.**

⇒ `statuses_count, Achars, Schars, Aretwee, Sretwee, Sments, Tments, Adays, Echars, Epop_sco, Eint_sco, Eintervals.`





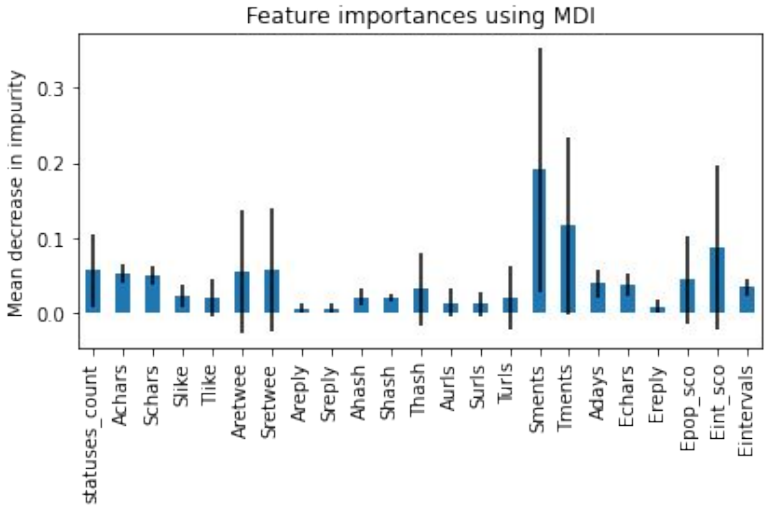
User profiling

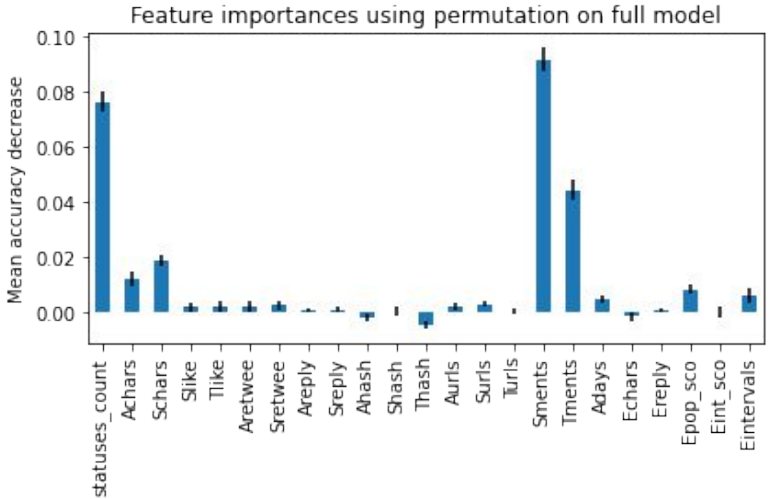
Numeric features selection:
Assess features importance with RF

- Features importance is computed as the mean of accumulation of the impurity decrease within each tree. High MDI = high importance.
  ⇒ bias toward high-cardinality features.

- Permutation feature importance, features are repeatedly shuffled and the model refitted to estimate the importance score.
  ⇒ more costly, more likely to omit a feature.

- 12 most important features: `statuses_count`, `Achars`, `Schars`, `Aretwee`, `Sretwee`, `Sments`, `Tments`, `Adays`, `Echars`, `Epop_sco`, `Eint_sco`, `Eintervals`.





Classification: models and method

Estimators, dataset and procedures

➢ CONSIDERED MODELS

- KNN
- GNB
- DT
- SVM
- NN
- MLP
- RF
- AB
- BAG
- VC

➢ IMPOSE BALANCING CONSTRAINT ON *USER* DATASET

To avoid the different classes' populations from affecting the predictive model performance, a **balancing constraint** between classes was imposed on the dataset**.**

⇒ **Random sub-sampling technique on genuine users**.

➢ VALIDATION SCHEMA

Dataset splitting

- 66% of the dataset was used as Design set in the model selection.
- 33% of the dataset was used as Internal Test set in model assessment.

Model selection was executed through **k-fold cross-validation** technique with **k = 3**. A **Randomized Grid Search** was used to explore the hyper-parameters space; the accuracy score was used for selecting the best configuration.

Model assessment is carried out through Hold-Out technique: each best model was retrained on the whole Design set and its performance are evaluated using the Internal Test set.

