Università di Pisa

Project report - Data Mining I

Glasgow Norms data-set analysis

Alessio Cascione, Jordanos Feyissa Gemechu

Contents

1 Data preparation and data understanding

1.1 Data semantics, distribution of the variables and statistics

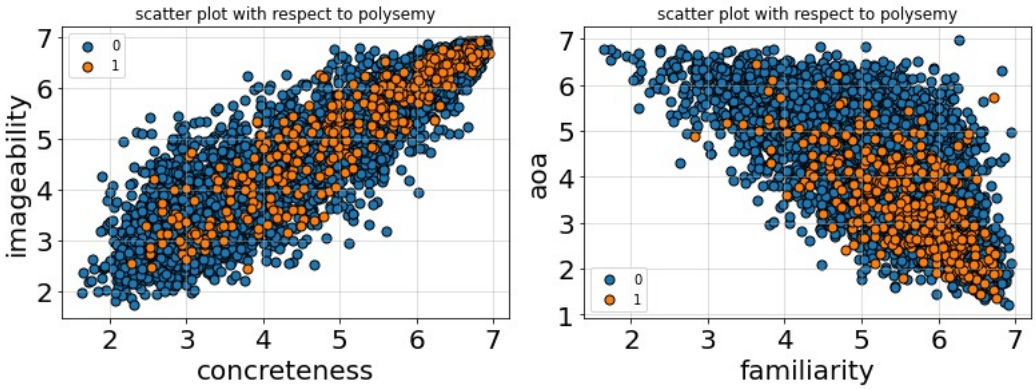
In this section a very brief explanation of the variables' meaning and their main characteristics will be illustrated. In the data-set, for each word, a length value along with a frequency value is specified: the length of the word indicates the number of letters that comprise it, while the frequency expresses the number of occurrences of that word with respect to the *Google Newspapers Corpus.* The other main psycholinguistics variables will now be listed. First of all, we have three dimensions used to characterize a word's emotional impact: "arousal" is used to measure excitement or calmness, "valence" for measuring the positive or negative value of the semantic content associated with the word and then "dominance" is used to express the degree of control with respect to that word. Their values can vary in the continuous interval of $[1-9]$. On the other hand, other psycholinguistics features used in the data-set can acquire a value in the interval $[1-7]$: these are "concreteness", that is used to measure the degree to which something can be experienced with our senses, "imageability", used to measure the effort to imagine the semantic content of a word, "familiarity", that expresses, with respect to the subjective experience of the participant in the linguistic experiment, how familiar a particular word is, "age of acquisition", that gives an estimation of when the participant learned a certain word (a 7-point scale is used, where a series of 2 year periods from 0-12 is specified along with a final 13+ period), "semantic size", that identifies the magnitude of the word, both in abstract or concrete sense, and "gender", that gives a degree of masculinity or femininity to a word[1].

The last interesting dimension is called "polysemy": it is a categorical variable used to express if a word has multiple meanings (1), or is unambiguous (0). In order to start understanding how the variables distribute across the target variable "polysemy", a first table presenting the value of the mean of the main continuous variable we will be focusing on with respect to the target categorical variable is introduced below:

	non-polysemous words	polysemous words
familiarity	5.242439	5.599404
age of acquisition	4.209967	3.387963
imageability	4.680850	5.201778
concreteness	4.517969	5.114697

Some preliminary interesting comments may be done: first of all it is possible to notice how the mean value of "imageability" and "concreteness" seems to be quite similar. Polysemous words recorded in the data-set tend to have the same degree of "concreteness" and "imageability", and the same can be said regarding non-polysemous ones. On the other hand, slightly different values are present when focusing on "age of acquisition" and "familiarity". In particular, non polysemous words seem to be acquired a bit later in time then polysemous one, even though the difference in age is not extremely large. Finally, polysemous and unambiguous words seem to share very similar value of "familiarity". Two scatter plots are introduced here in order to show the relationship between the variables that have been chosen for the analysis and the target variable:

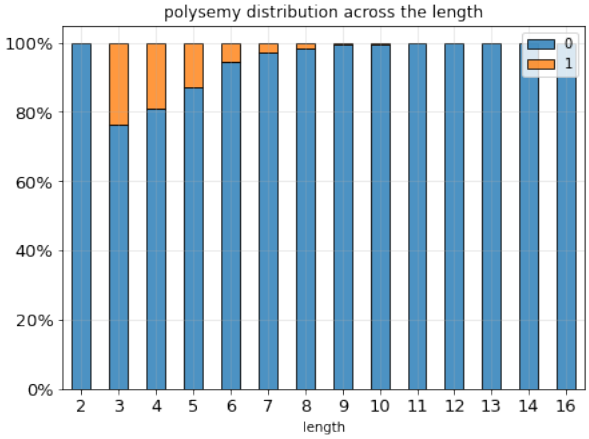
[1] Deeper explanations of the role of variables is available in the main related paper for the Glasgow Norms data-set [1].



As it can be seen in the first plot, polysemous words tend to be encapsulated inside the bigger group of non-polysemous ones when compared with respect to "concreteness" and "imageability". The relationship between the two variables is positive: as "concreteness" increases, so does "imageability". This is consistent with a first intuition that the more concrete the semantic content of a word is, the easier it is to imagine something associated with the meaning of that word. The representation suggests a particularly interesting correlation between the two variables in question. In particular, the Pearson's correlation coefficient between the two variables is about 0.90, which is the highest entry in the correlation . About the same value is obtained using Spearman's coefficient. This seems coherent when the result is compared with the graphic representation, showing almost a linear relationship between the variables.

The second plot allows us to see an interesting distribution, too. It compares the unambiguous and polysemous words with respect to "familiarity" and "age of acquisition". It is interesting to notice that at high degrees of "familiarity", "age of acquisition" value tends to be low. Inversely, low degree of "familiarity" are associated with a later "age of acquisition". This is quite consistent with our intuitions: it is reasonable to suppose that more familiar words will correspond to words learned in the first years of life. This conclusion is well supported by the correlation coefficients measured between the variables: Pearson's correlation coefficient and Spearman's coefficient are both about -0.67, entailing the relationship described before and being consistent with the negative shape of the distribution.

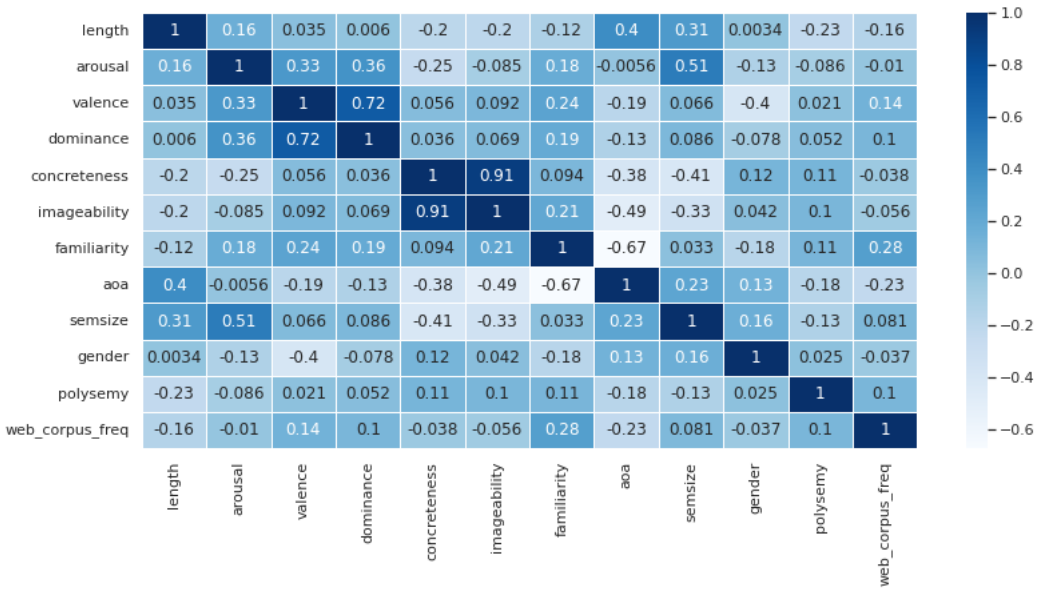
After having performed an analysis on the relationship between psycholinguistic variables and polysemy, we introduce a brief overview regarding the relationship between the length of a word and its polysemous feature. The bar-plot below illustrates how polysemous words distributes across the length:



