Logo

Description automatically generated

University of Pisa

Department of Informatics

Master’s Degree Program of Data Science and Business Informatics

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

DATA MINING: FOUNDATIONS

A.A. 2021-22

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Knowledge Discovery from Dataset

**Glasgow Norms**

*Mariami Narchemashvili 634789*

*Umberto Lamia 517110*

*Stefania Occhiuto*

*Daniel Vigilio*

[1 DATA UNDERSTANDING AND PREPARATION 3](#_Toc92324677)

[1.1 DATA SEMANTICS 3](#_Toc92324678)

[1.2 DISTRIBUTION OF THE VARIABLES AND STATISTICS 4](#_Toc92324679)

[1.3 ASSESING DATA QUALITY 5](#_Toc92324680)

[1.4 VARIABLE TRANSFORMATIONS 6](#_Toc92324681)

[1.5 PAIRWISE CORRELATIONS AND EVENTUAL ELIMINATION OF VARIABLES 7](#_Toc92324682)

[2 CLUSTERING 8](#_Toc92324683)

[2.1 CLUSTER ANALYSIS BY K-MEANS 8](#_Toc92324684)

[2.2 ANALYSIS BY DENSITY-BASED CLUSTERING 11](#_Toc92324685)

[2.3 ANALYSIS BY HIERARCHICAL CLUSTERING 11](#_Toc92324686)

[2.4 FINAL DISCUSSION 12](#_Toc92324687)

[3 CLASSIFICATION 12](#_Toc92324688)

[3.1 CLASSIFICATION BY DESICION TREE 12](#_Toc92324689)

[3.1.2 PERFORMANCE EVALUATION OF THE ALGORITHM 14](#_Toc92324690)

[3.2 CLASSIFICATION BY OTHER ALGORITHMS 15](#_Toc92324691)

[3.2.1 KNN CLASSIFIER 15](#_Toc92324692)

[3.2.2 RANDOM FOREST CLASSIFIER 15](#_Toc92324693)

[3.3 FINAL DISCUSSION 16](#_Toc92324694)

[4 PATTERN MINING 16](#_Toc92324695)

[4.1 FREQUENT PATTERN EXTRACTION 17](#_Toc92324696)

[4.2 FREQUENT PATTERN DISCUSSION 17](#_Toc92324697)

[4.3 ASSOCIATION RULES EXTRACTION 19](#_Toc92324698)

[4.4 ASSOCIATION RULES DISCUSSION 19](#_Toc92324699)

[4.5 FINAL DISCUSSION 20](#_Toc92324700)

The following report is analyzing two datasets, one representing 4682 English words and the other one specifying 379 ambiguous words from the first dataset which have several meanings. In total both sets are composed of 5553 English words, which are characterized by the following psycholinguistic dimensions: length of the words, arousal, valence, dominance, concreteness, imageability, familiarity, age of acquisition, semantic size, gender, and web corpus frequency.

To transform the raw input data into an appropriate format for subsequent analysis we go through following steps.

# **1 DATA UNDERSTANDING AND PREPARATION**

## **1.1 DATA SEMANTICS**

This section of the report represents two tables for both datasets – (1) Words Glasgow & (2) Words Polysemy which are provided as an input data. We can say that both datasets are high-dimensional as the number of attributes is quite high.

Through the variable “Name” we indicate the name of the attribute/variable we are considering. In total we have 13 attributes (in dataset (1)), with the following table we demonstrate each feature with its description, type, and domain.

The first three dimensions (arousal, valence, dominance) are used to measure the emotional and psycological empact of a word. All the other variables indicate the grade of knowing the word and the use of it.

|  |  |  |  |
| --- | --- | --- | --- |
| **NAME** | **Description** | **TYPE** | **DOMAIN** |
| Word | English words | Categorical - Nominal (string) | 4683 (number of records) |
| Length | Word length | Numerical - Discrete (integer) | [2; 16] |
| Arousal  (AROU) | Measure of excitement (excitement, calmness) | Numerical - Contininuous (float) | [2.057, 8.177] |
| Valence  (VAL) | Measure of value or worth (positive, negative) | Numerical - Contininuous (float) | [1.03, 8.647] |
| Dominance  (DOM) | Measure of the degree of control  (dominant, controlled) | Numerical - Contininuous (float) | [1.941, 8.371] |
| Concreteness  (CNC) | Measure of how concrete or abstract something is (concrete, abstract) | Numerical - Contininuous (float) | [1.636, 6.938] |
| Imageability (IMAG) | Measure of generating a mental image of something (imageable, unimageable) | Numerical - Contininuous (float) | [1.737, 6.941] |
| Familiarity  (FAM) | Measure of how familiar a word is  (familiar, unfamiliar) | Numerical - Contininuous (float) | [1.647, 6.939] |
| Age of acquisition  (AOA) | Measure of the age at which a word was initially acquired | Numerical - Contininuous (float) | [1.219, 6.971] |
| Semsize  (SIZE) | Measure of magnitude  (big, small) | Numerical - Contininuous (float) | [1.375, 6.912] |
| Gender  (GEND) | Measure of a word considered to be associated with male or female behavior  (masculine, feminine) | Numerical - Contininuous (float) | [1.0, 6.971] |
| Polysemy | Measure of semantically ambiguous words which convey multiple meanings  (homographs) | Categorical - Binary (integer) | {0,1} |
| Web corpus frequency  (WCF) | Measure of frequency of a word in Google Newspapers Corpus | Numerical - Discrete (integer) | [12770, 2022459848] |

Out of all features we distinguish 9 variables as 9 dimensions of each record. Those nine psycholinguistics variables are continuous: Arousal, Valence, Dominance, Concreteness, Imageability, Familiarity, Age of Acquisition, Size, and Gender.