Classification: pre-processing impact

Study model behaviour by changing preprocessing technique

Accuracy	Model	SS	MMS	MAS	RS	QT	PT	LT
TR	GNB	**0.806**	0.805	0.805	0.805	**0.806**	0.805	**0.806**
	KNN	**1.000**	0.900	**1.000**	**1.000**	0.849	0.849	0.847
	DT	0.897	**0.917**	0.912	0.900	0.899	0.897	0.900
	SVM	0.830	0.814	0.814	0.824	**0.897**	0.880	0.842
	NN	**0.933**	0.896	0.888	0.925	0.907	0.924	0.901
VL	GNB	0.805	0.804	0.804	0.804	**0.806**	0.805	0.805
	KNN	**0.846**	0.839	0.802	0.791	0.819	0.817	0.817
	DT	0.866	**0.867**	0.865	0.865	0.865	0.865	0.864
	SVM	0.822	0.813	0.813	0.818	**0.881**	0.870	0.838
	NN	0.874	0.863	0.857	0.885	0.883	**0.887**	0.880
Mean TR accuracy		0.893	0.859	0.883	0.891	**0.871**	**0.871**	0.859
Mean VL accuracy		0.842	0.837	0.828	0.833	**0.851**	0.849	0.841

Assumption:
Complex models such as MLP, or ensemble models might benefit the most from the same preprocessing chosen for simpler models.

- Identify the best preprocessing for each classifier
- Identify which preprocessing would best fit most models in order to perform a coherent comparative evaluation of the models performances.

On TR set higher accuracy score is reached with Standard Scaler, while on VL set with Quantile Transformation.

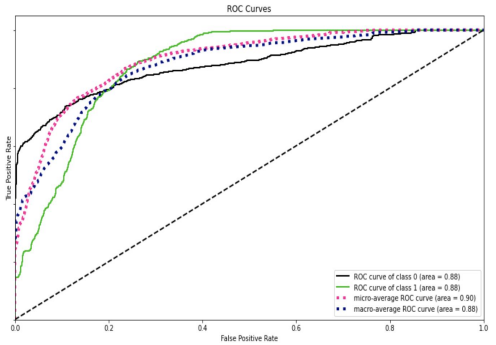
The overall best preprocessing the Quantile Transformation.

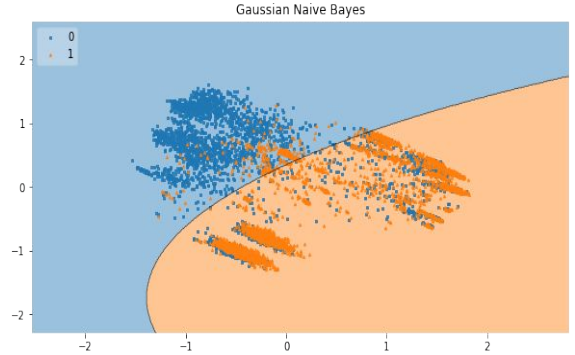
Classification: GNB and KNN

Predictive performance insights: weakness and strengths

➤ GAUSSIAN NAIVE BAYES

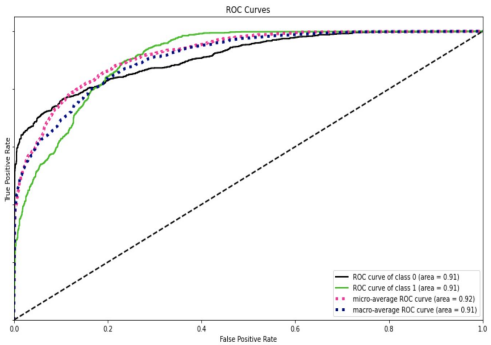
- The decision boundary with an oval/elliptical shape that causes a high number of misclassified data .
- GNB is way too simple for being a suitable classifier on this complex dataset.

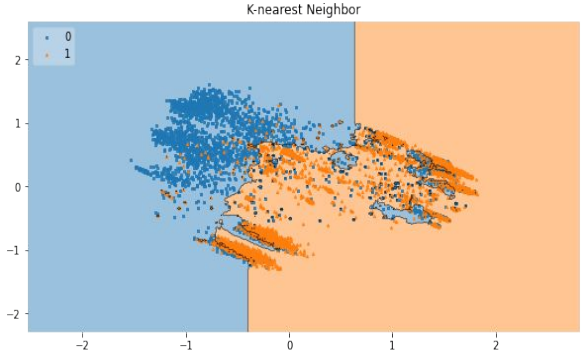




➤ K-NEAREST NEIGHBORS

- Low generalization capabilities on new samples from TS set, leading to a high level of misclassification.
- KNN model suffered the most from the final choice of preprocessing.



