Data Mining report on Glasgow Norms

Agnese Marchionneschi
586648

Michele Papucci
544376

Anno accademico 2021/2022

Contents

Chapter 1

Data Understanding and Preparation

In the first part of the project we analyzed the features of the raw dataset and pre-processed the data for the rest of the analisys.

1.1 Data Semantics

Glasgow Norms is a dataset composed of 4682 English words rated on nine different psycholinguistic features, two of which, gender association and semantic size, are novel to this dataset. Glasgow Norms also includes 379 ambiguous words[1] (Scott et al., 2018b).
In Table 1 is possible to see that all the nine psycholinguistics features are represented as continuos variables in their rating range.
For each of the word, the psycholinguistic feature represent the average value chosen by the participants for that word. The ranges of the features varies between them, and the specific meaning of each feature by rating is also found in Table 1. The dataset also presents two more variables for each word:

- Polysemy: it is a categorical binary variable which indicates if the word has multiple meanings;
- Web Corpus Frequency: is the frequency of a word in the Google Newspapers Corpus.
- Length: the length of a word expressed in the number of characters.

We've also introduced a new feature for each word called *Sentiment*. It has been introduced using Stanza Python library' *SentimentProcessor* (Qi et al., 2020). The variable has three possible values which represent how the word is, on average, perceived:

- 0 - Negative perception;
- 1 - Neutral perception;
- 2 - Positive perception;

Sentiment is semantically very similar to the already present valence feature, but we've chosen to introduce it nonetheless, to see if there's any kind of correlations between the two. In fact, while semantically similar, the values comes from two very different backgrounds. We also chose Sentiment as our target variable for the rest of the analysis to try to understand how the distribution of the nine psycholinguistic variables could affect the sentiment perception of a word out of context.

[1] An ambiguous word is a word that can be interpreted with multiple meanings. e.g.: *Toast(bread)* or *Toast(speech)*

Name	Type	Range	Description
Arousal	Numeric	1 - 9	Arousal is a measure of excitement versus calmness. A word with a high arousal value makes you feel stimulated, and a word with a low arousal value makes you feel relaxed or calm
Valence	Numeric	1 - 9	Valence is a measure of worth. A word with high valence is considered positive or good, while a word with low valence is considered negative or bad
Dominance	Numeric	1 - 9	Dominance is a measure of the degree of control you feel. A high dominance value word can make you feel dominant, while a low dominance word makes you feel controlled or influenced
Concreteness	Numeric	1 - 7	Concreteness is a measure of how concrete or abstract something is. The scale goes from very abstract to very concrete.
Imageability	Numeric	1 - 7	Imageability is a measure of how easy or difficult the word is to imagine. To a low imageability value corresponds a hard to imagine word and to a high imageability value corresponds a easy to imagine word.
Familiarity	Numeric	1 - 7	Familiarity is a measure of how familiar a word is. A word with a high familiarity value is a word you see/hear often and it is easily recognisable, while a word with a low familiarity value is relatively unrecognisable.
Age of Acquisition	Numeric	0- 7	A word's age of acquisition is the age at which that word was initially learned. The scale is defined as a series of consecutive 2-year periods from the ages of 0 to 12 years, and a final period for every word learned at 13 years or older
Semantic Size	Numeric	1 - 7	Size is a measure of something's dimensions, magnitude, or extent. A word with a high semantic size should be describing something perceived big, while a word with a low semantic size should describe something perceived small.
Gender Association	Numeric	1 - 7	A word's gender association is a measure of how strongly its meaning is associated with male or female behaviour. Low gender association values means a word is perceived very feminine while high gender association values means a word is perceived very masculine.

Table 1.1: Features description.

1.2 Variables Distribution

With the goal to unveil any kind of relations between the variables, various kind of data visualization tools have been used to explore the data.
Both the Polysemy and the Sentiment categorical variables are very unbalanced as shown in Figure 1.1. In fact, for polysemy we have only 8.1% of the words that are polysemic while 91.9% of the words are not, meanwhile for sentiment most of the words have been classified as neutral, and only 8% and 7% of the words have been classified as negative and positive. This will have an effect in how accurate some of the later analysis will be.

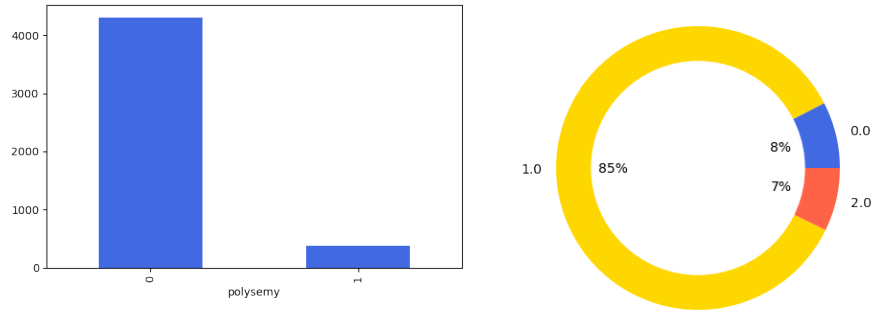

Figure 1.1: Distribution of categorical variables in the dataset

As shown in Figure 1.2, most of the nine psycholinguistic variables follow a normal distribution. Concreteness and imageability stands out, as they seem to have a similar bi-modal distribution with two distinct peaks, one around 3 and one around the end of the scale. These two similar distribution are further explored in section 1.5 since the two variables have the strongest correlation in the dataset. Familiarity shows a negative skew, with most of the data located around the end of the scale. This might be explained by the selection process of the words for the dataset: if the majority of the chosen words are very common, it should be obvious that most of the participants were familiar with the presented words, thus explaining the high average familiarity values.
That to a common words (with a higher frequency) may be related a higher Familiarity score is first of all expected from the definition of Familiarity (see Table 1.1), but will be also be confirmed in section 1.4, were we've found out that Familiarity and the logarithm of the Web Corpus Frequency have a relatively high correlation.
However since in Scott et al., 2018b the word selection for the dataset isn't detailed, we can't actually prove that the negative skew of Familiarity is related to the supposedly averagely high frequency words chosen to create the dataset.

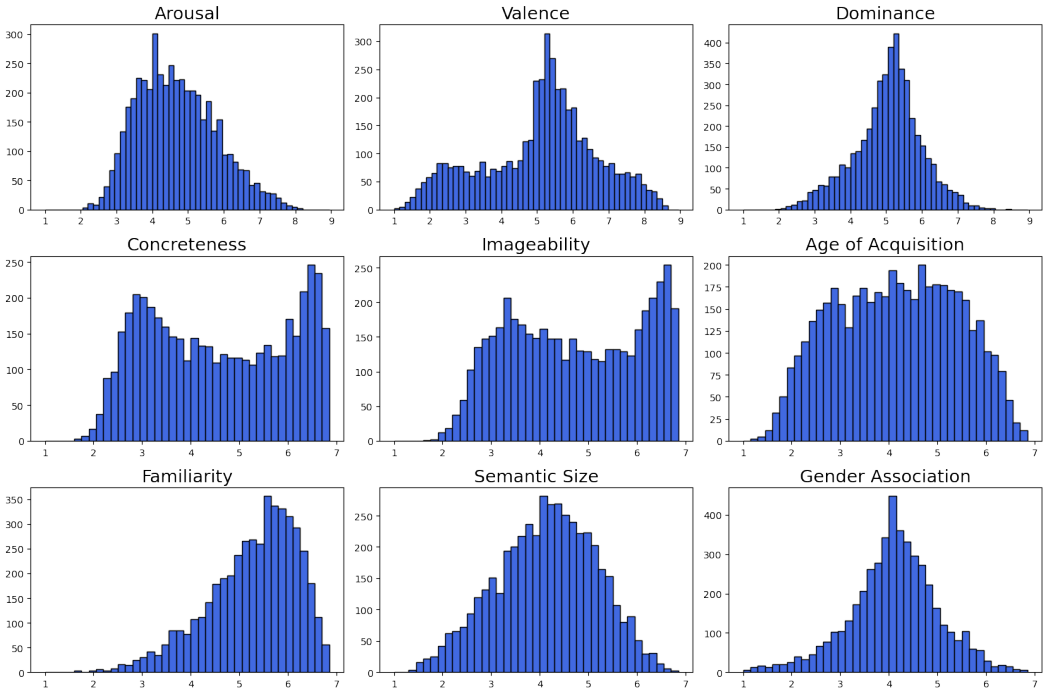

Figure 1.2: Distribution of the nine psycholinguistic features

1.3 Variable Transformation

Analysing the distribution of the various feature of the dataset, we've noticed a strong positive skew in the Web Corpus Frequency. This was expected, since it's well known in scientific literature that Word distribution inside corpora follow Zipf's Law (Lenci et al., 2005).