The following table represents dataset (2) – ‘Words Polysemy’, where in total there are 11 attributes (columns).

|  |  |  |  |
| --- | --- | --- | --- |
| **NAME** | **Description** | **TYPE** | **DOMAIN** |
| Word | English words | Categorical - Nominal  (string) | 872 (number of records) |
| Length | Word length | Numerical - Discrete  (integer) | [2; 16] |
| Arousal  (AROU) | Measure of excitement (excitement, calmness) | Numerical - Contininuous (float) | [2.057, 8.177] |
| Valence  (VAL) | Measure of value or worth (positive, negative) | Numerical - Contininuous (float) | [1.03, 8.647] |
| Dominance  (DOM) | Measure of the degree of control  (dominant, controlled) | Numerical - Contininuous (float) | [1.941, 8.371] |
| Concreteness  (CNC) | Measure of how concrete or abstract something is (concrete, abstract) | Numerical - Contininuous (float) | [1.636, 6.938] |
| Imageability (IMAG) | Measure of generating a mental image of something (imageable, unimageable) | Numerical - Contininuous (float) | [1.737, 6.941] |
| Familiarity  (FAM) | Measure of how familiar a word is  (familiar, unfamiliar) | Numerical - Contininuous (float) | [1.647, 6.939] |
| Age of acquisition  (AOA) | Measure of the age at which a word was initially acquired | Numerical - Contininuous (float) | [1.219, 6.971] |
| Semsize  (SIZE) | Measure of magnitude  (big, small) | Numerical - Contininuous (float) | [1.375, 6.912] |
| Gender  (GEND) | Measure of a word considered to be associated with male or female behavior  (masculine, feminine) | Numerical - Contininuous (float) | [1.0, 6.971] |

## **1.2 DISTRIBUTION OF THE VARIABLES AND STATISTICS**

Chart

Description automatically generated

In the following section the distribution of variables are represented with the help of different kinds of vizualization tools.

The first variable represented is ‘Polysemy’, that shows the ambiguouty of the word and that we consider as a target or also known as dependent variable. It is represented as a boolean variable giving two values 0 and 1 (word has one meaning or several meanings respectively). It has been observed that 4303 words has polymesy equal to 0, and 379 equal to 1.

Chart, pie chart

Description automatically generated

Out of ambiguous words there are 289 words with two different meanings; 69 – three; 19 – four; 2 – five. Figure 1.2.1 and 1.2.2 shows the distribution of words in dataset 1 with respect to ‘Polysemy’ attribute. From the figures presented, it can be assumed that the dataset 1 is quite unbalanced with respect to the target variable.

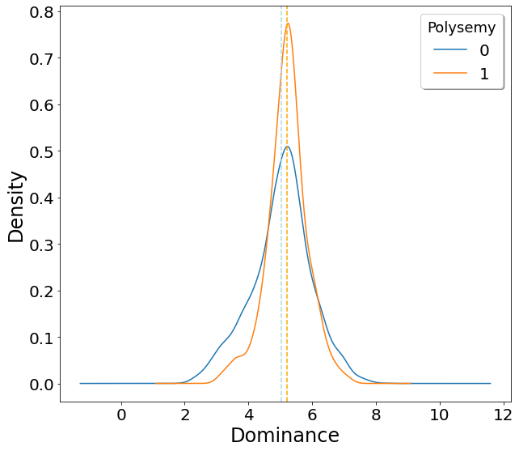
The following are the visualisation of density plots in order to learn the statistics and distribution of variables with respect to our target variable ‘Polysemy’. The dotted lines represent the mean values in both ambiguous and not ambiguous words’ sets (respectively same colors).

Histogram

Description automatically generated with medium confidenceChart, histogram

Description automatically generatedChart, histogram

Description automatically generated

Chart

Description automatically generatedChart, line chart

Description automatically generated

Chart

Description automatically generatedChart

Description automatically generatedHistogram

Description automatically generated with medium confidence

A picture containing histogram

Description automatically generatedA picture containing text

Description automatically generated

According to the Density plots, it is quite obvious that some of the attributes, such as: Concreteness, Imageability, Web Corpus Frequency are not well-disctributed across the dataset 1.

## **1.3 ASSESING DATA QUALITY**

The following section studies and evaluates data quality. In order to asses the quality of the dataset it explores and handles missing values, outliers, and any other semantic errors or inconsistencies.