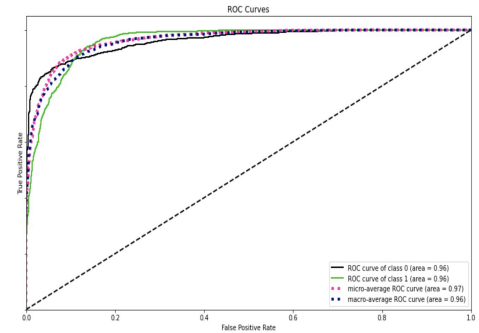


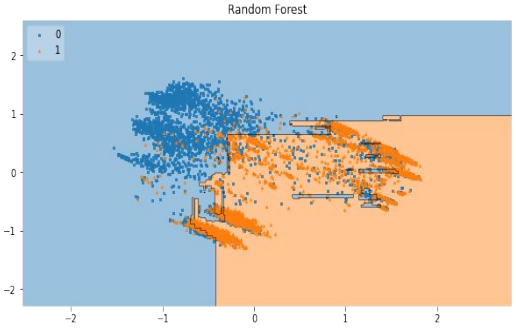
Classification: DT, RF, BAG and AB

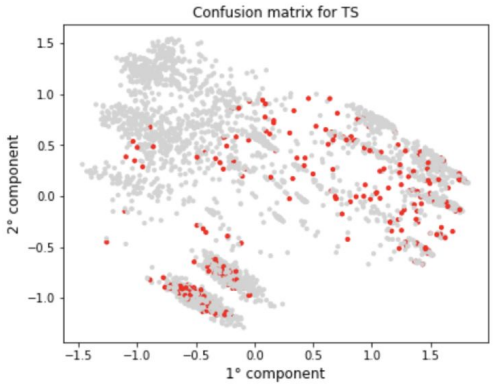
Predictive performance insights: weakness and strengths

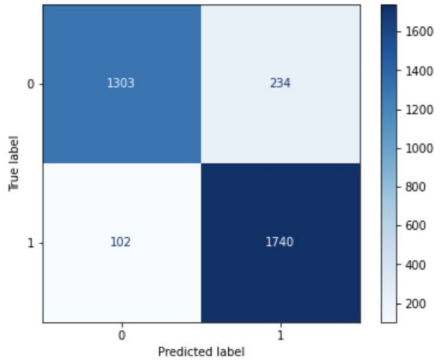
➢ DT-BASED MODELS

- For BAG, RF, and AB estimator ensembles, as well as for DTs, the decision surface exhibits extremely edgy and jagged geometric boundaries.
- the models learn better to discriminate a user between the 2 classes;
- the number of misclassified samples decreases significantly in both Design Set (i.e, in the final re-train) and TS set.
- High explainability.







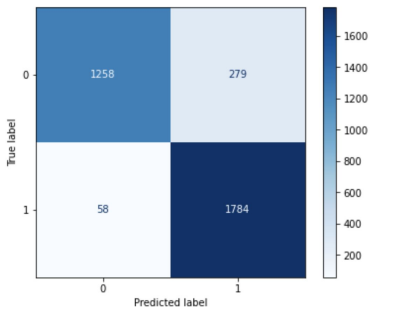


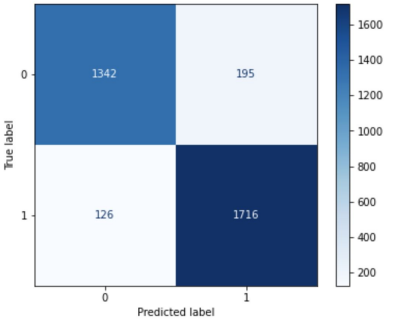
Classification: NN and SVM

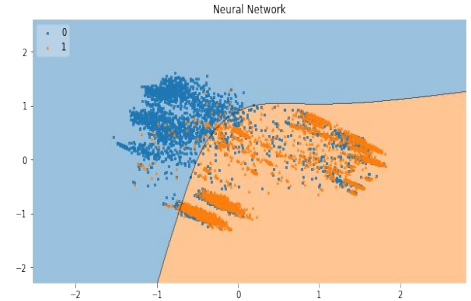
Predictive performance insights: weakness and strengths

- The most evident difference with other models lies in the decision boundary obtained: both SVM and NN we find a smoother and more curvilinear function.
- NN is more accurate in assigning correct the correct target class wrt SVM.
- Both model has high generalization capabilities but lower explainability.