To address this, we decided to apply a base 10 logarithm to the Web Corpus Frequency. As shown in Figure 1.3, with the transformation we maintained the order of magnitude of the frequency, while balancing the skewness of the data, obtaining a tighter normal distribution.

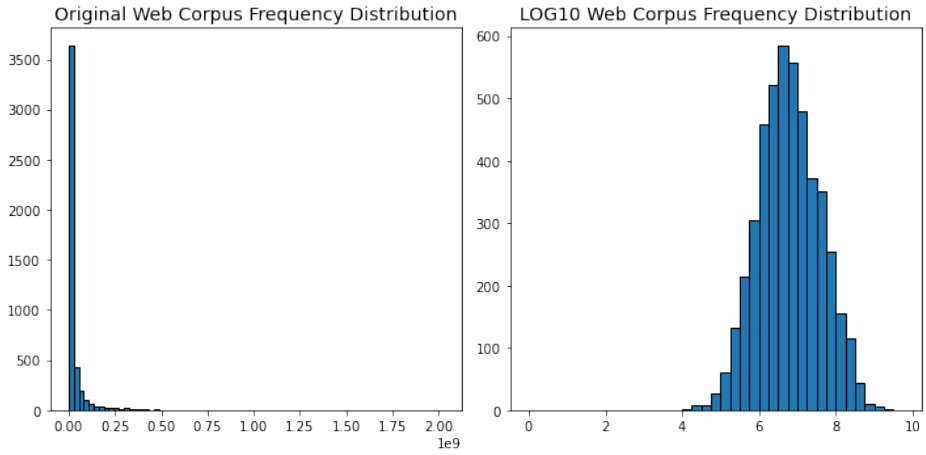

Figure 1.3: Web Corpus Frequency Distribution before and after the base 10 logarithmic transformation

1.4 Data Quality

During this phase of the analysis we've checked the dataset for missing values and outliers.

1.4.1 Missing Values

The dataset presented only 14 missing values for the Web Corpus Frequency variable. To handle these missing values we decided to fill them with the average frequency values grouped by Familiarity. The choice of using Familiarity has been made because of the relatively high correlation between the now less-skewed Logarithmic Web Corpus Frequency and Familiarity, which is shown in Figure 1.4.

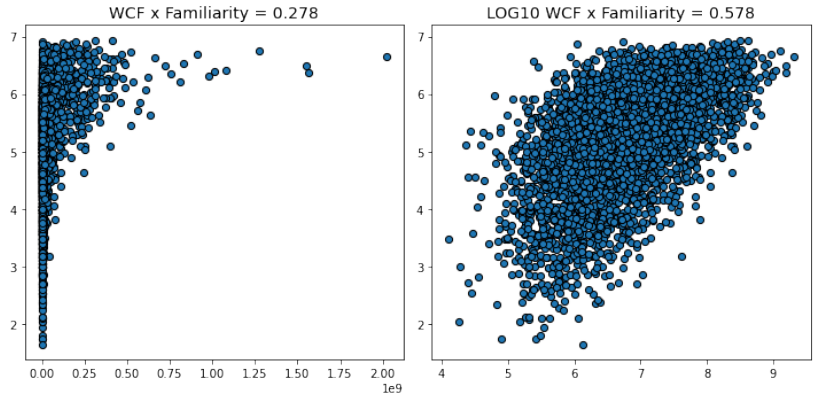

Figure 1.4: Correlation between Web Corpus Frequency and Familiarity before and after the base 10 logarithmic transformation

1.4.2 Outliers

The assessment of the outliers has been made visually by analyzing boxplots for each feature. Since the psycholinguistic features are limited within a set range we didn't have *true outliers*, but something interesting emerged nonetheless from this analysis.
As shown in Figure 1.5, the valence feature has many instances below the adjacent lower value. From its definition in Table 1.1 we know that low valence values are used to describe negatively perceived words. Looking at these outliers we found word like *rape*, *genocide*, *murderer*, *racist* and *cancer*, which are words whose valence values are expected to be low since they belong to a negative semantic dimension. Therefore these instances are not considered outliers but have a high information content.

We have drawn similar conclusions for the arousal attribute that shows instances that exceed the adjacent upper value. We looked at the five highest arousal value words and we found: *passionate*, *love*, *kiss*, *spectacular*, *aroused*. From the definition of the arousal value (see Table 1.1) we can say that high arousal values for these words was expected.

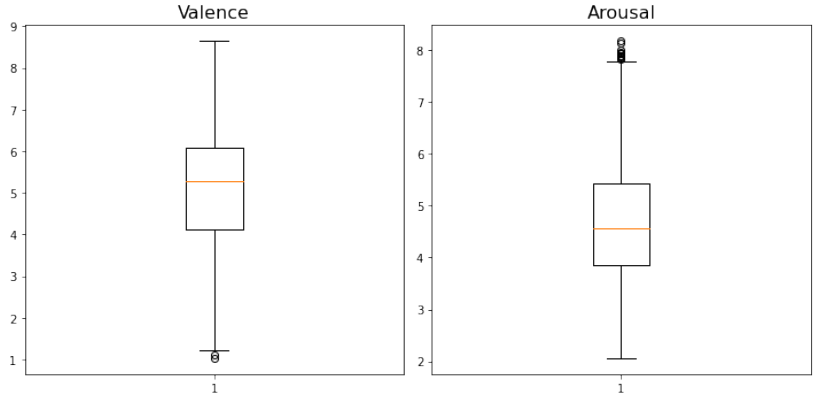

Figure 1.5: Boxplots for Arousal and Valence

As shown in Figure 1.6 the gender association feature presents two clusters of outliers in the extreme areas of the scale. We looked at the words corresponding to the five highest gender association values and found out *man*, *king*, *father*, *uncle* and *penis*. Since from its definition we know that high gender association values correspond to very masculine words we can say that these values are appropriate for the words. The same can be said for the low gender association values in the lower cluster, where the five lowest gender association values words are *lady*, *mother*, *mom*, *girly* and *woman*.

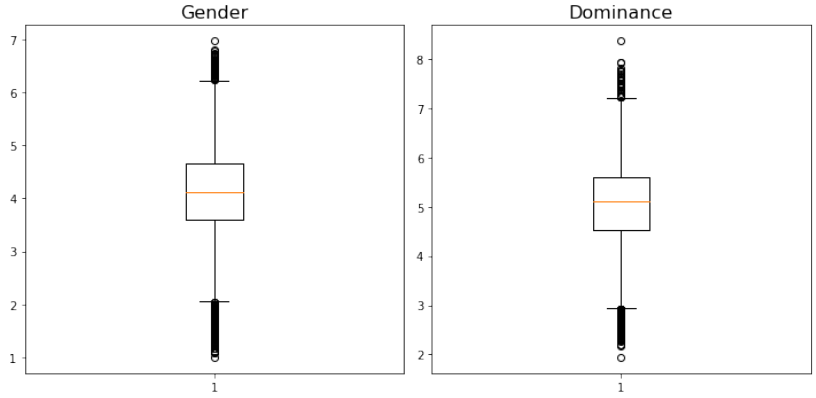

Figure 1.6: Boxplots for Dominance and Gender association

Dominance presents similar clusters to the ones of the Gender Association variables. We did a similar analisys as the one just discussed for gender association that yielded the same results: all of these value are not true outliers and have plausible scores, that are highly informative. As to why Gender Association and Dominance presents these kind of "clusters" at the extreme of the scale it may be that these features (masculine vs feminine and controlled vs dominant) are more usually interpreted and presented as binary, or categorized, values leading to a higher difficulty for the participants to pinpoint values on a larger scale.