**Dublicate Data**

First, we investigated dublicate records and we found out that in our data we do not face the issues regarding to it.

**Syntatic Accuracy**

Another thing that we checked was accuracy of the variable ‘length’, making sure the closeness of measurement to the true value. We counted actual length of the words’ strings. We compared string lengths to our attribute ‘length’ in order to make sure that the data represented with this attribute does not contain any inaccuracies. finally we found zero errors in these variable.

**Semantic Accuracy**

After that, we checked the number of ambiguous (polysemous) words. For this part we used both datasets. In dataset (2) we separated the description part from the actual word and counted the unique values. Finally, we compared the words from dataset (1) with the polysemy value equal to 1 and unique words from the dataset (2). On the first phase we faced the difference between the total number of words. However, we found out that the difference was caused only because of the word ‘apple’ which was represented in the second dataset in two differenc ways (with capital letter representing the brand - Apple).

With the help of previous checks of nominal, discrete and binary variables, we can say that in the dataset (1) we do not face any major semantic errors.

As for continuos attributes we checked for missing values and outliers, which is represented in the following sections.

**Missing values**

For this part we checked all the continuous attributes and detected some missing values. We found them in only one attribute which represents ‘web corpus frequency’. The number of those missing values was 14, which is not significant with respect to the number of all records. As there are several strategies (elimination or substitution) to tolerate poor data quality caused by missing values, in the following sections we show how we handled them.

**Outliers**

The outliers are anomalous objects that have different characteristics from all the others in the dataset. They have an unusal value of an attribute from the usual values of that attribute. In order to find those anomalous values we applied several methods.

In order to detect outliers in our dataset we used boxplots and z-score normalization for all the attributes. In total we found 325 outliers with respect to all the attributes. Only five of them were the same as missing values from web corpus frequency attribute. We decided to remove all of the outliers from the original dataset for further improvements.

## **1.4 VARIABLE TRANSFORMATIONS**

This section represents some transformations done on dataset 1 in order to represent data in a scale considered more suitable.

Graphical user interface

Description automatically generated with low confidenceChart, histogram

Description automatically generatedFirst transformation that was applied is logarithm transform of the attribute ‘web corpus frequency’ in order to stabilize the variance. As this variable is represented as high continuous numbers and is poorly distributed, it was not convinient to analyze it with other attributes.

In order to visualize data with histograms we used Struge’s rule to define the number of bins (k = dlog2(n)+1e, where n is the sample size).

Chart, histogram

Description automatically generatedAfter logarithm transformation and elimination of outliers, we filled remained missing values with mean. As it is obvious from the figure 1.4.2 that there is not any significant difference in the two ways of substitution (with mean or with median), we made the decision easily.

Chart, histogram

Description automatically generatedWe tried other types of transformations, such as square root and reciprocal. We applied square root transformation to all the attributes seperately and found important significance in the attribute arousal, as it improves the distribution of variable. However, when we applied reciprocal trasnformation to the attributes, it did not seem helpful for further improvements so we decided to not use it.

## **1.5 PAIRWISE CORRELATIONS AND EVENTUAL ELIMINATION OF VARIABLES**

It is possible to compute measures of correlation between attributes to confirm expected dependencies or to discover unexpected correlations between attributes.

For this part of the report we used Pearson’s Correlation Coefficient which is a measure for a linear relationship between two numerical attributes. Additionally, we used Spearman Rank Correlation Coefficient which intend to measure monotonous correlation between attributes where the function does not have to be linear.

Pearson’s correlation coefficient is widely used and it measures the strength of the linear relationship between normally distributed variables. When the variables are not normally distributed or the relationship between the variables is not linear, it may be more recommended to use the Spearman rank correlation method.

After applying both correlation methods to our dataset, we found just a little difference between the results of these two. It was quite expected because after the elimination of semantic innacuracies and transformation of variables, the data became much well-distributed.

Chart

Description automatically generated

Chart, bar chart

Description automatically generatedIn the following graph we only represent the attribute pairs which were highly correlated to each other.

Pairs of highly correlated attributes are the following: [CNC & IMAG] r = 0.91; the more concrete a word is, the easier it is to imagine; [VAL & DOM] r = 0.69; the more positive a word is, the more it provokes feelings of dominance; [FAM & AOA] r = – 0.67; the more familiar a word is, the earlier that word was learned; [SIZE & AROU] r = 0.51; the bigger the object or concept is to which a word refers, the more arousing the word is; [FAM & WCF];

We decided to eliminate one from each paired attribute (the ones of which variable distribution was worse).

# **2 CLUSTERING**

Clustering exploits similarities between the data to be analyzed, similarities that can be of various nature but which are essentially a distance between the dataset points. Different algotirhms can be used for clustering analysis, but the following sections will examine K-Means, Density-based and Hierarchical clustering. Before applying any of those, it is required to go through some preprocessing steps.