Consideration: all the model tends to make more error in correctly classify bot users.







Classification: VC and MLP

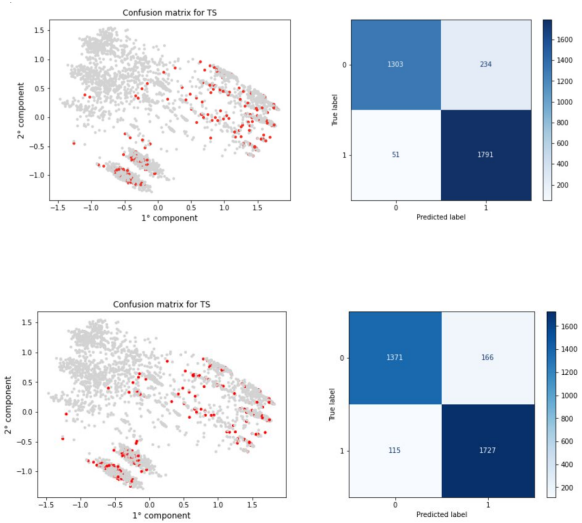
Predictive performance insights: weakness and strengths

➤ VOTING CLASSIFIER

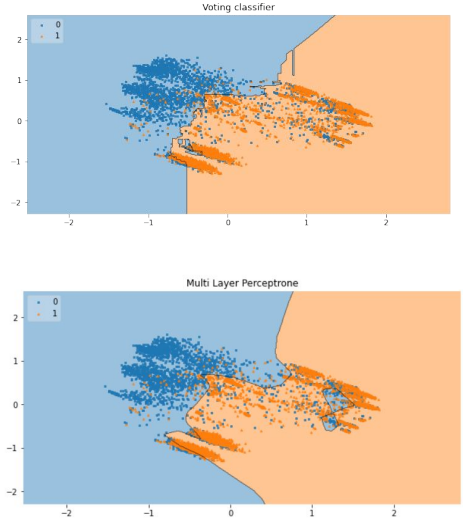
It weep out individual weaknesses of each model. The jagged decision boundary, sometimes angular and sometimes more blunt, allows to achieve some of the good inference performance.

➤ MULTI-LAYER PERCEPTRON

The more accurate decision boundary allow more competitive performance.
The higher number of hidden layers allowed us to implicitly extract patterns from the input data at different levels of hierarchy, and exploit them for an efficient prediction.



(a) Misclassified points: scatter and confusion matrix on TS set.



(b) Decision boundary obtained by training MLP with TR points scatter.

Model ranking

Identify the most suitable estimator based on TS set performance

- The ranking in based on the higher accuracy score achieved on TS set.
- The most suitable model for our classification purpose is the MLP classifier.
- Despite the complexity of this classification task competitive inference performance has been achieved.

➤ **COMMON EVALUATION METRICS**

		Design Set (retrain)			Internal Test Set		
Rank	Model	Acc	Pre	Rec	Acc	Pre	Rec
1°	MLP	0.911	0.908	0.932	0.917	0.9090	0.939
2°	VC	0.914	0.882	0.971	0.915	0.888	0.966
3°	AB	0.914	0.889	0.963	0.914	0.890	0.960
4°	RF	0.899	0.880	0.943	0.902	0.885	0.944
5°	BAG	0.901	0.877	0.950	0.902	0.881	0.949
6°	NN	0.900	0.893	0.928	0.902	0.896	0.928
7°	SVM	0.894	0.859	0.963	0.895	0.864	0.956
8°	DT	0.889	0.871	0.934	0.886	0.868	0.930
9°	KNN	0.829	0.781	0.955	0.830	0.780	0.959
10°	GNB	0.806	0.752	0.962	0.812	0.756	0.965