1.5 Pairwise correlations

We started the analysis by looking at the correlation matrices created with Pearson's correlation coefficient and Spearman's rank correlation coefficient. To do so, we removed the categorical values from the dataset which can't be analyzed with these two measures. We then created the two correlation matrices between the nine psycholinguistic feature plus the logarithm of the Web Corpus Frequency. As we can see in Figure 1.7 and in Figure 1.8 the two correlation matrices are very similar and we didn't noticed anything useful by this comparison.

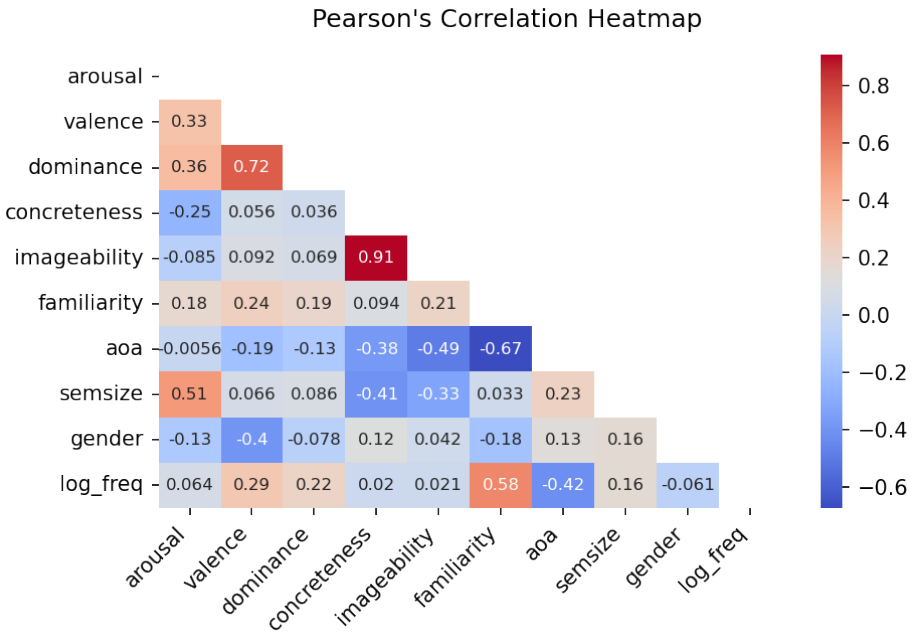

Figure 1.7: Pearson correlation coefficient matrix

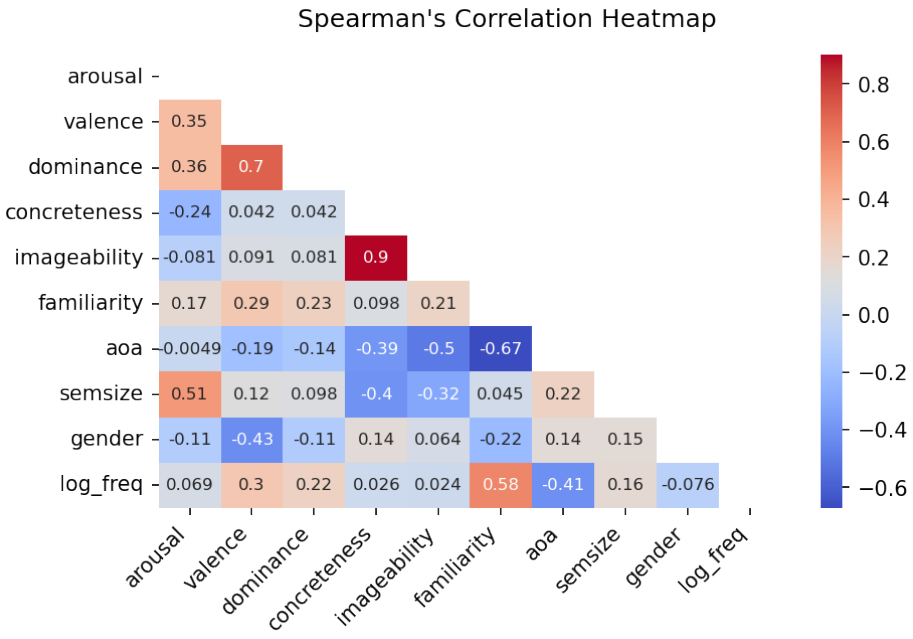

Figure 1.8: Spearman's rank correlation coefficient matrix

From the correlation values we can see that the highest correlation is the one between imageability and concreteness, basically this means that the more concrete the meaning of a word is, the easier it is to imagine and vice versa. This is an expected correlation since it's well known in psycholinguistic literature and it has been thoroughly studied in the past (e.g., Altarriba et al., 1999 and Richardson, 2007).

An interesting correlation is the one between age of acquisition and familiarity. Since it's a negative correlation it means that words learned at a younger age are more likely to be described as familiar. Age of Acquisition also has various degree of negative correlation with the logarithm of the Web Corpus Frequency, imageability and concreteness.

Taking this into consideration it is safe to to assume that to children tend learns earlier high frequency, more concretete and easier to imagine words, which then tend to be considered more familiar. Similar findings have already been published in various works (e.g, Gilhooly and Logie, 1980 and Stadthagen-Gonzalez and Davis, 2006)

Another interesting correlation is the slight tendency of having lower valence scores for very masculine

words and vice versa. Even if the correlation isn't that strong, is interesting nonetheless to look at some of the words with low valence and high gender association values: *aggressive*, *bastard*, *war*, *granade*, *destroyer*. At the other end of the scale we have words like: *angel*, *birth*, *actress*, *affection* and *beuty*.

1.5.1 Sentiment and Valence

We introduced sentiment as a new categorical variable into the dataset (see section 1.1) which has a very similar meaning to the already present valence feature. To see if sentiment could be a good target variable that summarize the valence feature spectrum we decided to test their relationship.
To do so, we discretized valence into a categorical value with the same classes of sentiment:

- Class 0: captures the lower end of the valence scale and is represent the negatively perceived words;
- Class 1: captures the middle of the valence scale and represent the neutral perceived words;
- Class 2: captures the upper end of the valence scale and represent the positively perceived word perceived words;

We then created a contingency table for the two variables and run a Chi squared test. From the results of the tests we calculated the p-value, which resulted lower than 2.2^{-16}. We can then reject the null hypothesis that there is no relationship between the two variables.

1.5.2 Variables elimination

We decided remove the original Web Corpus Frequency and tho only keep its logarithmic form as already explained in section 1.4. We also decided to remove the word lenght since from our analysis we didn't find it very useful or particularly insightful.
We decided to remove variables during the pre-processing of each subsequent analysis so that we could choose, based on the task, the best assortment of features to have the best results.

Chapter 2

Data Clustering

For this specific task we used three different algorithms for data clustering and compared the results. To do so it has been necessary to do a little pre-processing. Specifically we decided to remove various features to reduce the amount of dimensions and to help the cluster algorithm identify the clusters more easily. We have decided to only keep: *arousal*, *valence*, *imageability*, *familiarity* and *gender*. We chose this set of feature by trying to have variables with as little correlation between them as possible.
This line of work has been chosen to avoid redundant data, and to lower the number of dimension that make computing clustering, especially density-based and hierarchical one that have a higher complexity, more feasible.

2.1 Clustering with K-Means

The metric chosen for the analysis was the euclidean distance. To run the K-Means algorithm we had to find the optimal k value, and to do so, we used the the elbow method. That consists in plotting the Sum of Squared error for a list of increasing k value candidate and then, by visually assess in which point we have a drastic change in the SSE, we can choose the k value.
To have a better understanding of what could be the best k value might be, we also decided to plot the Silhouette Score for increasing k values (Figure 2.1). The Silhouette scores is computed by looking at the intra-cluster and at the inter-cluster distance, and goes from -1 to 1. The best score possible is 1 which tells us that we have very internally tight cluster that are very well separated.