**Choice of Attributes**

For the clustering algorithms the categorical and discrete attributes (such as ‘word’, ‘polysemy’, ‘length’) are dropped from the set and the further analysis are carried out with other variables.To increase the performance of the clustering algorithms we need to normalize the ranges of attributes. In order to complete this step we can use multiple kinds of scalers, but in our case we will discuss only the most widely used ones (Standard, Minmax, Robust). **Standard Scaler -** Scaled between std ranges;

**Minmax Scaler -** Scaled between the range [0, 1]; **Robust Scaler -** Works good on outliers and consideres interquartile ranges instead of std. Figure 2.1.1 represents applied Standard Scaler on chosen attributes in dataset. Even though all three of the scalers have been examined, the following analysis have been carried out with standard scaled attributes.

A picture containing box and whisker chart

Description automatically generated

## **2.1 CLUSTER ANALYSIS BY K-MEANS**

K-means is one of the most widespread and best performing clustering algorithms. Its is prototype-based, partitional clustering technique that attemps to find user-specified number of clusters (k), which are repsresented by their centroids.

Centroids are mean of the points of clusters, not real ones. Initial centroids are often chosen randomly. In this case we chose 3 as an initial number for clusters. In order to find optimal number of clusters there are two different approaches.

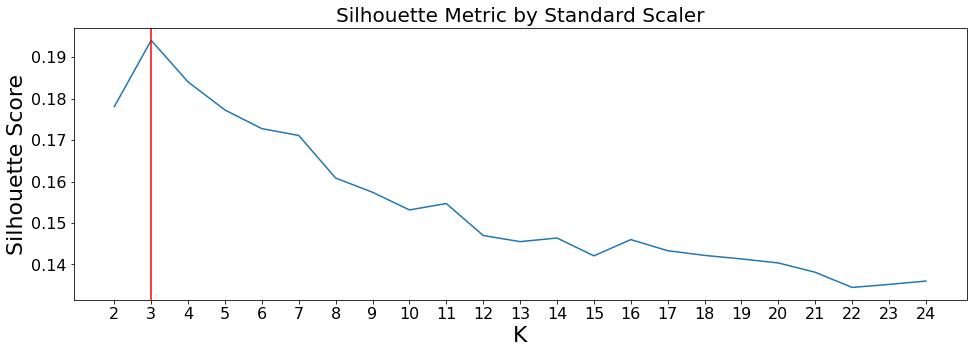
**Elbow Method** helps to plot the WCSS (Within Clusters Summed Squares) values ​​and selects the point where the parameter value falls more than the previous value. For each point, the Sum of Squared Error (SSE) is the distance to the nearest cluster. To get SSE, we square these errors and sum them. Finally we choose the optimal number of clusters considering the lowest error with respect to lower number of clusters. The lower the SSE the better.

Another method that can help to choose the number of clusters is **Silhouette method**. In this procedure the silhouette coefficient is plotted and the maximum value is selected. The higher the Silhouette Coefficient the better.

We run the algorithm in the range of: , where k is the number of clusters, and n is the the number of rows we have in our dataset. So we computed k’s highest value which could be .

Computing SSE and Silhouette with different scaling methods, we got the following results:

A picture containing shape

Description automatically generated

Finally, SSE was quite high considering the high-dimensionality of the data. For clustering analysis we get optimal SSE with respect to clusters’ number equal to 6 and Silhouette Coefficient with respect to clusters’ number equal to 3.

The following table is a demonstration of Silhouette Score and SSE with respect to number of clusters.

Table

Description automatically generated

Chart, scatter chart

Description automatically generated

A picture containing window, building, green, different

Description automatically generatedA picture containing colorful, different, colors, painted

Description automatically generated

Table

Description automatically generated

The records were almost equally distributed in each cluster, which states that the k-means algorithm worked well on the dataset 1 and separated the records in a balanced way.

Chart, line chart

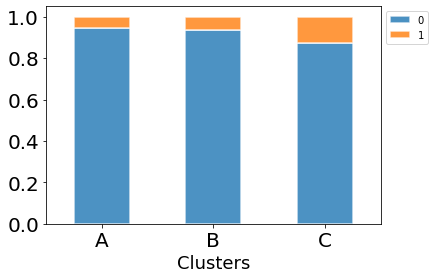
Description automatically generated

**Paraller-Coordinate Plot**

In order to see the seperation of mean values of each cluster with respect to each attribute, we visualize clustered data with paraller-coordinate plots. Mostly all the attributes’ mean values are well-separated in the clusters, except the one for ‘Imageability’.

**Barchart**

In order to see the distribution of variables with respect to clusters we visualized data with barcharts.

Chart

Description automatically generatedChart, bar chart

Description automatically generated

## **2.2 ANALYSIS BY DENSITY-BASED CLUSTERING**

In the analysis by density-based clustering number of clusters is not defined by us, but two other main parameters which are ε (eps), which defines the radius of neighbourhood around a point and min\_samples which defines the minimum number of neigbhours of the point within the radius. With DBscan we can see the actual clusters and the noise points.

Line chart

Description automatically generated with medium confidenceWe used this clustering method to compare detected noise points to the outliers explored previously by boxplots. For our dataset, as shown in the figure 2.7, the optimal value of eps has been calculated and found in the interval [0.30, 0.35].