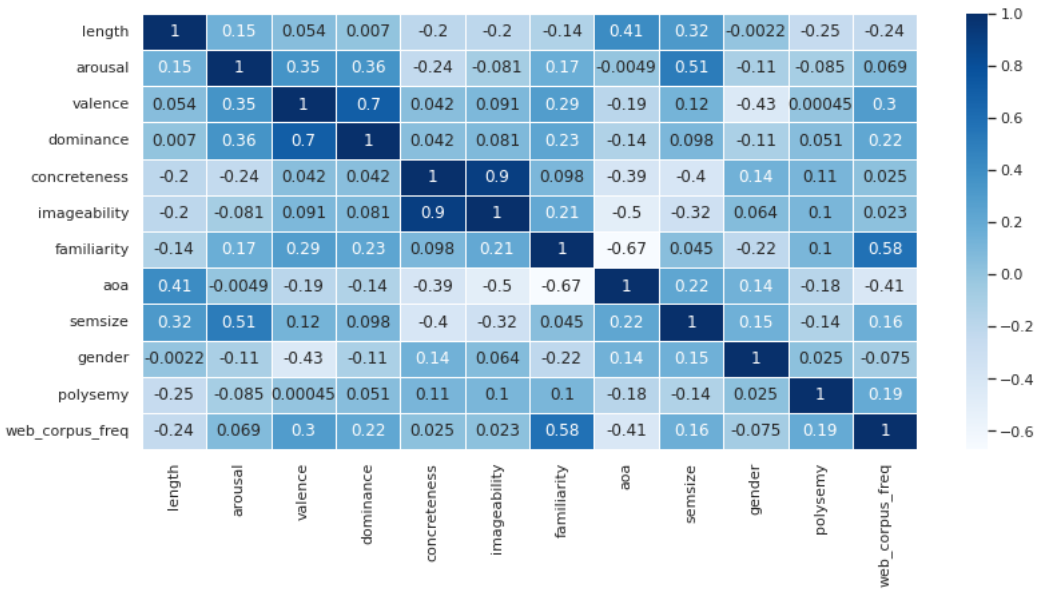
What appears evident is that most polysemous words fall in the range between 3 and 7 when compared with respect to their length. There are just few words of this kind having 8 or 9 letters, and just one polysemous word having 10 letters, namely the word "periodical". This seems to be a reasonable result: it makes sense that polysemous words, used in various contexts in order to convey different information, might tend to be shorter for reasons related to the "economicity" of natural languages: if the speaker of a certain language is going to use the same word multiple times in different occasions, it would be "cheaper" for her/him if the word in question could be expressed with less letters.

1.2 Pairwise correlation, data quality assesment and variable transformation

We start here a short analysis regarding the correlation coefficient introduced in the tables below. The first table introduces the value computed using Pearson's correlation coefficient, the second using Spearman's coefficient:



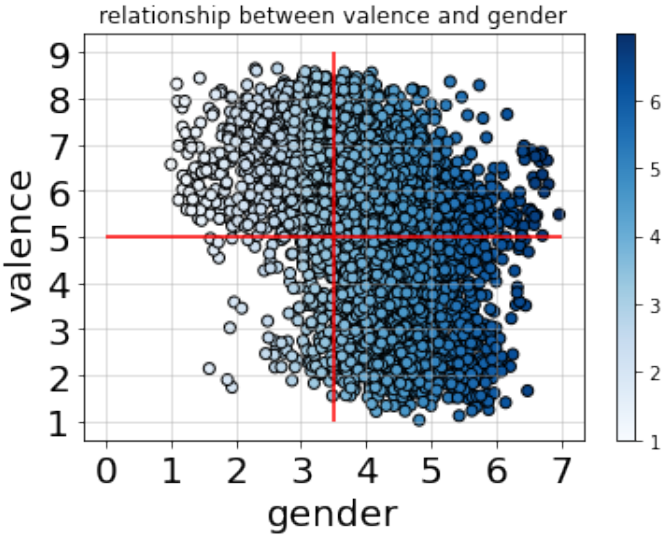
as it is clear at a first glance, for every couple of variable, coefficients computed with the two different methods do not vary significantly. As emphasized above in the scatter plot analysis, the couple made of "concreteness" and "imageability" is the most correlated, while "age of acquisition" and "familiarity" are negatively correlated in a strong way. Other interesting values arise with respect to "semantic size" and "arousal", which share a coefficient of about 0.51, suggesting that the bigger the object or the concept the word refers to, the stronger the emotion elicited by its meaning. "valence" and "dominance" also share a rather high correlation coefficient of 0.7: given that, it could be possible to speculate that words towards which a subject feels more control over are associated with a high positive value. Another interesting relationship is the one that connects the length of a word with "age of acquisition": the correlation coefficient is 0.41. This is a somehow intuitive correlation: we might expect that longer words are mainly acquired later in life. One further relevant relation is a negative one, between "gender" and "valence" (-0.43): from the distribution, it seems possible to conclude that words judged to be more feminine are tendentially considered more positive, too.



The anti-correlation is interesting enough to deserve a brief further analysis: a small table is introduced below, comparing some significant examples of words with respect to "gender" and "valence":

	gender	valence
lady	1.0	6.571
sentiment	2.469	6.333
womb	1.235	6.235
rapist	5.886	1.257
forceful	5.788	2.886
slaughter	5.849	1.438

These cases are not isolated examples of an anti-correlation between these two dimensions. Looking at the graph below, one can observe the global phenomenon with respect to every point in the data-set. The scatter plot uses various gradients of blue to emphasize the high or low degree assigned for a particular word to the value of the variable "gender". Darker shades of blue are used for more masculine words, while lighter shades identify more feminine ones. The division in quadrants makes the interpretation of the result clearer: the lower-left quadrant shows significantly less points then the upper-left one, suggesting a high number of feminine words having higher degrees of valence. The upper-right quadrant is slightly less populated then the lower-right one, suggesting that most of the masculine words tend to present a small degree of valence. This confirms the result entailed by the correlation coefficient.



Moving on to the assesment of data quality, we must mention the presence of a total of 14 missing values for the frequency variable: since the next task, involving the use of clustering algorithms, is better managed when working with complete data and since the missing value are relatively few, it might be better to devise a method in order to fill the empty entries of the variable. Although the distribution of the frequency of the variable, once normalized using the $\log_{10}$ function, seems to present a form that resembles a normal distribution, it might not be completely safe to just fill the empty values using the mean value of the dimension in question. This mainly depends on a fundamental empirical result established in computational linguistics, namely Zipf's law: given a text, there will be very few words having a particularly high frequency while most of the words will occur just a few times, having a very low frequency or having just one occurrence in the whole text. This implies, for most texts, a non-normal distribution of the frequency of words that does not allow us to safely assign the mean value to a missing entry of the frequency attribute: the word whose value must be filled could easily fall in the tail of the distribution, and it would be unrealistic that its frequency value is near the mean.

Another direction must then be taken: looking directly at the correlation between the frequency and other variables could be a promising solution. The correlation tables show a continuous variable that, with respect to every other variable, is better correlated with the frequency: familiarity and frequency share a Pearson's correlation coefficient of 0.28 and share a Spearman's correlation coefficient of about 0.58. The other correlation values are not as significant as the one introduced above.

But there is a parallel intuitive solution worth to be mentioned, related to background psycholinguistics knowledge. It revolves around using the length variable as a support to assign the missing frequency value: since in a text the most frequent words tend to be the shortest ones, and longer words tend to be less frequent, it might sound reasonable to assign to a word having a missing frequency value the mean of the value with respect to the length of the other words, such that, for instance, the word "Christmas" will get as mean value the mean frequency of the words having 9 letters, and so on. On the other hand, and quite surprisingly, the data-set presents a counter-intuitive situation: the coefficient of similarity between the length and the frequency is quite low (-0.24 when measured with Spearman's method, -0.15 with Pearson's). In order to clarify the decision of using the "familiarity" variable as a support to fill the missing values, it might be proper to consider some cases in which, for the same word, we consider the different possible outcomes with respect to the two filling methods advanced above[2].