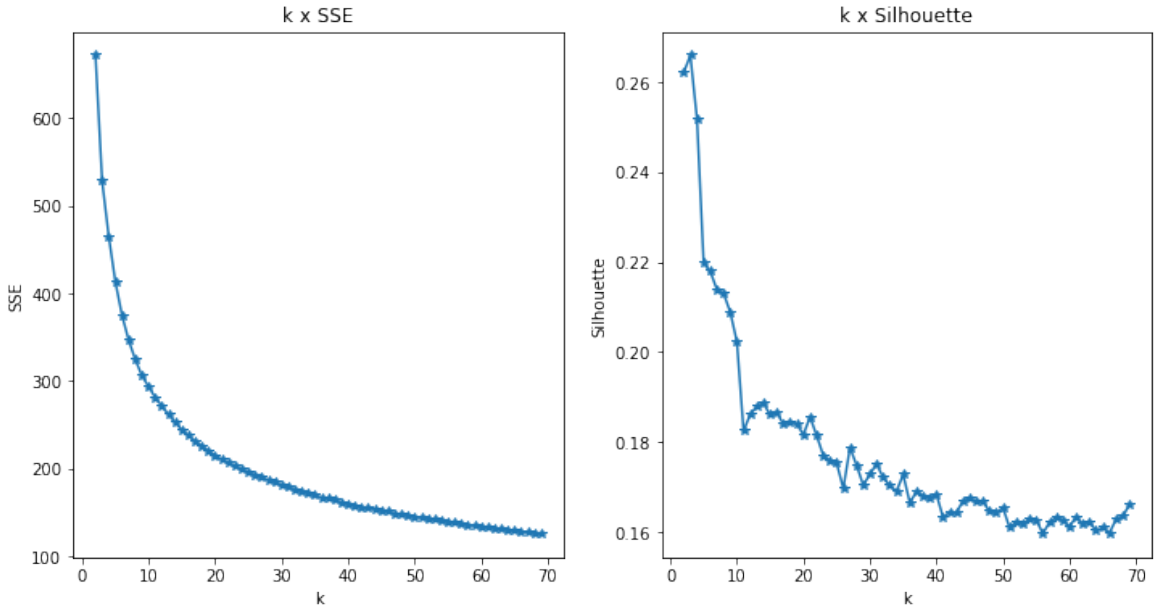

Figure 2.1: Rapport between k with SSE and Silhouette Score

By looking at the graphs in Figure 2.1, we were able to visually asses the best number of clusters that could give us the lowest SSE possible while trying to maximise the Silhouette score. In the end

by looking we chose 4 for the k value. To further validate this choice we've also looked at the line graph for the position of the centroids, and we confirmed that with $k = 4$ we had the most separated centroids (Figure 2.2).

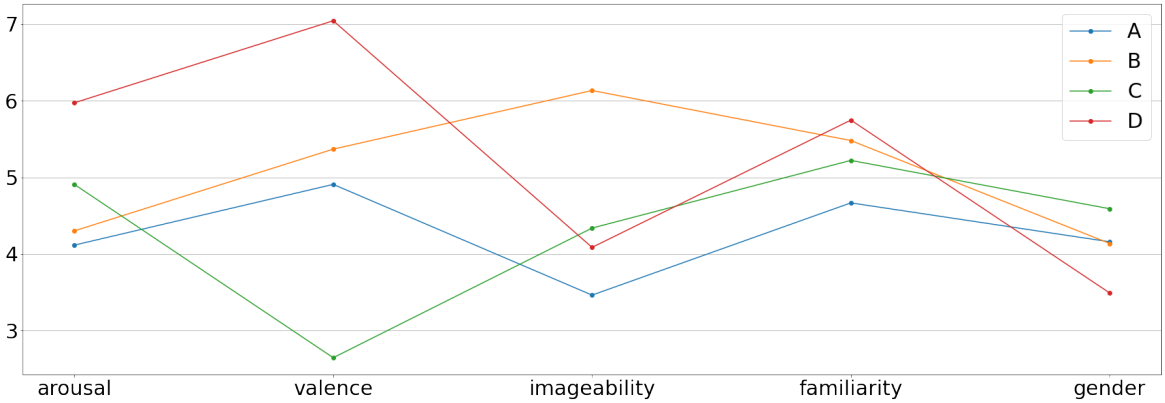

Figure 2.2: Line Graph for Centroid positions based on the featured clustered

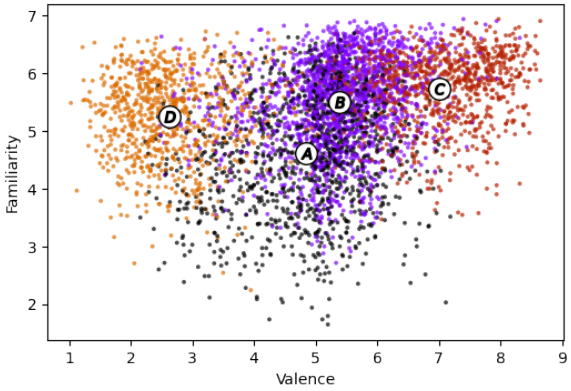

Figure 2.3

By choosing a $k = 4$ we got overall what we think is the best results possible using this algorithm, that however still isn't enough to say that we got optimal results.
First of all, we can say that we have a decently even distribution of data across the clusters (A: 803, B: 936, C: 1192, D: 1751), but we also have a rather sub-optimal SSE (464.50) and a low Silhouette score (0.25).
This is probably related to the shape of the data which, as shown in Figure 2.3, isn't spherical, but globular. The distribution of the data doesn't present uniform densities with well separated clusters without much noise between them. All of these factors badly affected how well the algorithm can perform his clustering task.

2.2 Density-based Clustering

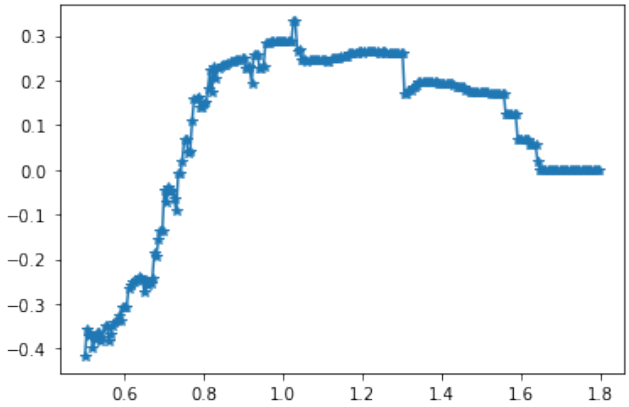

Figure 2.4: Epsilon x Silhouette Score

For this section we used the DBSCAN algorithm to perform a density-based clustering of the dataset, for which we used the same set of features as K-means.
For this type of algorithm we had to decide the values of two parameters, which are the minimum number of points (minPts) within a radius epsilon (eps). These values are used to decide whether or not a point is a core point or a border point, and therefore part of a cluster, or just noise.
To select this value we decided to plot a series of Silhouette x eps graphs with increasing minPts values to select the best assortment of variables that gave us the highest possible Silhouette score.
The selected configuration of variables is minPts = 7 and eps = 1.025 that, as shown in Figure 2.4, gave us the highest Silhouette score possible of 0.355.

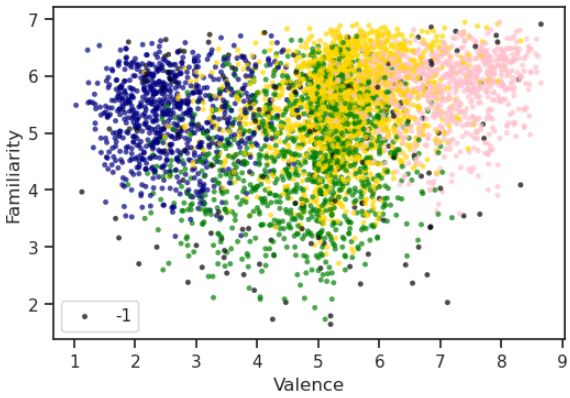

Figure 2.5

This set of parameter gave us 4 clusters, like in k-means, and the distribution of the data is also pretty similar (A: 1715, B: 1153, C: 892, D: 757 and Noise Points: 165).

2.3 Hierarchical Clustering

The last algorithm we tried was an agglomerative hierarchical clustering. As a metric of distance we chose euclidean because with continuous variable such as the one we are working with, is usually the best one to use. Then we tried using different type of linkage criteria: single, complete, group average and Ward method. In Table 2.1 we have a sum of the results we've obtained.