TASK 4 – Time-series analysis

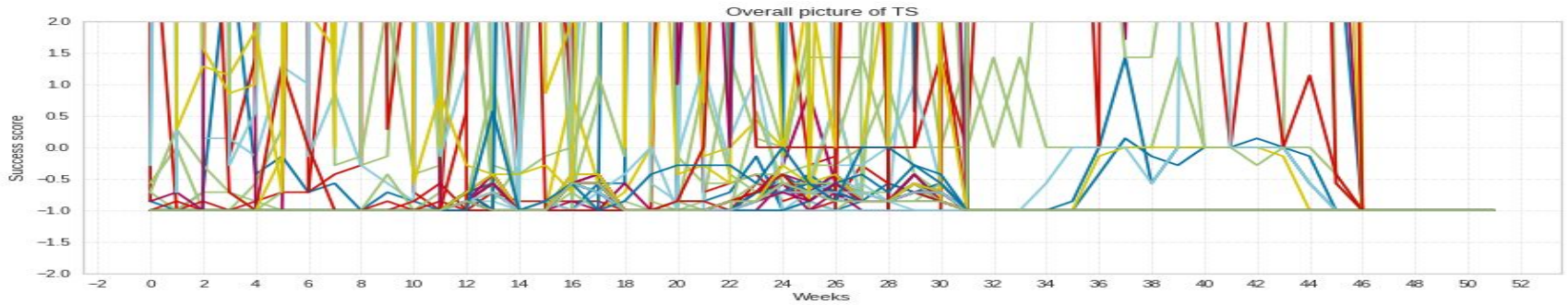
- Time-series extraction and segmentation
- Time-series clustering with K-Means
- Shapelets classification

TS generation

Extraction and segmentation of time-series

- We took in consideration the tweets.csv after cleaning phase.
- We took tweets only published in 2019.
- Generate a Success score for each day.

- We **segmented the all TS in weeks** (for clustering analysis).
- We detected no activity in the last 6 weeks thus we delete them.
- We applied the **MA** with slide windows equal to 4

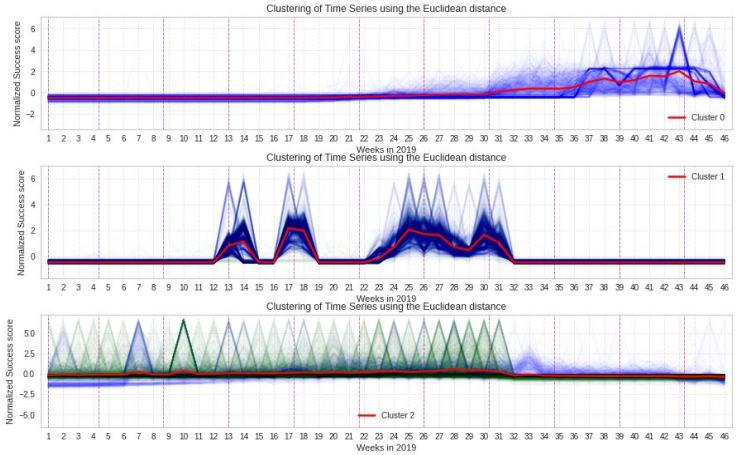


TS clustering

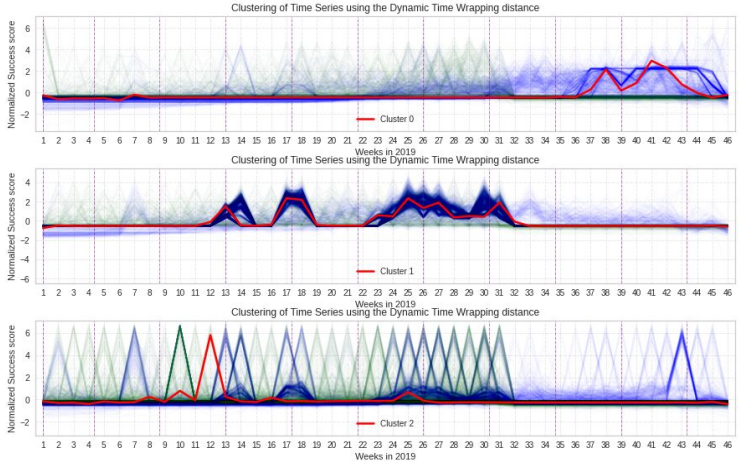
Z-score preprocessing

- In Cluster 0, we have major activity approximately from September to November and low activity in the rest of the year. The most relevant tweets have been published in October. This cluster has captured the users that have published relevant tweets during the autumn-winter seasons.
- In Cluster 1, there is a mix of real and bot users, in this case the cluster covers the spring-summer seasons. We can notice that there is a short period with low or null activity, this means that any of users have published relevant tweets.
- In Cluster 2, we have a bot predominance. We have no correlation with the seasons because there are some "spikes" that lays on all weeks. The major activity is given for the first half of the year. The bot users have the particularity of having the patterns repeating.