[2]We specify here that the filling of the frequency value for the word "Facebook" has been performed using the mean frequency value for words of familiarity 6.833 rather than 6.829, the original familiarity value for "Facebook". One main reason justified this choice: the word

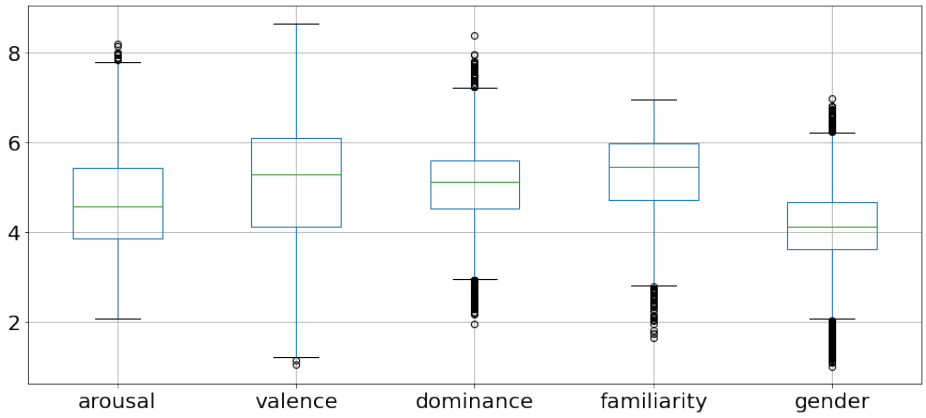
	Filled according to length	Filled according to familiarity
skijump	22125692	4740739
burgle	22855107	69763449
Mum	84529424	103260853
Facebook	19496258	264720374
Christmas	12911065	27318959
TV	829969374	29237400

Frequency values filled according to familiarity seem to be more consistent with our intuitions then the values obtained using the length as a support variable: it is quite reasonable that "Mum" should have high frequency values, being one of the first word used and learned. The filling according to familiarity assigns to "skijump" a value that is one order of magnitude lower then "TV" and two orders of magnitude lower then "Mum", too. This seems intuitively right: "skijump" should be much less frequent then those two words. The frequency values for "Facebook" and "Christmas" also seem more reasonable under the familiarity filling. These results are enough to justify the choice of this dimension in order to solve the issue related to the presence of 14 missing values.

In addition to this, we have performed a check on the "length" variable, with respect to the actual number of letters that each word in the data-set actually has. No error of sort emerged after the procedure. Furthermore, we have checked that each value fell exactly in the range it is meant to find itself in, for each single word in the data-set: no semantic inconsistencies arise from this perspective.

In order to conclude the data understanding section, it is relevant to mention that the main tasks that will be studied in the next chapters have been performed excluding one of the variables: the high positive correlation between “imageability” and “concreteness” allows us to remove the latter variable and perform further analysis using just the former. Then, in order to facilitate evaluation over the frequency of words, words’ values with respect to the dimension in question have been normalized using the $\log_{10}$ function. At last, two box-plots showing possible outliers for the distribution are presented below. After a first analysis, we noticed that both "familiarity" and "arousal" present a rather heavy tailed distribution, suggesting that the values identified through the box-plots as possible outlier should be preserved rather then deleted for further analysis. The two dimensions "gender" and "dominance" presents a high amount of points falling outside from the limit fixed by the 25th-percentile and the 75-percentile, both multiplied for a factor k of 1.5: deciding to eliminate these points could lead to a loss of relevant information for the next analysis. Finally, even though the distribution for "valence" does not seem to fall in these two kind of scenarios, we also decide to keep the outliers in this dimension, for reasons related to further possible analysis.

"Facebook" does not share its familiarity value of 6.829 with any other word and its frequency value is absent. Therefore, we can’t assign to that word the mean frequency value of the words of familiarity 6.829, since the mean value can’t be computed, as there are no other words we can consider to get an actual value for the mean. Choosing the closest familiarity value is one way to solve the specific issue, without changing the other values as well. Also, the values in the tables have been rounded where needed.



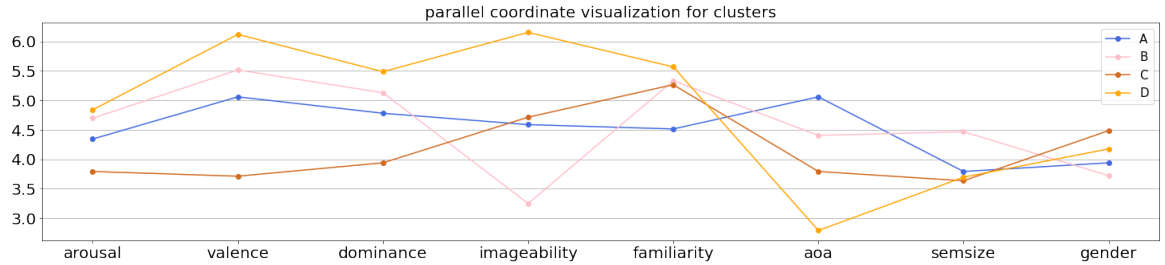
2 Clustering analysis

2.1 K-Medoids analysis

The first clustering approach we are going to discuss is K-Medoids. The reason behind the choice of K-Medoids algorithm rather then K-means is related to the identification of centroids for our analysis: although both approaches can be considered centroid-based clustering methods[3] K-Medoids chooses the centroids for the analysis among real points of the starting data-set: these points correspond to the "least dissimilar" elements from all the other data points in the cluster. For each cluster, they are the point x such that the sum of the distances between x and any other point y included in the cluster is minimal. This is a slightly different approach from the K-Means: with K-Medoids, we are sure to get a real point as centroid of our cluster, whereas with K-Means centroids do not necessarily correspond to real elements of the original set. By choosing to use this kind of approach, we are able to gather better information regarding the words selected as medoids during the clustering process. For our analysis, we will be considering how the clustering algorithm performs on the main continuous psycholinguistic variables in the data-set, excluding the categorical ones. We start by considering how the sum of squared errors (SSE) changes according to different choices of k in order to find the optimal number of clusters for the data-set: given a partitioning in clusters performed by K-Medoids, lower values of SSE imply an overall better divisions in groups. One of the heuristic suggested when using a centroid based clustering algorithm is that of finding certain values for the number of clusters such that, for higher values of k, the decrease in terms of SSE is not so relevant. In this way we identify those values of k that might consistitute a good trade-off between number of clusters and SSE. The result for the data-set showed as a good range the values between 4 and 14, and we decided to use the lowest suggested value. The reason behind the choice of a $k = 4$ is the following: after having considered the measures of k falling in the range designated as reasonable according to the heuristic, clusters soon started to become more and more closer to each other for every continuous variable taken into account, having very similar medoids, too. Hence the preference for proceeding the analysis with a lower k value. We will show how, in this way, K-Medoids allows us to have at least some better differentiated clusters with respect to the main psycholinguistic dimensions in the data-set. We also specify here that the number of iterations considered before fixing the best medoids configuration has been of 100.

[3] Both methods identify a number k of starting centroids and accordingly build clusters around them by doing comparisons between centroids and their respective closest points. Then, the algorithms recompute the centroids of each individual clusters and compute again the distance between the other points, up until a satisfactory partitioning is reached.

We start with a visualization of the parallel coordinate plot. This allows us to have a broad overview regarding medoids' values with respect to every continuous psycholinguistic dimension.



In particular, with a $k = 4$, stricter separations between identified clusters arise with respect to "valence" and "age of acquisition". The variable "imageability" presents an interesting phenomenon, too: clusters D and B stand far apart from each other, while A and C share medoids having a rather similar value according to that variable. Keeping in mind the result illustrated in the parallel coordinate graph, we may proceed to identify which words have been chosen by the algorithm as medoids of the identified clusters. These four points are introduced in the table below. We decided to provide the actual values only for dimensions that appear of a certain interest considering the previous plot.

	valence	dominance	imageability	age of acquisition
A - cluster	5.059	4.781	4.588	5.059
B - theme	5.515	5.129	3.25	4.406
C - drag	3.714	3.941	4.714	3.794
D - swimmer	6.118	5.485	6.152	2.794

Some comments regarding the medoids under analysis can be made: as it is visible in the parallel coordinates plots, the difference in terms of "imageability" between the D-cluster and the B-cluster is consistent with the value that their respective medoids acquire, as "swimmer" is much easier to imagine then "theme", considering the abstractness of the latter. On the other hand, "drag" and "cluster" do not reasonably differ so much in terms of imageability, hence the similarity of the respective clusters in terms of that variable. On the other hand, the difference in terms of "age of acquisition" between "swimmer" and "cluster" is coherent with the distance between the two medoids under that dimension. An interesting qualitative observation may be done regarding the D-cluster and the variable "valence": the medoid of the former cluster stands quite apart from the others, that slightly tend to be similar. It might be of interest to check how representative these four medoids are with respect to each cluster: therefore, we have computed for each cluster the mean value of each dimension, obtaining the mean point of the aggregates. Then, we used euclidean distance to compare the mean point of each clusters with the actual medoids. The following distances were obtained:

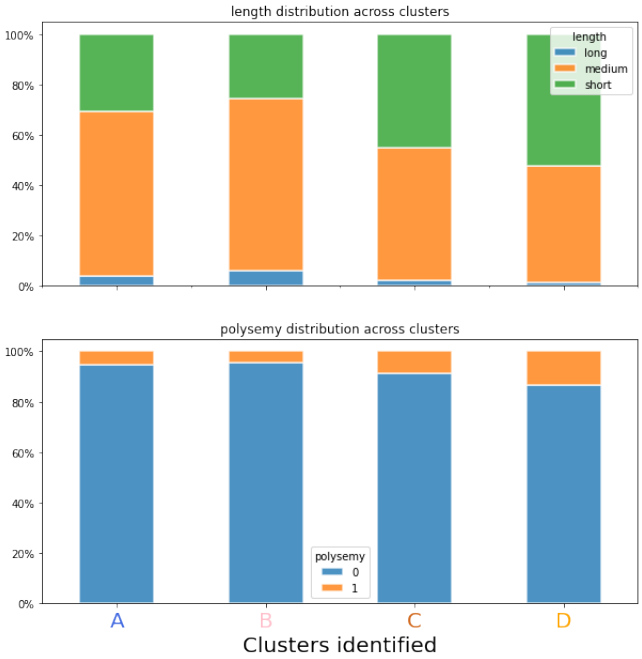
(A) 3.8731728529640224

(B) 1.884520504086131

(C) 3.4513473066808737

(D) 5.313633832698843

Interestingly, we notice that "swimmer" is the most distant point from the mean in comparison with the other clusters, suggesting that it is only partially representative of D[4]. At this point, it might be also interesting to show how categorical variables not used in the core clustering algorithm distributes across the identified groups. The bar-charts below aims to show exactly this relations, emphasizing how polysemous words and words with various length tend to distribute for each cluster[5].