A picture containing window, building, crossword puzzle

Description automatically generatedThe algorithm with different values of the parameters have been examined for several times and collected information about the different clusters. The optimal number of eps and min\_samples have been achieved after several trials.

Figure 2.33 presents 2 clusters one much bigger than the other and the number of noise points. Finaly we decided to keep the clustering obtained with parameters min\_samples = 5 and Epsilon = 0.30.

DISCUSS n. of noise points w.r.t. eps

## **2.3 ANALYSIS BY HIERARCHICAL CLUSTERING**

Graphical user interface, text

Description automatically generatedHierarchical Clustering method builds clusters step by step. The final result is visualized as a dendrogram. The following analysis has been conducted on an agglomerative clustering method.

There are four different methods to visualize dendogram: Average (good against noise and outliers, bad with clusters of different number of elements); Complete; Single; and Ward (good against noise and outliers, good for spherical or rounded clusters). Even though we examined all of them, the best visualizaiton was achieved by the ‘single’ method. However, the figure 2.3.1. demonstrates that it does not apply on the dataset well in terms of clustering. Therefore, it has been decided to chose the highest after the ‘single’ method which is ‘average’. Figure 2.3.4. is the demonstration of best performing hierarchical method on our dataset.

The following is the visualization of dendrogram using ‘average’ method.

A picture containing bar chart

Description automatically generated

## **2.4 FINAL DISCUSSION**

Table

Description automatically generatedIn order to evaluate the best clustering approach, we decided to consider the silhouette score of the all clustering algorithms applied in the previous sections. Table 2.4 represents the silhouette score for K-means, DBScan and Hierarchical algorithms. The values of the parameters used for each algorithm are: In K-Means - K = 3 used; In DBScan - eps = 0.30 and min\_samples = 5 used; In Hierarchical – ‘average method’ - Euclidean metric with 3 clusters;

To conclude, given the results obtained from the previous sections, it is feasible that the clustering method that fits best our dataset is Dbscan.

# **3 CLASSIFICATION**

Data for classification task consists of collection of instances/records and each of them is characterized by the tuple (x, y). X – representing attribute/predictor and Y- representing class/response.

Classification model serves two important roles: (1) It is used as a **Predictive Model** to classify unbalanced instances; (2) It is used as a **Descriptive Model** to identify characteristics that distinguish instances from different classes. Below are the steps that will be carried out in the following sections.

* Learning algorithm: systematic approach to learn classification model on training set.
* Induction: by using learning algorithm, building classification model.
* Deduction: applying classification model on test set (unseen test instances).

**DATA PREPARATION**

**Choice Of The Attributes**

In order to start working on classification we have to do some preprocessing, such as, dealing with missing values (section 1.3) and removing/dropping useless variables (section 1.5).

Also, it is obvious that we dropped two categorical attributes, ‘word’ which is useless in terms of classification, as it provides unique value for each record, and ‘polysemy’ which is our target attribute. Final choice of the attributes is the following: Length, Arousal, Valence, Imageability, Age of Acquisition, Size, Gender, and Web Corpus Frequency.

Additionally, we have to prepare the dataset, meaning separating training and test sets from the original one. As the dataset is not balanced with respect to the target attribute (‘polysemy’), we use random oversampling, in order to have equal number of records for both, positive and negative classes (respectively polysemy = 1 and polysemy = 0).

After oversampling, we split data into training and test sets, and also we decided to split training set into train (D.tr) and validation (D.val) sets. D.tr takes 2/3 of a training set and we use it to build model, whereas we use D.val to estimate generalization error.

Even thought we use oversampling technique to balance the data, in the following sections we represent the results obtained from both balanced and unbalanced data.

## **3.1 CLASSIFICATION BY DESICION TREE**

Desicion tree is a clasiffication technique which is structured hierarchically representing organized series of questions and their possible answers.

**Identify The Best Parameter Configuration**

In order to identify the best parameter configuration before modeling decision tree classifier, we decided to apply parameter tuning. We used Gridsearch and RandomizedSearch methods with an f1-scoring criterion, as f1-score describes the harmonic mean of precision and recall. Parameters are the following:

* **criterion**: The function to measure the quality of a split. Available: GINI, Entropy.
* **max\_depth**: The maximum depth of the tree. Range: None + (2, 20). If None, then nodes are expanded until all leaves are pure or until all leaves contain less than min\_samples\_split samples.
* **min\_samples\_split**: The minimum number of samples required to split an internal node. Range: {2, 5, 10, 20, 30, 50, 100}.
* **min\_samples\_leaf**: The minimum number of samples required to be at a leaf node. Range: {1, 5, 10, 20, 30, 50, 100}.

The following table represents the suggested criterions to use for Decision Tree classifier on the balanced data.

Table

Description automatically generatedThe best combination of maximum depth, minimum sample split and minimum sample leaf has been chosen, in order to avoid underfitting or overfitting in classification model. For Decision tree we used best estimation results provided by Randomized Search.

After that, we started to build the Decision Tree model.

A picture containing icon

Description automatically generated

We measured feature importance for splitting. The following table demonstrates feature importance in an ordered way.