Linkage	**Silhouette Score**	**Distribution**
Single	0.15	A: 4666, B: 9, C: 6, D: 1
Ward	0.34	A: 1847, B: 836, C: 1193, D: 806
Average	0.17	A: 83, B: 26, C: 6, D: 4567
Complete	0.16	A: 21, B: 36, C: 4564, D: 61

Table 2.1: Silhouette scores and clusters distribution using various type of linkages

We chose to end every algorithm at 4 clusters and then decide which one was the best. We chose 4 because is the number of clusters we obtained both during the DBSCAN runs and is the value of k we chose for k-means. From the table we can see that the Ward linkage is the only one that gives us a similar silhouette score to our other attempts with different algorithms. It is also the only run that gave us a more even distribution of data across the clusters. This is also evident by comparing the dendograms of Ward linkage versus the one of our worst performer, the single linkage, in Figure 2.8

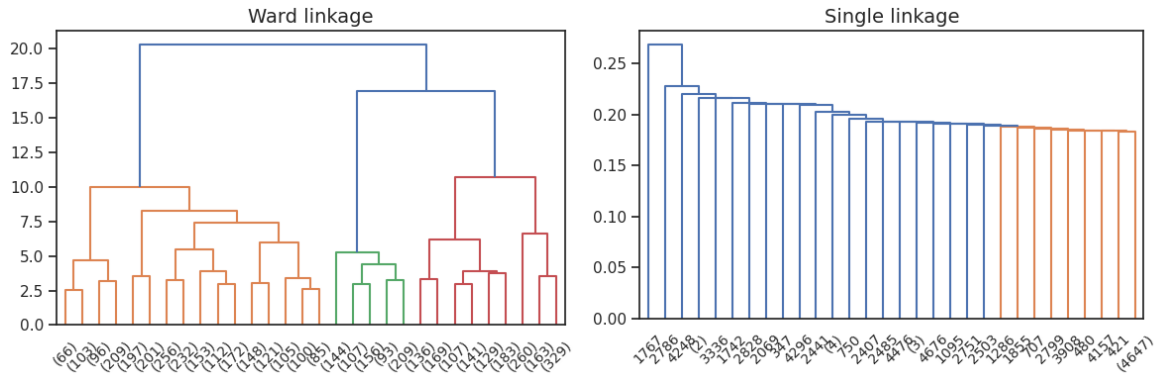

Figure 2.6: Ward Linkage and Single Linkage dendograms

We can see that while Ward Linkage only connect very far apart clusters in the last steps of the algorithm, while Single linkage tends to continuously add points to the already large main cluster connecting very afar clusters and resulting in an extremely uneven distribution of points across the clusters.

2.4 Data Clustering Discussion

After reviewing all results we can say that none of the algorithm did an overall good job at clustering the data. All of three performed in a very similar matter, with very similar distribution and Silhouette score (Table 2.2).
Our best performing algorithm was DBSCAN which was kind of surprising to us, because we thought that since the data presented a lot of different density zones the algorithm couldn't perform very well.

Algorithm	**Silhouette Score**	**Distribution**
K-Means	0.25	A: 803, B: 936, C: 1192, D: 1751
DBSCAN	0.36	A: 1715, B: 1153, C: 892, D: 757 and Noise Points: 165
Ward Linkage Hierarc.	0.34	A: 1847, B: 836, C: 1193, D: 806

Table 2.2

We've also looked at the distribution of our target variable, Sentiment, across the clusters we've found. As shown in Figure 2.7, the distribution is pretty similar across the three algorithms and shows the Neutral sentiment dominating all the clusters. This was expected from how the sentiment categories are distributed, since there are a lot more Neutral values than Positive or Negative. The other interesting thing is that Positive and Negative rarely show up in a significant way in the same clusters. This fact could be interpreted as that, since some of the features that we decided to use for this clustering task are correlated with Valence (which is semantically similar to Sentiment) are then able to express the distance between negative and positive words.

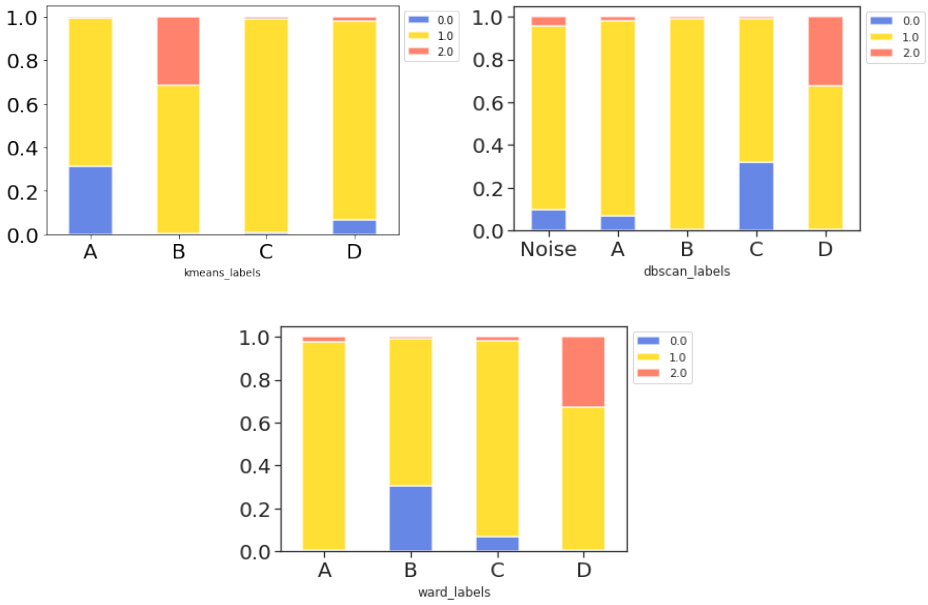

Figure 2.7: Distribution of Sentiment across the clusters obtained with K-Means, DBSCAN e Ward-linkage Agglomerative Hierarchical

Chapter 3

Data Classification

For the task of Data Classification we decided to try to predict our target variable Sentiment. The pre-processing for this task was very light, we removed from the dataset the Variance variable since, as shown in Chapter 1, is very closely related to Sentiment and it didn't seem fair to use a "continuous version" of our categorical target variable to predict it. So the dataset for all classification tasks we did was composed by the other 8 psycholinguistic features plus the logarithmic Web Corpus Frequency and Sentiment. We then splitted the dataset in two subset one for training the algorithms which was the 70% of the initial dataset, and one of test purposes which was the last 30%.

3.1 Classification by Decision Tree

For this algorithm we performed a Grid search to do a hyper parameter tuning and find out the best possible value for its parameters. The three parameter we performed the search on are *max_ depth*, *min_ samples_ split* and *min_ samples_ leaf* trying to maximize the F-Measure Score.
We decided to hyper tune towards higher F-Measures scores because since it's an harmonic mean between precision and recall, and it gave us a better idea on how many False Negatives and False Positive the model was generating. This was crucial since other measures, like accuracy, couldn't perform very well due to how unbalanced the Sentiment feature is. We performed two searches, one with the Gini index, and one with Entropy as the impurity measure, and we got a sightly better F-Measure score with Gini (0.833) than with Entropy (0.829). So at the end the best Model had this configuration:

Impurity Measure	**Max Depth**	**Min Samples Leaf**	**Min Samples Split**
Gini	None	5	17

With this model we then evaluate the model's prediction both on the train set and the test set, and after obtaining the confusion matrix we computed Precision, Recall, Accuracy and F-Measure score for both to evaluate the performance of the model. The results are reported in Table 3.1

Sentiment Class	Precision	Recall	F-Measure Score	Accuracy
Train Set				
Negative	0.74	0.67	0.70	0.92
Neutral	0.94	0.97	0.95	
Positive	0.82	0.61	0.70	
Test Set				
Negative	0.45	0.37	0.41	0.85
Neutral	0.91	0.92	0.91	
Positive	0.52	0.54	0.53	

Table 3.1

From this results we can see that the Accuracy between the train and the test set are very similar, this is because in both tests our Neutral class performed very well since it's the most common. That is why we didn't choose to hyper tune towards Accuracy. We can se that all of our measures decrease

drastically from the train set to the test set for the Negative and Positive classes, We've also plotted the ROC Curve (Figure 3.1), but since our variable, Sentiment, isn't binary, we had to plot three lines, one for each possible category, where the confusion matrix from which the ratios have been calculated was populated as that class versus the rest.