EUCLIDEAN DISTANCE - INERTIA: 24.773



DTW DISTANCE - INERTIA: 6.858

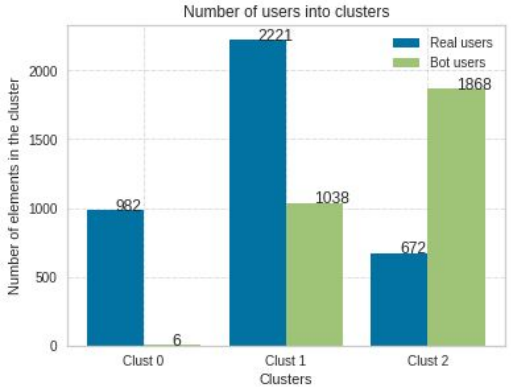


TS clustering

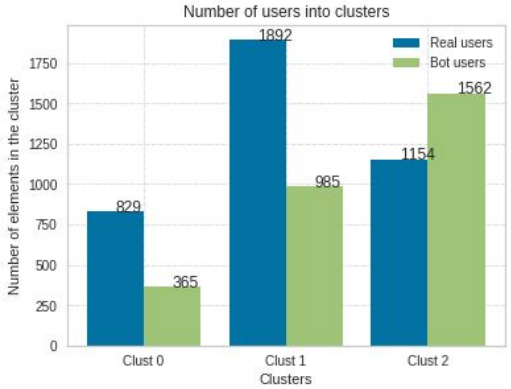
Clustering characterization and further considerations

➢ GENUINE/BOT CHARACTERIZATION and CONCLUSION

- Even if DTW has low inertia, the **Euclidean distance is more suitable** to identify Genuine/Bot TS.
- It has been tried also the same analysis with **MinMax** normalization, but the **results are slightly worse** (in terms of genuine/bot clustering).