In this graphical representation we can see which attributes play the most important role for the further analysis. We can also identify the most important features in the decision tree provided below built on balanced data.

Diagram

Description automatically generated

Figure 3.1.3 represents the first levels of the Decision Tree considering the previous features and parameters. Each node of the tree represents: test condition attribute, the index of Entropy (impurity measure of target class distribution), total number of samples and the numbers of each target instances (positive and negative) distributed for the child nodes. For example, the first node’s enropy measure is equal to 1, which means that the polysemy values (0, 1) are equally distributed for the child nodes. In the next nodes, we can see that criterion ‘class’ represents the boolean values correspondingly to target attribute, the one which is superior to the other considering the number of records.

Diagram

Description automatically generated

For the unbalanced data we repeated same steps, using the same algorithms that were used to select the best hyperparameters, measuring the importance of the features, and finally implementing Decision Tree model shown on (fig. ). The main difference that should be noticed is the value of impurity mesure, which is lower than in the previous case, as the data is unbalanced with respect to target class, and therefore there is less-balanced distribution in the child nodes.

### **3.1.2 PERFORMANCE EVALUATION OF THE ALGORITHM**

For the evaluation phase we discuss methods for estimating its generalization performance on unseen instances outside of D.tr (train set).

Table

Description automatically generated

In the following table we represent cost-sensitive measures such as: Precision, Recall, F-measure, and Accuracy, on both balanced and unbalanced data, in order to make a comparison between them.

More in detail, to establish which is the model that performs better, it is necessary to compare the evaluating measures as Accuracy, F1 Score, Precision and Recall of both sets. Looking at the figure 4.2, the model built on the balanced train set feels to perform better. It shows in the validation set an accuracy value (0.92) greater than the classifier built on the unbalanced training set (0.90). Going more into deep details, the difference between the accuracy of these two sets is quite small, and we can say that even the unbalanced model is quite perfect in predicting polysemy values. However, it has poor score in classifying ambiguous words, the recall value is really low which is not acceptable.

Another evidence, concerning the preferability of a balanced model is given by the Roc curves (figures 4.4

and 4.5). Balanced model presents a higher level of sensibility at a higher level of specificity. Due to this evidence, it was decided to continue the analysis and the comparison with the other models using the balanced classifier.

A picture containing line chart

Description automatically generated**A picture containing chart

Description automatically generated**

A picture containing text, screenshot, electronics

Description automatically generatedWe used confusion matrix for the performance evaluation, where we focus on the predictive capability of a model. In the following graphs we present confusion matrix done on D.val (validation set). The values provided are normalized. 0.97 represents the normalized number of True Positive instances, and 0.86 represents the normalized number of True Negative instances.

## **3.2 CLASSIFICATION BY OTHER ALGORITHMS**

### **3.2.1 KNN CLASSIFIER**

A picture containing chart

Description automatically generatedKNN classifier is one of the techniques that are known as lazy learner algorithms and which are different from Decision Tree classifier tehcniques. The algorithm is the following, given a test instance, we compute its proximity to the training instances according to one of the proximity measures. In order to find the best number of neighbours, as initial step we took the K range of (1, 40) and used accuracy scoring. The graph 3.2.1.1 represents computed optimal number of K with respect to misclassification error rate.

Table

Description automatically generated

According to the results, the red dotted line shows the chosen number of K (neighbours), and the relative performance are summarized in the table 4.4 below.

A picture containing text, electronics, screenshot

Description automatically generated

Provided confusion matrix represents the results of KNN classifier done on D.val (validation set). All the results are normalized. TP = 0.99, FP = 0.24, TN = 0.76, FN = 0.006.

### **3.2.2 RANDOM FOREST CLASSIFIER**

A picture containing text, electronics, screenshot

Description automatically generatedThe Random Forest is an ensemble algorithm which was used in this case to combine different decision tree models. Randomized Search technique was used to determine the best parameters to be used into the implementation. The parameters are: min\_samples\_split: 10, min\_samples\_leaf: 5, max\_depth: 46, criterion: 'entropy', with a mean validation score of 0.93 (std: 0.010). The evaluation was made computing the accuracy of training and validation sets as before. The figure 1.2 represents the confusion matrix on validation set (D.tr) where we can see that the classifier predicted well 98% of ambiguous records as polysemy = 1 (True Positive) and it misclassified 8% as False Positive; on the other hand, it well classified 92% of non-ambiguous records as Polysemy = 0 (True Negative) and misclassified only 1.7% as False Negative.

Table

Description automatically generated

## **3.3 FINAL DISCUSSION**

In this section of the report the evaluation is done on all three classifiers presented above (Decision Tree, KNN, Random Forest). A comparison was made for all the models. In order to choose the best classifier and confirm its efficiency in classifying the test set, evaluations of validation sets should be taken into consideration. The accuracy and the recall are the performance metrics that should be most valued, in such a way to have a model that correctly predicts when a word is ambiguous (polysemy 1). The model that best performed according to these criterions is the Random Forest Classifier, as previously shown the evaluation metrics: accuracy equals to 0.95 and f1-score [1] equals to 0.95. On the test set the following metrics were scored by the model, the results are shown on table 4.5 and confusion matrix 4.6. This model, applied to a set of data never seen before, performs well enough.