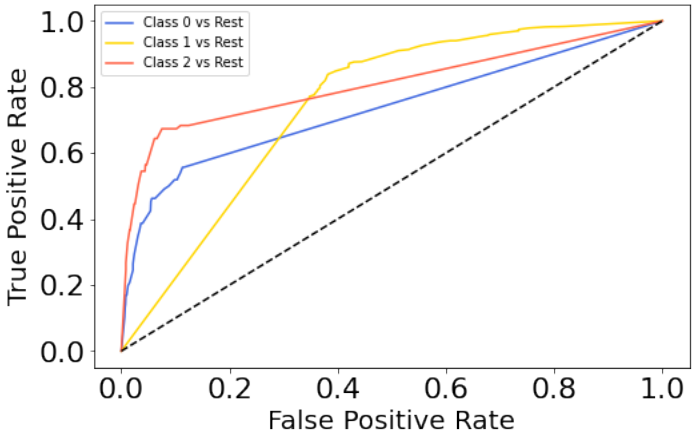

Figure 3.1: ROC Curve for the Decision Tree Classifier model

Last but not least, from the generated tree, in Figure 3.2, we can see that the most used features in the firsts steps is dominance. This isn't that surprising giving the high correlation between dominance and valence and the similarity between valence and Sentiment.

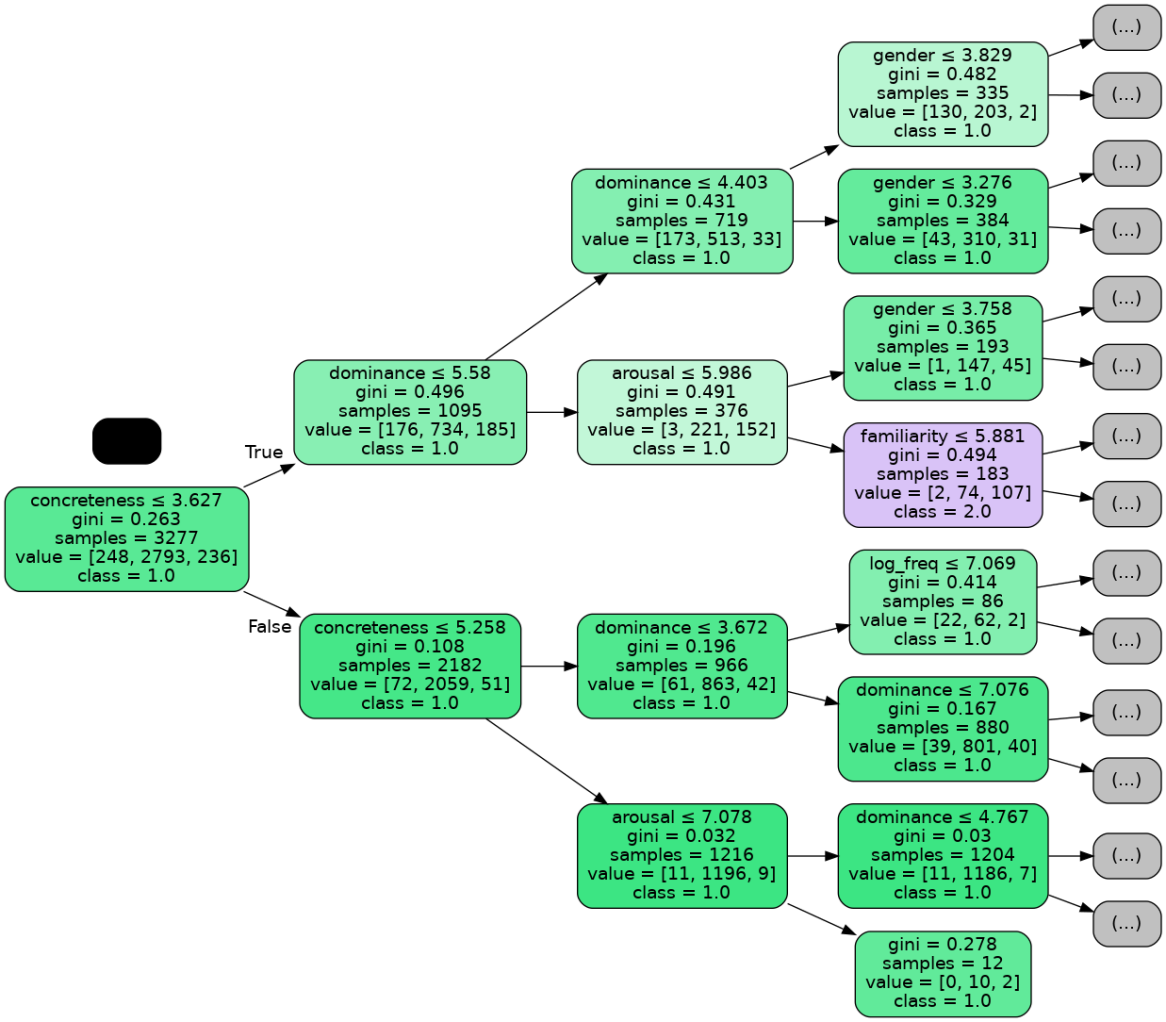

Figure 3.2: Decision Tree Classifier model tree output

3.1.1 Undersampling and Oversampling to fix class distribution

To improve the performance of our classifier, especially towards the Negative and Positive classes, we decided to try re-sampling techniques to balance the distribution of the classes across our dataset. We tried three different ver sampling algorithms (SMOTE, ADASYN and Random Over Sampler) and two different under sampling algorithms (Cluster Centroids and Condensed Nearest Neighbour). For each one of this algorithm we created a pipeline containing the re-sampling algorithm that, after re-sampling the train set, used it to train a Decision Tree Classifier. To have better result, a grid search was then performed onto the classifier of the pipeline to hyper tune its parameters to the new re-sampled dataset.
After that, the classifier model was tested using the original test set, which, for comparison purpose, always remained the same.
The results from each technique is reported in the Table 3.2

Sentiment Class	Precision	Recall	F-Measure Score	Accuracy
Original Decision Tree Classifier Test Set				
Negative	0.45	0.37	0.41	0.85
Neutral	0.91	0.92	0.91	
Positive	0.52	0.54	0.53	
SMOTE Decision Tree Classifier Test Set				
Negative	0.27	0.44	0.33	0.77
Neutral	0.91	0.82	0.86	
Positive	0.38	0.58	0.46	
ADASYN Decision Tree Classifier Test Set				
Negative	0.29	0.59	0.39	0.76
Neutral	0.93	0.78	0.85	
Positive	0.38	0.69	0.49	
RandomOverSampler Decision Tree Classifier Test Set				
Negative	0.27	0.29	0.28	0.81
Neutral	0.90	0.88	0.89	
Positive	0.46	0.50	0.48	
ClusterCentroids Decision Tree Classifier Test Set				
Negative	0.23	0.72	0.35	0.66
Neutral	0.95	0.64	0.76	
Positive	0.31	0.81	0.45	
CondensedNearestNeighbour Decision Tree Classifier Test Set				
Negative	0.23	0.79	0.35	0.70
Neutral	0.92	0.72	0.81	
Positive	0.42	0.42	0.42	

Table 3.2: Results from a Decision Tree Classifier trained with train set re-sampled with various different over sampling and under sampling techniques

To better visualize the effects of the techniques on the dataset we've also plotted the re-sampled dataset for each algorithm which can be seen in Figure 3.3.