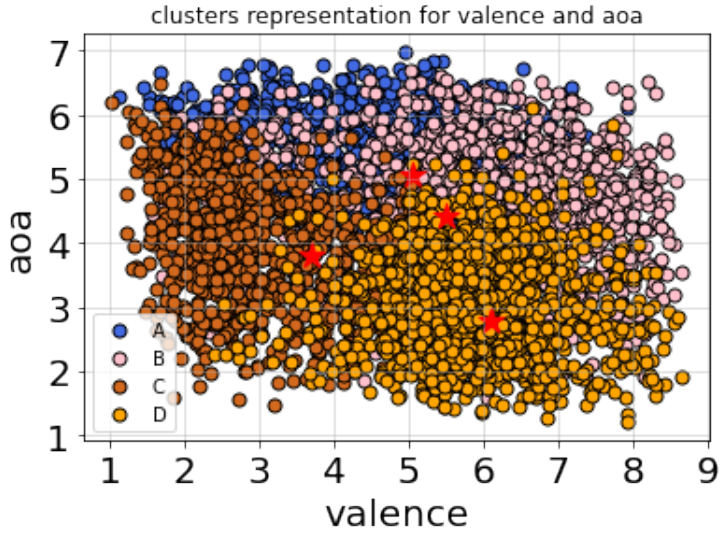


As it appears evident from the bars, it is the D-cluster that contains both the largest number of short words and polysemous ones. An explanation for this result could probably depend on the size the cluster has: it encapsulates the highest number of points (1431), followed by B which stores 1314 elements and A, with 1063 words. The C-cluster is the smallest one, counting 874 words. At the same time, it is interesting to notice that the two clusters which store the highest number of shorter words tend to present a higher percentage of polysemous words, making the intuition that there is a deep relation between the length of a word and its being a polysemous one stronger.

The next scatter plot gives a graphical representation of the clusters with respect to the dimensions for which we can obtain better separated groups.

[4]For the sake of completeness, we add here the mean of the standard deviations with respect to the main continuous variables for each cluster. A) 0.8802333293241871; B) 0.9202062471982387; C) 0.888427653727963; D) 0.8605254625693849. The four aggregates share more or less a very similar value in this respect and the less representative value of D's medoid is worth to be mentioned.

[5]The elements have been grouped according to their length: words having 5 letters or less have been grouped as "short" ones, those having a length value between 6 and 10 included have been labeled as "medium" and all words longer then 10letters as "long".



Red markers on the graph are used to identify medoids. As the parallel coordinate graph was displaying before, *C* and *D* are quite well separated clusters from the "valence" perspective, while *A* and *B* seem to form aggregates which are not so distinct. For "age of acquisition" a somehow analogous discourse can be made, though we notice that also in this case clusters tend to slightly merge with each other when we consider the points at the border of the groups.

2.2 DBSCAN clustering analysis

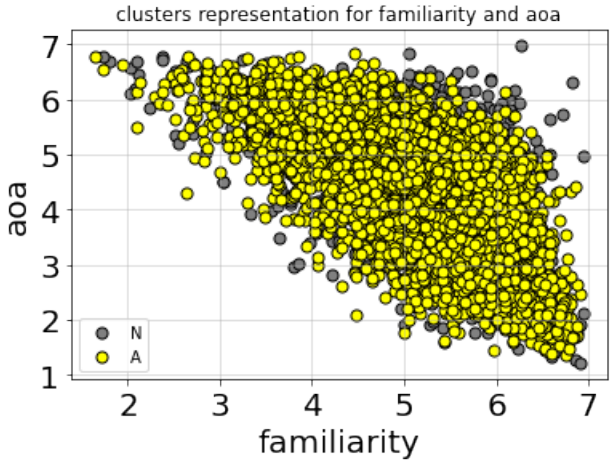
We proceed now with a discussion concerning the application of a density based clustering method, the DBSCAN algorithm, over the Glasgow Norms data-set. We will use ε to denote the radius associated with each point and we will call the minimum number of points required to be inside the radius in order to label a certain element a core point *MinPts*. In order to identify the best value of ε for the algorithm in question, we consider for every point in the data-set its distance with its *k*th-closest neighbor (*k*-dist), then sort these points from the one that has the lowest distance from the *k*th-closest neighbor to the one presenting the highest distance. If the *k* value is equal to the *MinPts* value and we select the ε value according to the distance, then points for which *k*-dist is less than ε will be labeled as core points[6]. This gives us a heuristic to pick the optimal value for ε: ideally, it should be among the *k*-dist values of the points in the sorting such that the points after them present an abrupt increase in terms of *k*-dist . In this way, we are reasonably sure of having considered a value of ε such that many reasonable core points are labeled as such while points that could become core points only at the price of excessively extending the radius are considered border or noise elements.

We checked how the distance changed according to different values of *k* in a range of 1 to 20 in order to identify the best radius length. The different plots appear quite similar, and the best value tend to be concentrated in the range between 0.25 and 0.30. We will consider the former value for further analysis using DBSCAN algorithm.

After having checked the scatter matrix for values of *MinPts* ranging from 6 to 16 and a fixed value of ε, it was possible to notice how in most of the cases, the result obtained consisted of a big individual cluster and a small percentage of the original points considered as noise. We present here a concrete example with respect to "familiarity" and "age of acquisition", for a *MinPts* of 16, but emphasize that very similar clustering results verify

[6]For instance, supposing *MinPts* =*k*=4 and we fix ε to 0.20, then every point that has a 4-dist less then 0.20 will be considered a core points, while the other will end up being considered as border or noise points.

with different *MinPts* values for the same ε:



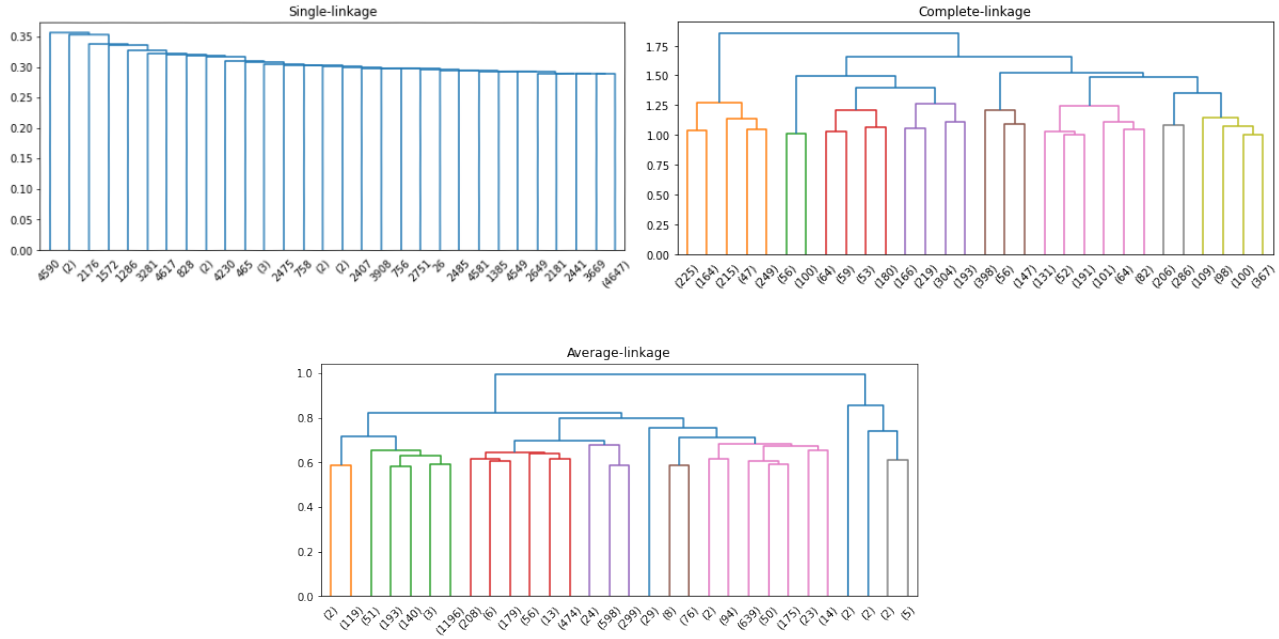
As was anticipated just before, most of the points fall in just a single cluster, and some of the elements on the borders end up being recognized as noise. At least, the scatter plot provides useful information to establish why DBSCAN's performance on Glasgow Norms does not appear very efficient: qualitative observation shows that points tend to form a compact shape. Almost every element has a high number of neighbors, having short euclidean distance from it. By using a scatter matrix, the reasoning could be generalized, observing the distance that points share with respect to every possible couple of continuous variables. Histograms of the continuous variables also give us some interesting information: most of the distribution of the variables have long continuous tails, where the points tend to gradually increase or decrease in value, without abrupt changes. We might try to generalize the result of the scatter plot in virtue of these observations, claiming that probably each point tends to have, for each dimension, many neighbors, sharing with him a relevant similarity in terms of value. But, if the vast majority of the points present this feature, then many of these points will be identified as core elements. It is reasonable to think that we obtain a single compact cluster because there might be a "chain" of core points and border points sufficiently long to link core points having a very high value with core points having a very low value, for each dimension. This might explain the poor performance of DBSCAN over Glasgow Norms.