Euclidean distance

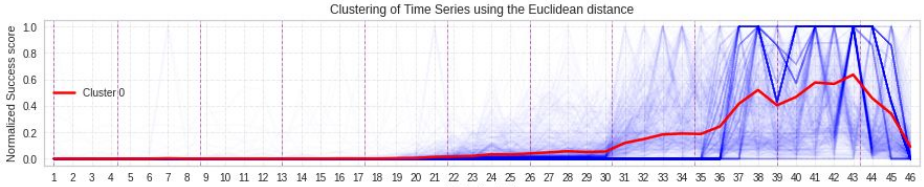


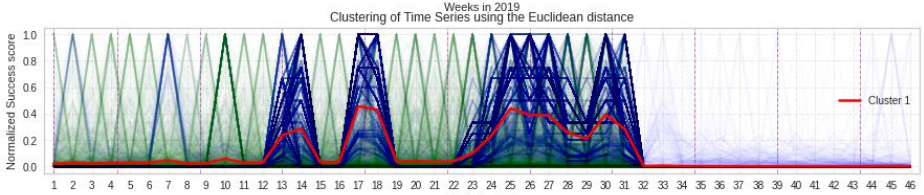
DTW distance

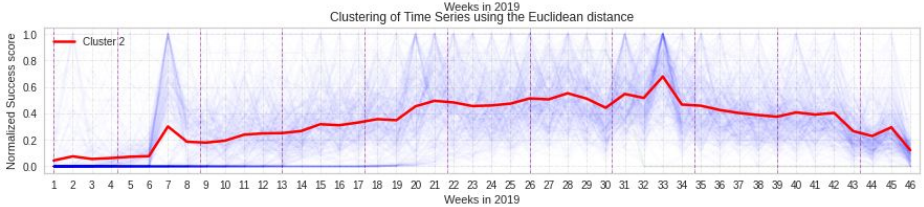
TS clustering

Clustering results with Min-Max preprocessing

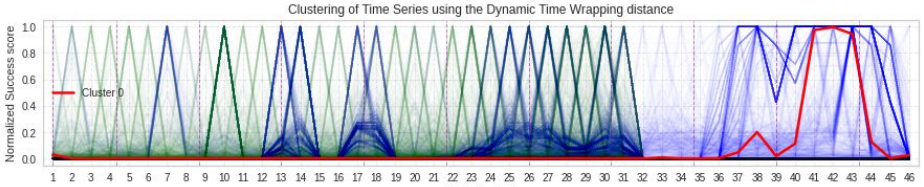
EUCLIDEAN DISTANCE

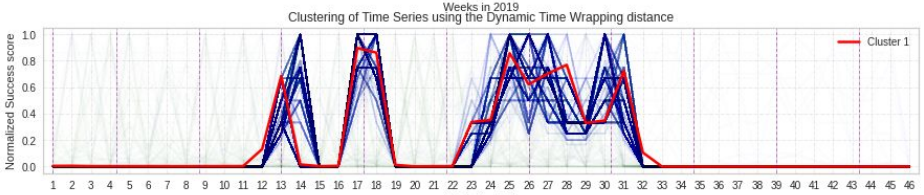


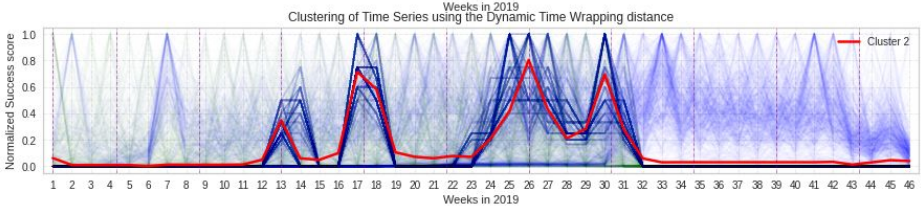




DTW DISTANCE







Shapelet classification

Explore hyperparameter space with grid search

➢ SETTING-UP GRID SEARCH and CORRESPONDING RESULTS

For detecting the optimal configuration:

- 2 kind of algorithms was considered (pyts, tslearn)
- 2 types of normalization technique was analyzed (Z-Score, MinMax)
- 2 different length of TS was generated (weekly TS, full TS)

Top 8 with Z-score

Epochs	Shp. Len	N. Shap.	Score
100	0.2	0.3	***0.6958***
100	0.2	0.2	*0.6936*
100	0.1	0.3	*0.6877*
200	0.1	0.3	*0.6877*
100	0.1	0.2	*0.6671*
200	0.1	0.2	*0.6671*
100	0.3	0.1	*0.6627*
100	0.2	0.1	*0.6575*

Top 8 with MinMax

Epochs	Shp. Len	N. Shap.	Score
100	0.1	0.3	***0.6458***
200	0.1	0.3	*0.6458*
500	0.1	0.3	*0.6458*
100	0.1	0.2	*0.6200*
200	0.1	0.2	*0.6200*
500	0.1	0.2	*0.6200*
100	0.1	0.1	*0.5655*
100	0.2	0.1	*0.5655*

Grid search with Z-score and Minmax utilizing weekly splitted TS and apply a pyts algorithm.

Top 8 with Z-score

Epochs	Shp. Len	N. Shap.	Score
200	0.5	2	***0.7746***
100	0.5	2	*0.7356*
200	0.3	3	*0.7319*
200	0.4	2	*0.7290*
200	0.2	4	*0.7091*
50	0.5	2	*0.7002*
200	0.05	4	*0.6966*
50	0.2	4	*0.6944*

Top 8 with MinMax

Epochs	Shp. Len	N. Shap.	Score
100	0.05	4	***0.6207***
200	0.2	3	*0.6163*
50	0.05	4	*0.6156*
50	0.05	3	*0.6141*
50	0.1	1	*0.6134*
200	0.2	2	*0.6126*
100	0.05	3	*0.6119*
200	0.05	1	*0.6119*

Grid search with Z-score and Minmax utilizing weekly splitted TS and apply a tslearn algorithm.

Top 8 with Z-score

Epochs	Shp. Len	N. Shap.	Score
200	0.5	2	***0.6708***
50	0.5	2	*0.6686*
500	0.5	2	*0.6627*
50	0.05	2	*0.6524*
100	0.5	2	*0.6509*
500	0.2	3	*0.6502*
200	0.05	4	*0.6487*
50	0.1	1	*0.6480*

Top 8 with MinMax

Epochs	Shp. Len	N. Shap.	Score
50	0.5	1	***0.6627***
50	0.4	1	*0.6067*
50	0.5	2	*0.6060*
50	0.05	1	*0.6016*
200	0.05	1	*0.6001*
50	0.05	2	*0.5972*
200	0.05	2	*0.5949*
50	0.1	4	*0.5942*

Grid search with Z-score and Minmax utilizing the full TS and apply a tslearn algorithm.