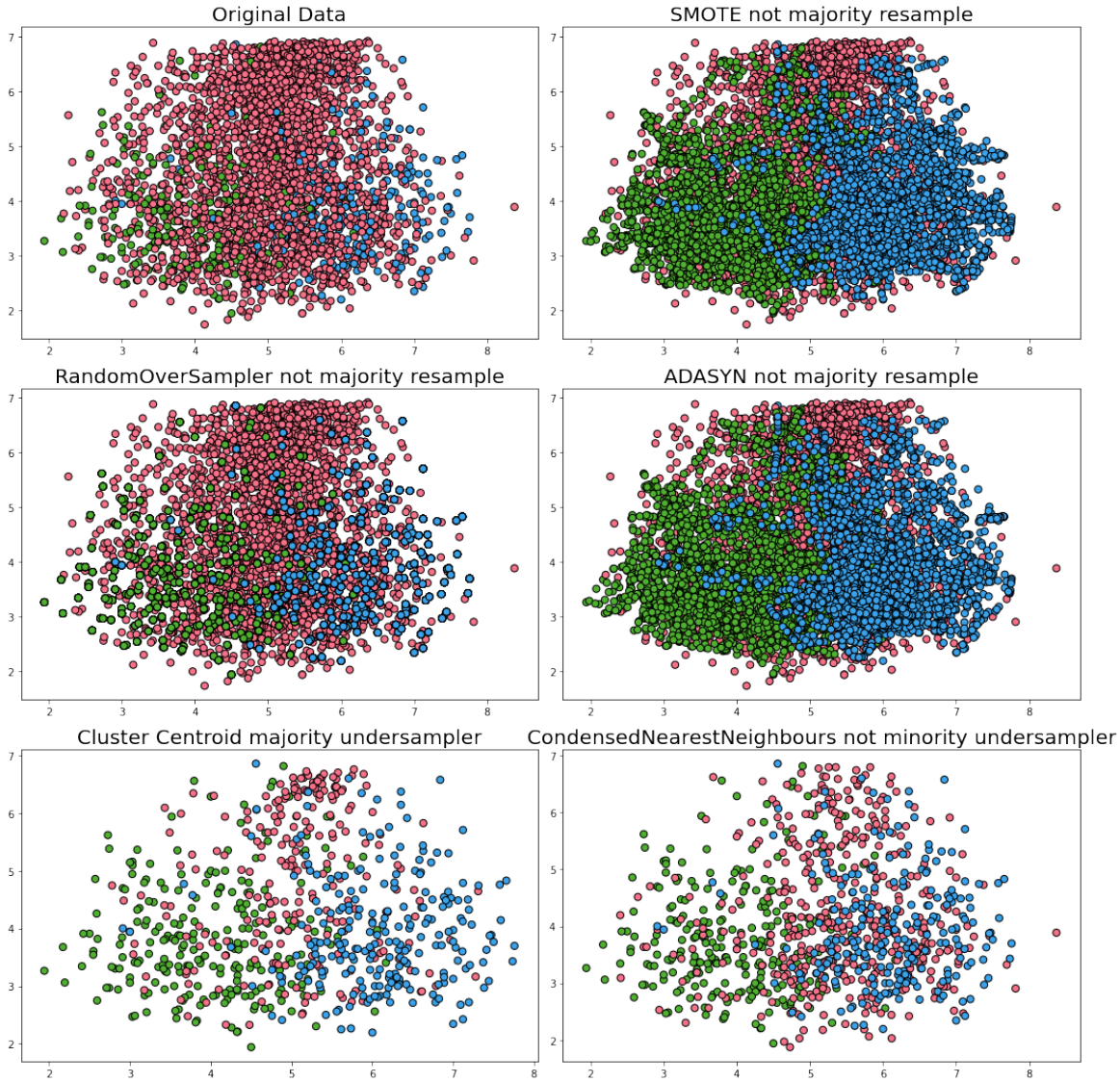

Figure 3.3: How the various re-sampling techniques affected the dataset

From the results we can say that none of the re-sampling techniques performed astonishingly better than the original Decision Tree Classifier. The SMOTE and ADASYN re-samplers have improved upon the recall measure for the Negative and Positive classes, this however came at the cost of a lower precision and accuracy. The same thing can be said for both of the under sampling techniques which also lowered considerably the recall measure for the Neutral class and both have an even lower accuracy.
In the end since most of the trade-offs that this re-sampling techniques came with, we decided to prefer our original model trained with the original dataset.

3.2 Classification with Random Forest

We also tested a Random Forest Classifier to compare it to our decision tree. This algorithm randomly selects observations and features to build several decision trees and then averages the results. The parameters in random forest are: number of estimators: the number of trees the algorithm builds before taking the maximum voting or taking the averages of predictions, max features: the maximum number of features random forest considers to split a node, min sample leaf: the minimum number of leafs required to split an internal node. Through a grid search we did an hyper-parameter tuning,

which resulted in the following configuration: min sample leaf = 2, min samples split = 12, and number of estimators = 200.

Sentiment Class	Precision	Recall	F-Measure Score	Accuracy
Train Set				
Negative	0.83	0.52	0.64	0.89
Neutral	0.92	0.98	0.95	
Positive	0.84	0.57	0.68	
Test Set				
Negative	0.51	0.33	0.40	0.87
Neutral	0.91	0.96	0.94	
Positive	0.66	0.49	0.55	

Table 3.3

From these results we can see that the accuracy between the train and the test is very similar, and the reason why is the same as for the Decision Tree Classifier: our Neutral class performed very well since it's the most common.
We can see that all of our measures decrease from the train set to the test set for the Negative and Positive classes, just like the previous analysis results, probably because the two classes are way under represented. We've also plotted the ROC Curve (Figure 3.2), and computed the AUC for it which is 0.88.

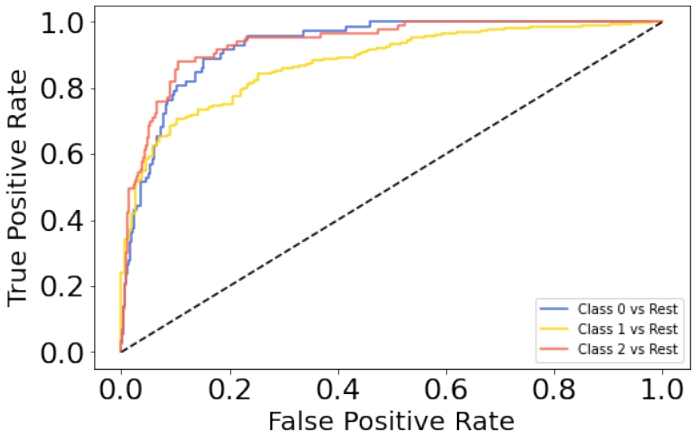

Figure 3.4: Roc Curve for the Random Forest Classifier model

3.3 Data Classification Discussion

First of all we can say that none of the models we've trained performed well enough during their test phases, both our Decision Tree Classifier (even those with re-sampled datasets) and our Random Forest Classifier had average results.
Comparing the two we can see that while the Decision Tree has better recall scores, the Random Forest has better precision and the F-measure ends up being pretty similar between the two. However the Random Forest Classifier has a better ROC Curve and AUC score.
We've also ran a cross validation for both of our models, and they obtained very similar average scores: 0.83 for the Decision Tree Classifier and 0.88 for the Random Forest Classifier. So, in the end we can say that our Random Forest Classifier model does, overall, a slightly better classification than the Decision Tree Classifier one.

Chapter 4

Pattern Mining

This section is focused on association rules/pattern mining, where the purpose was to find frequent item sets and interesting association rules in the in the Glasgow Norms dataset. Before starting to discover all the possible frequent item sets, in order to implement apriori algorithm it was necessary a pre-processing phase. We decided to keep all of the the psycholinguistic features except for valence plus the logarithmic Web Corpus frequency.

So we discretized the remaining variables, assigning labels to the bins, in order to have greater clarity in the results. Specifically we decided to divide the psycholinguistic feature arousal, dominance, familiarity, concreteness, imageability and semantic size, by following the definitions presented in Scott et al., 2018a, and by doing that we ended up to have three bins that reflected a low, medium and high score for the attribute in question.