Even though DBSCAN's performance on the data-set is poor[7], we can still exploit noise points for an outlier analysis and proceed to compare the outliers found through the DBSCAN with the ones identified using the box-plots in the first section. DBSCAN identified a total of 733 words as outliers while, by analyzing the data in the box-plots, we are able to discover that only 361 total points can be considered outliers, according to a comparison between the points' values and the percentiles of each variable. Their intersection has a cardinality of 158 and an interesting observation regarding the outliers shared by the two sets might be done: many words in that intersection are probably identified as outliers in virtue of their "gender" value or their "valence" value: for instance, some of the words identified as outliers by both methods are "wife", "Mum", "boy", "fireman", which have a very high or low degree of masculinity, and considering a different sequence, "rape", "pain", "genocide", "depression", which all share a low "valence" value. These last observations on DBSCAN performance on the data-set end the analysis of the algorithm in question.

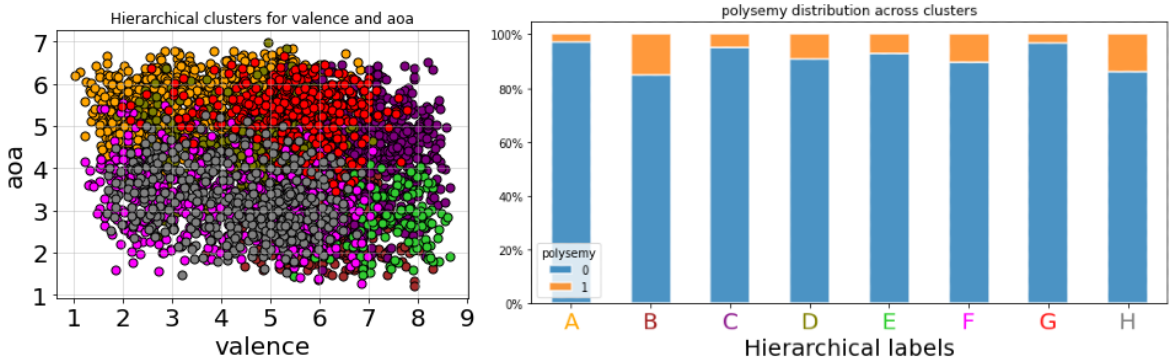
[7]This is the reason why we are not presenting how categorical variables distribute across the clusters, since just a single one was obtained.

2.3 Hierarchical clustering analysis

Hierarchical clustering has the aim of generating multi-level partitions for the original data-set. The visualization tool used for this methodology is a dendrogram graph. We therefore present here different dendrograms obtained from applying an agglomerative hierarchical clustering technique using euclidean distance as a reference metric to compare the samples, considering three different metrics in order to evaluate inter-cluster distance during the partitioning: we first consider a single-linkage method, where the similarity of two sets of points is given by the similarity of the two respective closest points on the "border" of the sets; then a complete-linkage, where similarity is computed considering the two most distant respective points; at last, we consider the result obtained with average-linkage, where we compute the average of all the possible distances and decide whether or not to merge[8].



The single-linkage method offers the worst performance overall, as evident from the graph. The other two cases display similar performances both in terms of number of clusters and in terms of cardinality of each single cluster: in both cases, there are examples of larger clusters against rather smaller ones. Still, we decided to proceed taking as a reference a complete-linkage clustering and the two following visualizations below are used with the aim of comparing hierarchical clustering results with K-medoid ones.



[8] In all the cases, numbers in brackets indicate the amount of elements belonging to a particular cluster.

We notice how the larger amount of clusters affect the partitioning process: there are almost no clear borders separating the groups isolated by the algorithm. We also emphasize how the grey and pink clusters tend to overlap, and the red and dark green ones as well. The additional graph is provided in order to show how polysemous words distribute across the clusters. In a sense, the graph provides interesting information: it shows us, consistently with results obtained with K-medoids, that larger clusters (B, F, H) tend to encapsulate a higher percentage of polysemous words.

2.4 Conclusive remarks

We are finally in a position to briefly summarize the results obtained by the three algorithms for a broad qualitative comparison. The worst results were obtained with a DBSCAN clustering method for the simple reason that it produced a single cluster obtained, with a small percentage of noise points departing from it. Even though a DBSCAN algorithm is not capable of performing a satisfactory division of the samples, it is still interesting to notice how many noise points identified through it were also labeled as outlier by previous methods used in data understanding part. Hierarchical clustering with complete-linkage is surely more efficient, but suffers of some issues due to the absence of definite borders between clusters. K-Medoids probably shows the best kind of results, providing us with 4 clusters more or less balanced and which, as visualization tools show, seem to be better separated from each other than what we could observe for clusters derived from previous methods[9]. Hence, we could consider K-Medoids a satisfactory clustering method for the Glasgow Norms data-set, even if the partitioning it produced is not the most desirable one.

3 Classification task

3.1 Decision tree classification analysis

This section has the aim of showing and discussing the performance of a decision tree approach for the resolution of a classification task over the Glasgow Norms data-set, taking into account as target variable the "polysemy" of a word and using the other main interesting features as a foundation to build split-conditions for the decision tree. Before actually assessing the performance of the algorithm, we first discuss some pre-processing work made upon the training set in order to enhance the predictive ability of the model under assessment. We observe that the data-set under analysis is strongly unbalanced with respect to the target class (with approximately 92% negative instances against just 8% positive ones): given such distribution, the classifier will surely be strongly biased towards the identification of negative values of the target attribute. One possible solution is performing a random oversampling of the training set with respect to the target variable: this consists in randomly duplicating instances of the minority class and expanding the training set with them in order to solve the issue. In the following paragraphs of this section, we will take into account a decision tree classifier trained on an upsampled version of the original data-set, since the performance metrics without having made any upsampling result quite low[10].

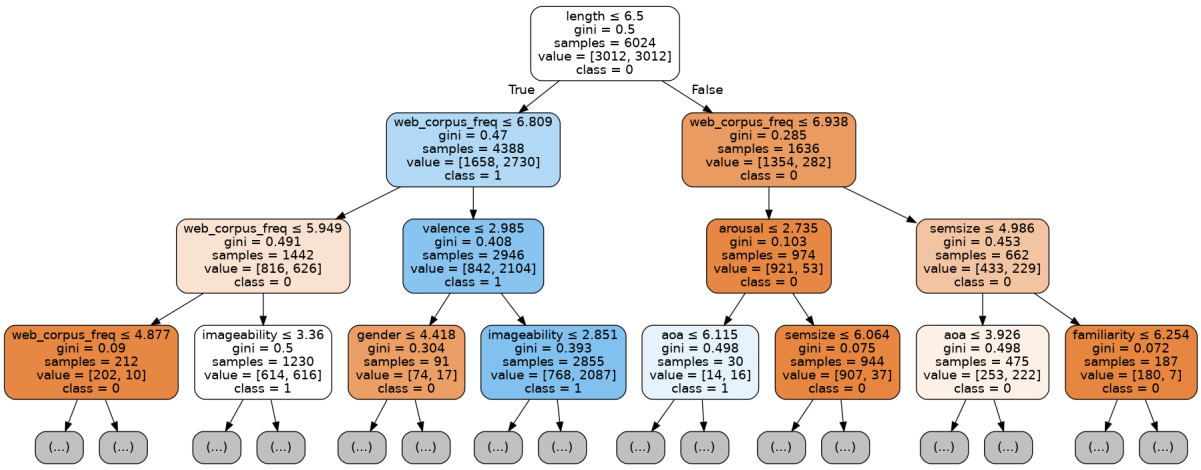
First of all we tuned the parameters of the classifier according to the results of a grid search, using the $F1$ score (harmonic mean of *Precision* and *Recall*) as a reference[11]. It is important to mention a detail concerning

[9] For completeness, we might specify Silhouette score for each clustering methods identified: Hierarchical: 0.06; K-Medoids: 0.16; DBSCAN: 0.12.