Shapelet classification

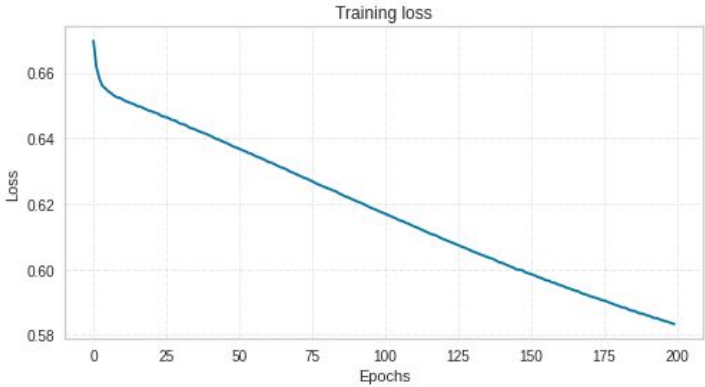
Learned shapelets and considerations

The best score (accuracy) achieved is equal to 0.7746 by full TS with z-score and tslearn algorithm. We can point out basically two things:

- For this kind of task the z-score perform better respect to minmax indeed we can see that the score achieved in z-score are higher respect others normalization.

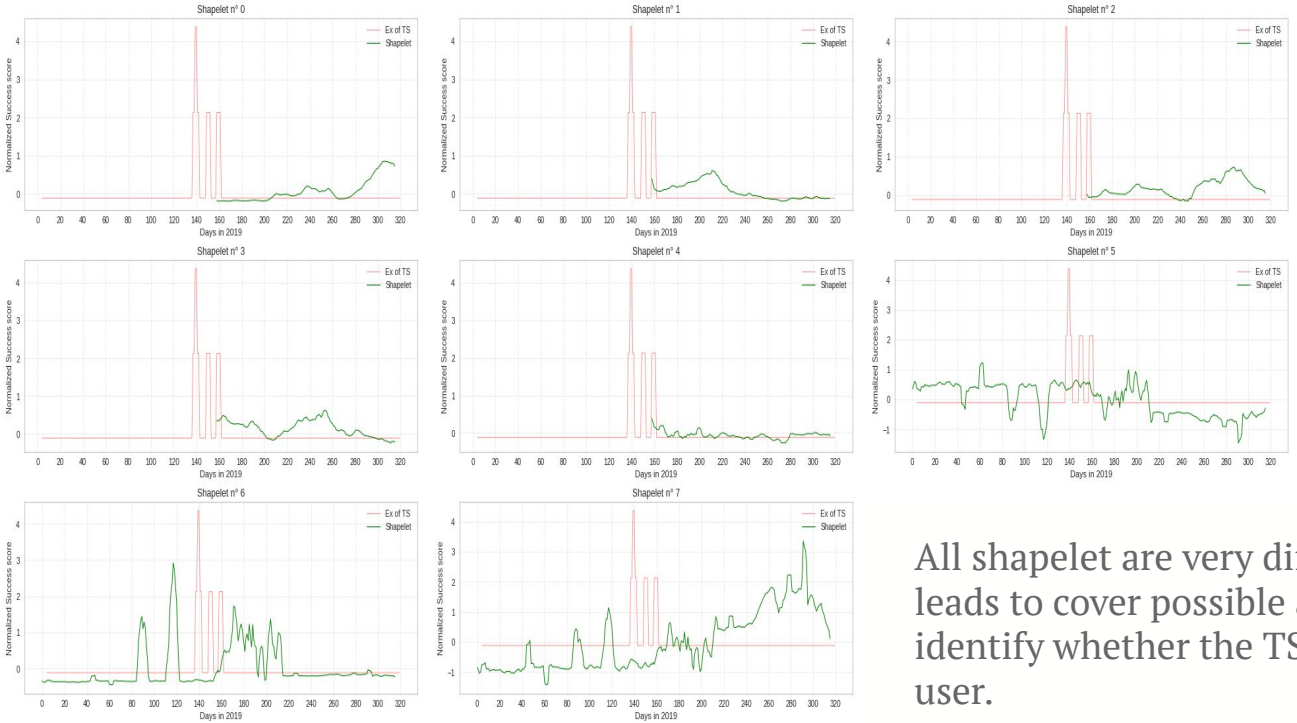
- Best parameters for short and full TS are exactly the same, but a full time series is more suitable for this classification task.

Concerning the values of parameters we can notice that the higher scores are associated to mainly to a few and long shapelts with the low number of epochs.



Shapelet classification

Shapelets visualization and conclusions



- The algorithm found in total 8 shapelets, the first 5 have a length equal to 158 days, the last 3 have a length equal to 316 days and cover the entire length of all TS.
- The longest shapelets seems to be more unstable respect to the smaller one.

All shapelet are very different from each other and this leads to cover possible aspect/characteristic that identify whether the TS is associated to a bot or real user.

Thank you for the attention!