Age of Acquisition was discretized by treating it for age groups of two years and Gender was also treated according to the definition on the guidelines: so depending on the score female, neuter or male. Finally for the logarithmic frequency we decided to divide it in five different degrees of frequency, expressed in the following way: "Very Uncommon", "Uncommon", "Moderately Common" , "Common", "Very Common". In the end we had a Dataframe like in Figure 4.1, which was then transformed into a list.

	sentiment	arousalBin	dominanceBin	concretenessBin	imageabilityBin	familiarityBin	aoaBin	semsizeBin	genderBin	log_freqBin
0	1.0	Moderate Arousal	Controlled	nor Abs. nor Concr.	Moderately Imageable	Unfamiliar	7.0	Very Big	Very Masculine	Very Uncommon
1	1.0	Very Unarousing	Controlled	Concrete	Moderately Imageable	Unfamiliar	5.0	Very Big	Very Feminine	Moderately Common
2	1.0	Very Unarousing	nor Cont. nor Dom.	Abstract	Hard to Imagine	Moderately Familiar	6.0	Very Small	Neuter	Very Uncommon
3	1.0	Moderate Arousal	Controlled	Abstract	Hard to Imagine	Unfamiliar	6.0	Very Big	Very Masculine	Very Uncommon
4	1.0	Very Unarousing	nor Cont. nor Dom.	Abstract	Hard to Imagine	Unfamiliar	6.0	Very Big	Very Masculine	Very Uncommon

Figure 4.1: Discretization end result

4.1 Frequent Pattern Extraction

In this subsection we will deal with the extraction of frequent patterns. The first thing we've looked at is the support distribution of the item sets, and what we observed is that for both closed and maximal item sets the number of item sets decreases very rapidly when the support increases (Figure 4.2).

We weren't able to get meaningful results for the "Positive" and "Negative" classes of the Sentiment attribute, since the class that collects the majority of the records is "Neutral".

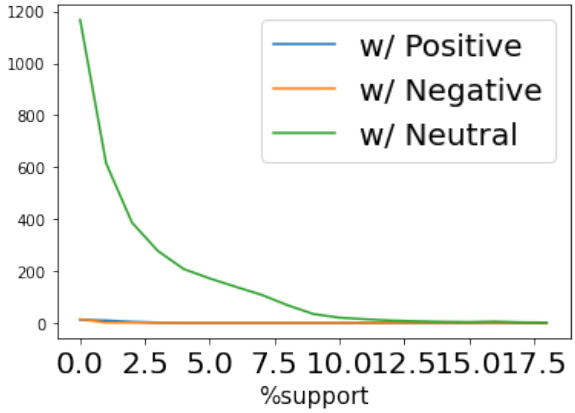

Figure 4.2: Plot of number of item sets based on support percentage

For the minimum support we chose 10% to select the most meaningful ones. In the Table 4.1 is it possible to observe the most frequent item sets found. These item sets reflects the correlations found in Data Understanding, for example the first one says that unfamiliar words are often words learned at a later age.

Frequent Item set	**Support**
('11-12', 'Unfamiliar', 'Neutral')	11%
('Very Common', 'Familiar', 'Neutral')	11%
('Very Uncommon', 'Unfamiliar', 'Neutral')	10%
('5-6', 'Concrete', 'Easy to Imagine')	10%
('5-6', 'Concrete', 'Easy to Imagine', 'Neutral')	10

Table 4.1: Frequent Item sets

4.1.1 Association Rules Extraction

To extract association rules, in addition to the minimum support value, it is also necessary establish the minimum confidence value for the Apriori algorithm. As shown in the plot (Figure 4.3) as the confidence value increases, the number of rules slowly decrease.

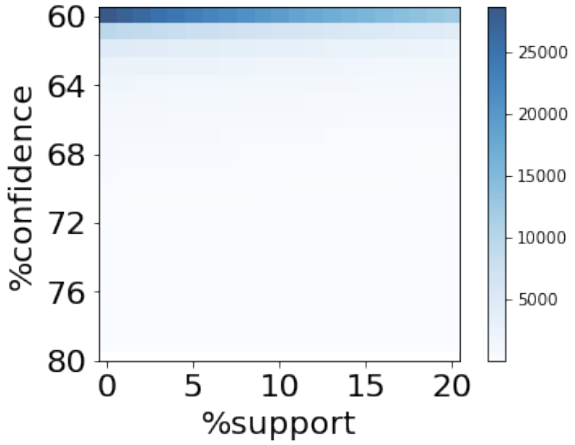

Figure 4.3: Heat Map Number of Rules for percentage of Confidence and Support

In the end, we decided a minimum confidence percentage equal to 60%, and then we ordered the associations rule by the lift value. As it is possible to observe in Table 4.2, the rules with the most informative associations are expected: the words voted as unfamiliar are those that were learned at a later age and hard to imagine, the most easy to imagine words are the concrete ones and uncommon words are also unfamiliar.

Association Rule	Support	Confidence	Lift
Hard to Imagine, Unfamiliar, Neutral → 11-12	6%	60%	3.82
11-12, Neutral → Unfamiliar	10%	78&	2.31
Concrete, Neutral → Easy to Imagine	11%	20%	2.07
Very Uncommon, Neutral → Unfamiliar	13%	96%	2.91
5-6, Concrete, Neutral → Easy to Imagine	10%	80%	2.71

Table 4.2: Association Rules

4.2 Target Variable Prediction

To test our association rules we decided to see how effecting some of this rules are at predicting our target variable, sentiment. We decided then to compare the results with our best classification model. To do that, we selected the two best association rules taking lift and confidence values into consideration. As shown in Table 4.3 their information value is very low, the lift value is low and their confidence is very high.
We've also only found rules for the Neutral class of sentiment, this was expected since the other two class are not very frequent and therefore have probably been cut out by their lower support value.

Association Rule	Support	Confidence	Lift
5-6, Concrete, Easy to Imagine → Neutral	10%	99%	1.16
Concrete, nor Cont. nor Dom., Easy to Imagine → Neutral	14%	99%	1.16

Table 4.3: Selected rules for target variable

We decided to compare the results from a "prediction" made with the second rule, which ended up being the best of the two, with the results from our Random Forest Classification model. Since the rule only predicts Negative values, we've compared it only to the classification model ability to predict neutral values. From the table 4.4 we can see that the recall and F1 scores are all far worse than our classification model. The only score that is relatively high is precision and that could be explained by the high imbalanced class distribution of sentiment.

Model	Precision	Recall	F-Measure Score
Association Rule	0.15	0.78	0.25
Random Forest Classifier Model	0.91	0.96	0.94

Table 4.4: Comparison between prediction results of Random Forest and extracted rules

4.3 Final considerations

The results that emerged from the pattern mining analysis didn't return particularly interesting results. Most of the rules we've got out are a simple "transcription" of the correlation we found out during the Data Understanding phase of the project. Also, the most frequent rules didn't have our target variable in it, and when looking at rules with sentiment in it, we've only found rules for the neutral class. When we tried to apply this rules at a "prediction" task, we've had pretty bad results.

Bibliography

Altarriba, Jeanette, Lisa Bauer, and Carlos Benvenuto (Dec. 1999). "Concreteness, context availability, and imageability ratings and word associations for abstract, concrete, and emotion words". In: *Behavior research methods, instruments, & computers : a journal of the Psychonomic Society, Inc* 31, pp. 578–602. DOI: 10.3758/BF03200738.

Gilhooly, Ken and Robert Logie (July 1980). "Age-of-acquisition, imagery, concreteness, familiarity, and ambiguity measures for 1,944 words". In: *Behavior research methods, instruments, & computers: a journal of the Psychonomic Society, Inc* 12, pp. 395–427. DOI: 10.3758/BF03201693.

Lenci, Alessandro, Simonetta Montemagni, and Vito Pirrelli (2005). *Testo e Computer. Elementi di Linguistica Computazionale*. corso Vittorio Emanuele II, 229, 00186, Roma: Carrocci Editore.

Qi, Peng, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning (2020). "Stanza: A Python Natural Language Processing Toolkit for Many Human Languages". In: *"Association for Computational Linguistics (ACL) System Demonstrations.*

Richardson, John (May 2007). "Concreteness and Imageability". In: *Quarterly Journal of Experimental Psychology - QUART J EXP PSYCHOL* 27, pp. 235–249. DOI: 10.1080/14640747508400483.

Scott, Graham, Anne Keitel, Marc Becirspahic, Bo Yao, and Sara Sereno (2018a). "Supplementary Materials - The Glasgow Norms". In: DOI: 10.3758/s13428-018-1099-3.

— (2018b). "The Glasgow Norms: Ratings of 5,500 words on nine scales". In: *Behavior Research Methods* 51. DOI: 10.3758/s13428-018-1099-3.

Stadthagen-Gonzalez, Hans and Colin Davis (Nov. 2006). "The Bristol norms for age of acquisition, imageability, and familiarity". In: *Behavior research methods* 38, pp. 598–605. DOI: 10.3758/BF03193891.