[10] For instance, the best classifier we could build with respect to the raw data-set had a $F1$ score of 0.13, a *Recall* of 0.08, a *Precision* of 0.48 and an *Accuracy* of 0.50, all examples of unsatisfactory results on the training set. This motivates the choice of using an oversampling methodology.

[11] Performing a grid search estimation allows us to use different combinations of parameters in order to find the best configuration with

the usage of a grid search method in order to identify the right parameters: the upsampling of the data-set has not been done before applying the grid search method but was rather made part of the process for the tuning. The reason is the following: *first* performing an oversampling of the data-set and then splitting the data-set itself for cross-validation might lead to duplicate instances of the same element ending up both in the training set and the validation set, eventually resulting in too optimistic and definitely wrong results of the classifier's performance. The solution in this case is upsampling the training and validation sets obtained *after* having performed a split for cross-validation purposes, making oversampling part of the cross-validation process: this eludes the problem discussed above and allows us to draw more plausible conclusions[12]. We can now start to analyze the main characteristics of the classifier working on the upsampled data-set: the three most important features identified for the splitting, considering the *Gini* impurity measure, are respectively "length", "frequency" and "imageability"[13] and it is relevant to see how their importance affects the building of the respective decision tree below:



Having performed a random oversampling on the original data-set, the first node contains the exact same number of positive and negative values for the target class and the first split is made according to the number of letters in a word: it is possible to notice how smaller values of length lead to a left child with a higher number of positive instances. On the other hand, longer words populate the right node and just a minority of them are polysemous. Even if an upsample of the data-set has been performed, the result is still intuitively reasonable: as we have specified in the past sections, polysemous words might tend to be shorter for communication purposes. The second most important feature used for the splitting is the frequency in the *Google Newspapers Corpus*. Two slightly different thresholds have been selected for the first two children nodes and it is interesting to notice that performing a split according to the frequency on the most positively populated node leads to sensible results: most of the polysemous words end up in the right node after the split, that being the node including the previous samples that are above the chosen threshold frequency-wise. Polysemous words tend in fact to have higher frequencies, being used in different contexts with different purposes. At the same time, although the split on the right node leads to two children both having a higher percentage of negative instances, we see that the great majority of the

respect to the maximization of a certain score measure: we perform different cross-validations and rank the best parameters according to the highest mean cross-validation score. We also specify that 30% of the original data-set was isolated for testing purposes, while the remaining 70% was used for training task and parameters tuning.

[12]For the sake of completeness, here we specify the best parameters chosen for the classifier on the upsampled data-set: the best parameters are 5 as the minimum number of splits that a node must have in order for a split to be performed and 1 as the minimum number of samples to be in both nodes after a split. Furthermore, the tree must have a maximum depth of 4.

[13]For completeness, we add here the degree of importance of these features: they are respectively 0.443, 0.282 and 0.097. In order to compute an attribute's degree of importance, for each split involving that attribute we take into account the error reduction of the node derived from that split, along with the number of samples that end up being in the node at the end of the split.

positive instances of the starting node end up in the child which hosts more frequent words, a result consistent with what happens with the left node. For the last depth level of the tree displayed in the figure, some brief comments concerning how the splits have affected the *Gini* measures could be of some interest: giving a look at the branches on the right, we notice that a relevant number of generated pairs tend to share quite unbalanced values of *Gini* impurity. It is clear to notice that some nodes showing an incredibly low level of impurity (0.09, 0.075 and 0.072) are derived from splits that also lead to quite impure nodes (0.5, 0.498). Without displaying the whole tree, we specify that this kind of tendency is present in the last step of the tree-construction and a certain amount of heterogeneity in this sense is present in some of the leaves: many of them are pure and quite populated, many others are pure but present just few instances in them, and these same characteristics are present for less pure leaves, too.

Having built the classifier, we can go on with an evaluation according to three main performance metrics, *Precision*, *Recall*, *F1*. The table below is used to convey this information, showing the result of the classifier both on the upsampled training set previously used for the grid search and on the test set. The three main measures are taken into account with respect to the negative and positive value of the target class:

Training measures	Precision	Recall	F1
0	0.93	0.57	0.71
1	0.59	0.96	0.80

Testing measures	Precision	Recall	F1
0	0.97	0.57	0.72
1	0.14	0.82	0.24

Furthermore, the *Accuracy* score on the upsampled training set reaches the value of 0.76, but on the test set we have a slightly lower value, 0.59. Giving a general look at the results, we must admit that they do not look as promising as we would have desired, even after having adopted an oversampling methodology to tackle the lack of positive instances for the target variable: *Precision* and $F1$, considered with respect to the positive values of the target class, show the most significant kind of distance from training values. As emphasized before, the original data-set is strongly unbalanced, with the majority of the target class values being negative, affecting also the distribution of the test set. This might explain the very low level of *Precision*: since most of the instances are negative, many of the positive guesses the classifier will perform will end up being wrong ones. On the other hand, the tendency with *Recall* is different: as one of the next confusion matrices will confirm, the number of false negative instances is rather small. The model tends to produce less errors over the positive instances simply because there is a significant shortage of them.

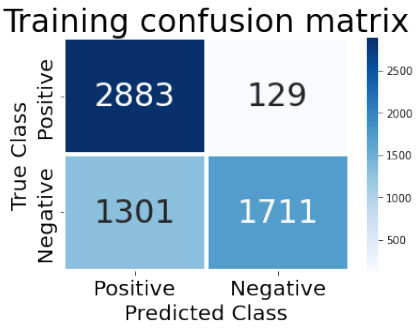
It might be interesting to notice how the results on the test set could be compared to the ones derived from a dummy classifier, namely a classifier that randomly labels the elements in a uniform way, i.e., assigns to a sample the same probability of being a positive or a negative instance of the target class.

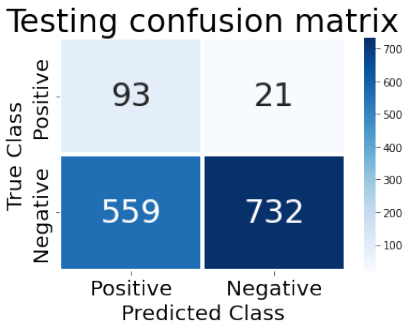
Testing measures	Precision	Recall	F1
0	0.91	0.51	0.65
1	0.07	0.44	0.12

We can infer that even if the overall performance of the classifier having the best selection of parameters are not so satisfactory, the model still performs better then a random dummy model. Also with respect to the *Accuracy*

value we observed better results, having the dummy classifier an accuracy level of 0.48 when evaluated on the test set.

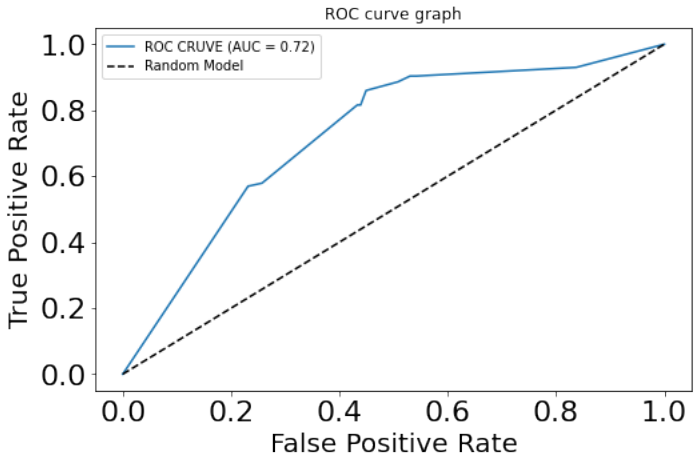
It is therefore possible to proceed with a deeper analysis regarding the performance metrics defined above, specifically investigating how true/false negative and true/false positive distributes across the training and test set. For this purpose we introduce two confusion matrices in order to better visualize the results.