A picture containing chart

Description automatically generated

Table

Description automatically generated

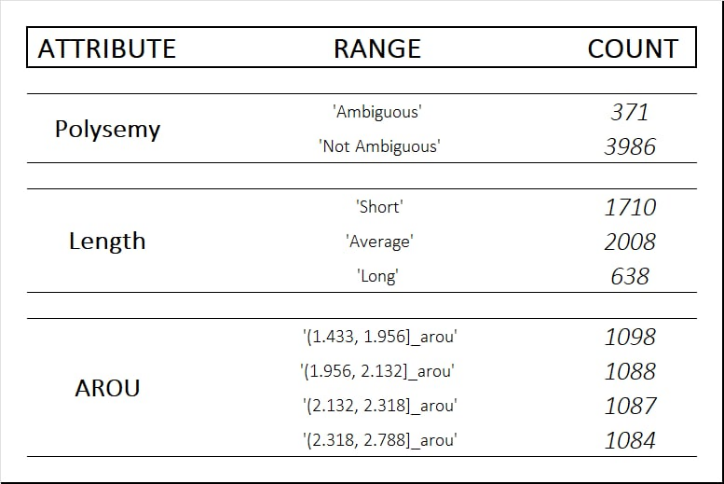
# **4 PATTERN MINING**

This section presents a methodology of **Association Analysis**, which is useful for discovering interesting relationships hidden in large data sets. The uncovered relationships can be represented in the form of sets of item, which are known as **Frequent Itemsets**, and **Association Rules**, that represent relationships between two itemsets.

Therefore, the strategy is to decompose the problem into two major subtasks:

* Frequent Itemset Generation, where objective is to find all the itemsets that satisfy the minimum support threshold.
* Rule Generation, where objective is to extract all the high confidence rules from the frequent itemsets found in the previous step.

In the following sections will we see, at first, some preprocessing steps in order to prepare dataset for the further analysis. After we will extract frequent patterns from the data and then discuss those patterns with qualitative and quantitavice analysis. As a next step we will extract association rules using different values of confidence, followed by the discussion of explored rules. As a final step, we will see the summerized evaluation of all the activities carried out through this section.

**DATA PREPARATION**

To make correct association analysis, it has been decided to: firstly, split continuous attributes into intervals; secondly, transform the categorical attributes; lastly, for a better understanding, add abbreviations at the end of each value to recognize the attribute at which they refer.

Figure 4.0.1 represents the final result of the transformed attributes. On the figure we cans see only two categorical and one continuous attribute, only because all other continuous attributes have been first treated and then visualized in the same way.

## **4.1 FREQUENT PATTERN EXTRACTION**

Frequent itemsets extraction has been carried out by Apriori program from Pyfim library.

**APRIORI ALGORITHM**

Apriori is a program to find association rules and frequent items with the Apriori algorithm, which carries out a search on the subsets and determines the support of item sets by subset tests. This implementation is pretty fast.

The minimum support and minimum confidence are set by us, and are parameters of the Apriori algorithm for association rule generation. These parameters are used to exclude rules in the result that have a support or a confidence lower than the minimum support and minimum confidence respectively.

Table

Description automatically generatedChart, line chart

Description automatically generatedDuring the process of frequent itemsets extraction different values of support have been tested (3%, 10% and 20%) considering different types of itemsets (all, closed, and maximal). Table 4.1.2 shows the number itemsets of each type by different support values. Considering the relationship between all presented types and looking at figure 4.1.3 we can say that ‘closed’ and ‘maximal’ represent subsets of itemsets extracted by ‘all’. Nevertheless, increasing the support, the number of itemsets decreases, but at the same time converges to the same value (sup = 20%, itemsets extracted = 30 for each type);

## **4.2 FREQUENT PATTERN DISCUSSION**

Including qualitative and quantitative analysis, e.g., how the number of patterns w.r.t k, min\_sup changes;

Chart, line chart

Description automatically generated

As we already mentioned in classification analysis our dataset records are not balanced with respect to target class, polysemy. As the number of records of ambiguous words (polysemy = 1) is really low, it has been decided to set all the parameters for frequent pattern extraction algorithm in a way that we could get patterns including both, positive and negative class. After examining, we decided to set k=2 as a minimum number of items in each itemset, in order to get some significant patterns of our target class. Figure 4.2.1. represents minimum number of items w.r.t different values of support.

Chart

Description automatically generated

Even though we mentiond imbalance of our target class in extracted patterns, we can clearly see on figure 4.2.2. the difference between positive and negative class w.r.t changing support values.

In order to evaluate the performance of previously presented different types, the itemsets extracted have been sorted by support value and presented first five (concequently the most frequent patterns) on figure 4.2.3. As ‘closed’ and ‘all’ frequent itemsets have shown almost the same results, only ‘all’ and ‘maximal’ have been analized with respect to changing support values. As we showed previously on figure 4.1.2 as support value increases the different types of frequent itemsets converges to the same sets. For example, on the following figure we see that for minimum support = 0.03 (3%), we have different set of first five most frequent itemsets with different support values.