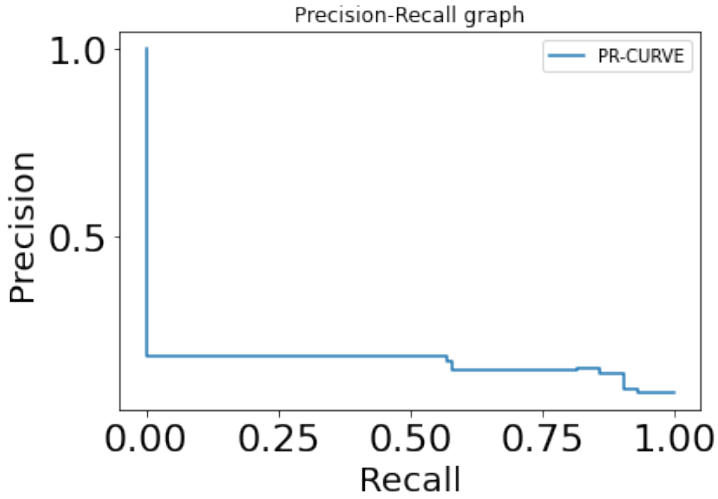


Focusing especially upon the confusion matrix obtained for the test set, we see that the model does not present so much difficulty in the identification of negative instances of the class: in fact, as observed in one of the tables above, the *Recall* value is satisfactory and this is consistent with results shown by the matrix. More problematic issues arise with the classification of positive instances, as the *Precision* measure suggests: where for the training set the number of true positives is more than the double of the number of false positives, performances on the test set are much worse - most of the errors made in the evaluation phase consist in wrongly identifying a negative instance as positive, rather then a positive one as negative. We shortly discussed the situation shown above, linking it with the unbalaced nature of the data-set.

The last kind of analysis we can perform concerns the ROC curve, which allows us to understand how the number of false positives and the true positives tend to change according to various possible thresholds used to assign a value to the data-set's samples: given a sample, the classifier assigns to it a certain probability of being a positive instance of the target class, but the samples gets actually labeled as positive only if the value assigned is above a certain threshold. Therefore, by changing the threshold, the overall number of positive instances will be affected: setting the threshold to 0.0 will maximize the *Recall*, having the highest number of true positives and the lowest number of false negatives; in a specular way, setting the threshold to 1.0 will nullify the number of true positives and false positives, since no value could be higher then that very threshold; values in between will lead to intermediate situations displayed in the graph. This gives us a visual aid for understanding how the classifier behaves: if lowering the threshold leads to quite consistent gains in terms of true positive with respect to false negative, i.e., the model still tends to correctly predict the samples even if the probability requirements are less stringent, then we might have an additional reason for being satisfied with its performance.

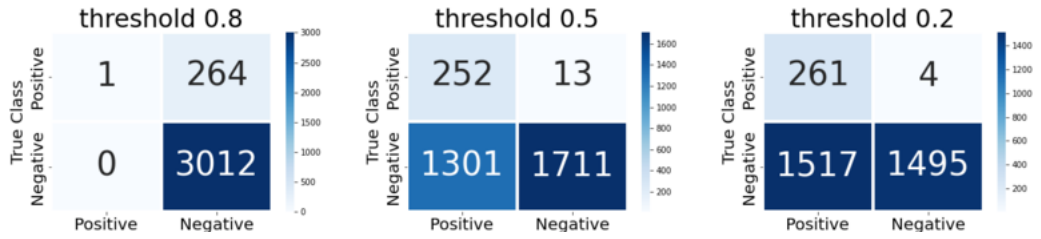


If we take a look at the ROC curve, we observe that in general lowering the threshold leads to acceptable gains in terms of true positive instances, even though especially in the middle section of the line the number of true positives tends to grow at a smaller rate. We introduced here also a *Precision-Recall* graph[14] in order to compare how these values tend to vary with different thresholds. The performance in this last case is poor:



As expected, the *Recall* value increases as the threshold is lowered, since the number of false negatives ends up being reduced: in the graph, the increasing process is stable and is paired with a reduction of the *Precision* value. It is easier to understand this last tendency once the respective confusion matrices for different relevant thresholds are taken into account: we observe the relevant change in terms of true and false positives when the threshold is shifted from 0.8 to 0.5, coupled with a reduction of the overall number of false negatives responsible for the *Recall* growth.

[14] For higher values of the threshold we will get a lower amount of false positive instances, resulting in an increase in terms of *Precision*, but also a decrease in terms of *Recall*, since many actual positive values won't be considered as such. Reading the graph left to right, we see how lowering the threshold affects these two metrics for our classifier, starting from the best setting for the *Precision* and reaching the best one for *Recall*.



We are now in the position to provide a final comment over the decision tree classifier method applied to the Glasgow Norms data-set: we emphasize how the application of an oversampling methodology, as noticed at the very start of the section, leads to overall better results then simply using the original unbalanced data-set. Still, general performances are not so satisfying, as both ROC curve and *Precision-Recall* graph demonstrate. One positive aspect is related to the better results of the decision tree when compared with a dummy random classifier, even though this is not sufficient, especially looking to the actual results of performance metrics used in the evaluation, to label our classifier's results as good.

4 Pattern mining task

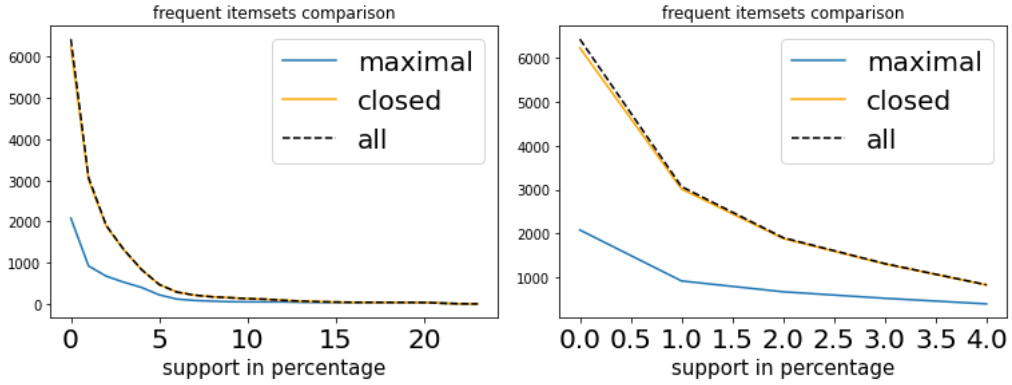
4.1 Frequent pattern extraction and analysis

In this section, we will be focusing on the performance of the apriori algorithm on the Glasgow Norms data-set for pattern mining purposes. Before identifying the frequent itemsets, we must face the problem of discretizing continuous attributes that comprise our feature space. We performed a discretization over the values of continuous attributes, considering 4 intervals to be used for mapping continuous values onto single bins. For a simpler analysis over the "length" attribute, we decided to avoid considering the exact length of every word and rather partitioned them as short, medium and long words: the criterion used for the partitioning is the same employed in clustering analysis. As a first task, we consider here the most frequent itemsets with respect to their support values expressed as a percentage.

Frequent itemsets	Support in percentage
Length: medium, Non-polysemous	56.21%
Length: short, Non-polysemous	32.20%
Age of acquisition: (5.152, 6.971], Non-polysemous	24.56%
Dominance: (1.940, 4.529], Non-polysemous	24.17%
Length: medium, Age of acquisition: (5.152, 6.971], Non-polysemous	17.32%

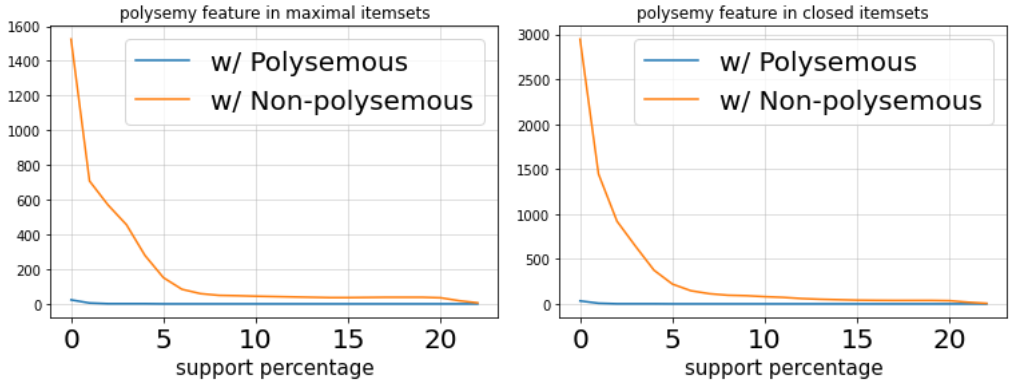
The first two most frequent itemsets show quite predictable results: it is not surprising to find out that most of the words are of short-medium length and are non-polysemous ones, especially considering the original distribution of the attributes' values in the data-set. More interesting results concern itemsets having a support of about 24%: as we can notice, properties such as a rather high "age of acquisition" and smaller values of "dominance" tend to characterize non-polysemous words. This seems to be consistent with what is observable on the correlation matrices displayed in the first section on data understanding: "polysemy" is in fact not positively correlated with the two variables in question.

For a general analysis of the results, additional graphs are provided below displaying a simple study regarding the relationship between various increasing support thresholds and how the change affects maximal and closed frequent itemsets, too[15]. For this purpose, we considered increasing support thresholds up until reaching one of 25%, considering a minimum itemsets length of 2.



We immediately notice the strong divergence between the whole number of frequent itemsets and the maximal ones, a rather predictable result: we observe how for very low support thresholds, only about 2000 of the starting frequent itemsets can be regarded as maximal. A more interesting relationship is the one linking closed itemsets with the whole collection of frequent sets: when we consider very small thresholds, almost every frequent itemset is closed, such that the set of frequent itemsets and the set of closed frequent itemsets are almost the same. Rising the threshold value soon leads to a complete identity of the two sets. The plot on the right is provided in order to emphasize this small difference between the two, harder to notice when larger ranges of thresholds are visualized.

Moving on, we study how the polysemy values are distributed across maximal and closed frequent itemsets.

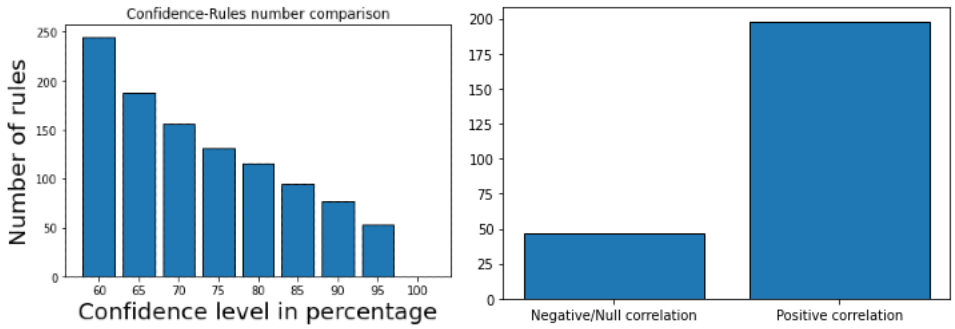


The results of the visualization are quite intuitive, taking into account once again the distribution of the samples in the Glasgow Norms data-set: most of the words are non-polysemous so it is reasonable to find out that maximal frequent itemsets that have as a member the value "Non-polysemous" are much larger in number then the ones having "Polysemous" as a member. The same reasoning can be applied to closed frequent itemsets.

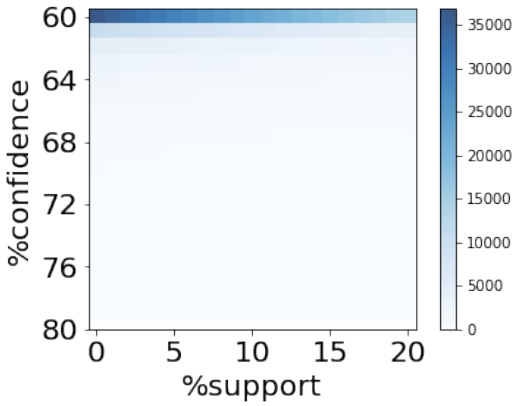
[15] An itemset is maximal frequent if none of its immediate supersets is frequent. On the other hand, an itemset is closed if none of its immediate supersets has the same support as the itemset in question. It is easy to see that the set of maximal frequent itemsets comprise a subset of the set of closed frequent itemsets. The same is true for the latter with respect to the set of all frequent itemsets, given a threshold value.

4.2 Association rules extraction and analysis

Association rules starting from the previously identified frequent itemsets have been generated by the apriori algorithm: we decided to focus on rules having at least a cardinality of 2 with respect to the union of the consequent with the antecedent. In the next analysis, we try to discuss the most interesting extracted rules involving as a consequent itemsets including values that the "polysemy" attribute can assume: we are especially interested in understanding how other features might highlight a positive or negative correlation with the target variable. Hence, we first display how the overall number of rules changes according to different confidence thresholds, keeping a support value of 10%, and for a confidence threshold of 60% we check how the lift values are distributed.



From the graph, it is reasonable to observe a constant decrease in terms of number of rules as the confidence levels are set on higher values. Regarding the lift, we can observe how most of the antecedents share a positive correlation with the consequents[16]. We also provide a heatmap to display the variation of the amount of rules identified by the apriori algorithm according to different support and confidence thresholds. As it is plain, slightly higher degrees of confidence are sufficient to trigger a relevant decrease in terms of rules identified.

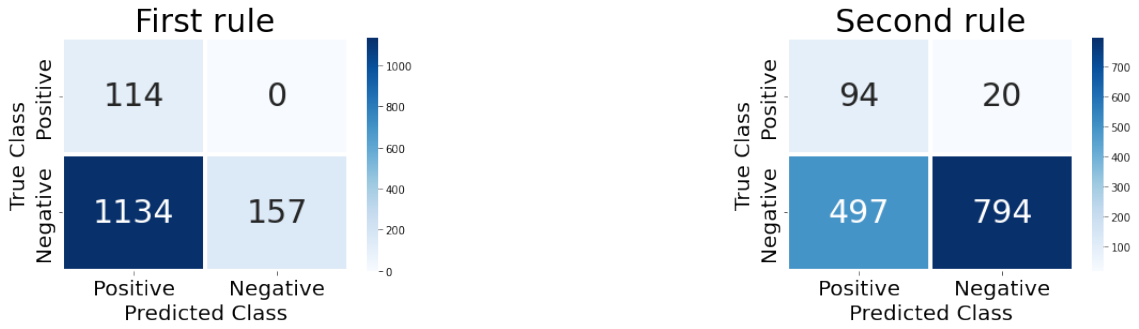


We then present a small table with the best association rules according to lift values, before focusing mostly on rules implying the itemsets with values for the target variable. The first rule reasonably links words associated with higher values of "dominance" and "arousal" to higher values of "valence"; the second links words mastered later in life, although quite frequent, with the property of being not very familiar; the third rule, similarly to the first one, establish a relationship with high values of "valence", "arousal" and "dominance".

[16]We consider a correlation between itemsets to be a positive one in terms of lift when the lift value is higher then 1. On the contrary, we speak respectively of a negative or null correlation, when the lift is less then 1 or equal to 1.

Association rule	Support, Confidence and Lift values
{Dominance: (5.6, 8.371], Arousal: (5.419, 8.177], Non-polysemous} $\Longrightarrow$ {Valence: (6.088, 8.647]}	10.25%, 80%, 3.2
{Age of acquisition: (5.152, 6.971], Frequency: (4.105, 6.223]} $\Longrightarrow$ {Familiarity: (1.646, 4.706]}	10.05%, 76%, 3.06
{Valance: (6.088, 8.647], Arousal: (5.419, 8.177]} $\Longrightarrow$ {Dominance: (5.6, 8.371]}	10.70%, 74%, 2.9

Since for desirable support and confidence thresholds no association rule is identified having as a consequent an itemset having as a member the positive value for the target class, we take as a reference an association that actually establishes a correlation with the negative value for "polysemy" and decide to assign positive values to the samples in the test set for which that particular rule does not apply. Therefore, we have considered the rule with the highest lift value, i.e., the rule $X \Longrightarrow$ {Non-polysemous} such that the conditional probability $P(\{\text{Non-polysemous}\}|\, X)$ over the $P(\{\text{Non-polysemous}\})$ is the highest, given the support and confidence threshold previously fixed, and as a different rule the one having the highest support expressed as a percentage. The first rule identified in order to perform the classification task is the following one, having a confidence of 99% and a lift of 1.08: {Age of acquisition = (5.152, 6.971], Frequency = (4.105, 6.223]} $\Longrightarrow$ {Non-polysemous}. The second rule, with a support of 52.21% is {Length: medium} $\Longrightarrow$ {Non-polysemous}[17].



The first rule performs quite badly: it is sufficient to check the related confusion matrix above to realize the main problem related with using this rule as a classification tool: too many non-polysemous words are actually labeled as having multiple meanings. On the other hand, the second rule looks more promising and even similar to what we obtain using a decision tree: there is still a relevant tendency in identifying negative values as positive, but the overall results look better balanced[18].

[17] We remind here that frequency values have been transformed using a $\log_{10}$ function. The interval taken into account results from a process of discretization, as we specified in the first part of the section.

[18] For the sake of completeness, we specify here the main performance metrics regarding the *Precision*, *Recall*, *F1* and *Accuracy* scores of the rules considered: 0.09, 1.00, 0.17, 0.19 for the first rule; 0.16, 0.82, 0.27, 0.63 for the second rule.

References

[1] Graham G. Scott, Anne Keitel, Marc Becirspahic, Bo Yao, and Sara C Sereno. The glasgow norms: Ratings of 5,500 words on nine scales. *Behavior research methods*, 51(3):1258–1270, 2019.