As a maximal frequent itemset is a frequent itemset for which none of its immediate supersets are frequent, therefore it is obvious to have highest support equal to 8.95, while in the other case the highest support is 44.29. Additionally, it is natural that the first highest support itemset in ‘All’ does not change per value of support, while in ‘Maximal’ we have differences. Finaly, as support increases both types of itemsets converges to the same variables and max support value as mentioned earlier, which is represented on the figure as a last row corresponding to the minimum support value equal to 20%.

Graphical user interface, text, application

Description automatically generated

## **4.3 ASSOCIATION RULES EXTRACTION**

A picture containing shape

Description automatically generated

This part of the report extracts and analyses association rules between previously examined frequent itemsets. Figure 4.3.1 shows the relationship between value of confidence and number of rules with a minimum support of 3%. The flow demonstrates how the number of rules decreses with respect to the incresing number of confidence.

Chart, line chart

Description automatically generated

The figure 4.3.2 represents different values of confidence {20%, 30%, 40%, 50%} with respect to the relationship between support and number of itemsets. As the confidence and support increases the number of itemsets decreases.

## **4.4 ASSOCIATION RULES DISCUSSION**

Including qualitative and quantitative analysis, e.g., how the number of rules w.r.t k min\_conf changes, histograms of rules’ confidence and lift;

This section provides qualitative and quantitavi analysis on how the number of rules changes with respect to minimum confidence and support values. Figure 4.4.1. represents the different categories of rules with respect to threshold value of confidence of [20%, 30%, 40%, 50%] and support of [3%, 5%, 10%]. The different categories provided are total number of rules, rules where lift is higher than 1, rules where negative and positive target classes are predicted.

Graphical user interface, application

Description automatically generated with medium confidence

As the dataset is unbalance with respect to the target class, the number of positive class (ambiguous words) are very low, therefore increasing min confidence or support value was decreasing the chance of getting association rules predicting positive class. Finally, the analysis of association rules are provided only for these low values of min confidence and support.

The following figure represents the relationship between minimum confidence and number of rules. The parameters are the following min\_sup = [2, 11], min\_conf = {20, 100} and k = 2.

Chart, histogram

Description automatically generated

## **4.5 FINAL DISCUSSION**

Final discussion about the association rules are carried out by taking into consideration the accuracy and preciseness of predicting target variable. In order to get the association rules that perform on target class, it has been decided to choose the parameter values presented in the previous section.

As a first phase, min\_sup = 3% and min\_conf = 30% has been examined on the whole dataset. Association rules predicting target class as positive was six in total. The rule with the highest lift value of 3.9709 have been chosen, implying antecedents as word’s ‘Length’ = short, ‘Imageability’ = (4.657, 6.031], and ‘Web Corpus Frequency’ = (24.445, 30.913] and consequent as ‘polysemy’ = ambiguous word. Different pairs of antecedent have been chosen and evaluated.

First, only length and Imageability values, that predicted in total of 515 words as ambiguous where the true positive values were 99. **Accuracy** = 0.8421, **Precision** = 0.1922, **Recall** = 0.2668, and **F1-score** = 0.2235.

Second, only web corpus frequency and length values, that predicted in total 548 words as ambiguous, where true positive values were 136. **Accuracy** = 0.8515, **Precision** = 0.2482, **Recall** = 0.3666, **F1-score** = 0.2959.

Third, only imageability and web corpus frequency values, that predicted in total 246 words as ambiguous, where true positive values were 61. **Accuracy** = 0.8864, **Precision** = 0.2479, **Recall** = 0.1644, **F1-score** = 0.1977.

On the second phase, the association rules have been extracted from the Training set built in classification part and the most interesting rule has been tested in predicting a target class on Test set. Only two association rules have been extracted from training set, the one with the highest lift value provided the same actecedents as previously, so the second one has been chosen for further analysis. It implied antecedents as word’s ‘Size’ = (1.374, 3.412], and ‘Gender’ = (4.121, 4.647], and consequent as ‘polysemy’ = ambiguous word. Total number of words that have been predicted as ambiguous were 71, and out of total 14 prediction have been True. **Accuracy** = 0.8823, **Precision** = 0.1972, **Recall** = 0.1261, **F1-score** = 0.1538.

It is significant for this dataset to know how many true instances have been correctly predicted, therefore the main performance indicators that have been taken into consideration are recall and F1-score, which is also a harmonic mean of precision and recall.

In conclusion, the strongest association rule selected with min\_sup = 3% and min\_conf = 30% threshold values, have been the one implying antecedents as word’s ‘Length’ = short and ‘Web Corpus Frequency’ = (24.445, 30.913] (the highest quarter range). However, the extracted rules in general are unsuitable to predict the entire dataset since they have very low level of support and confidecence. therefore, it has to be taken into consideration that the extracted rules are not good enough to apply rule-based classifier.