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**

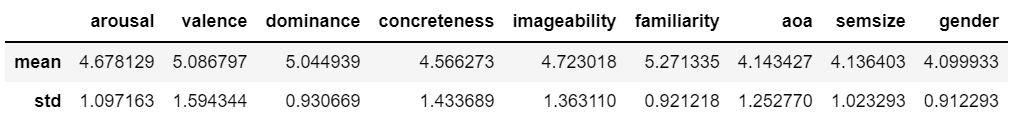
This section of the report represents two tables for both datasets – (1) Words Glasgow & (2) Words Polysemy which are provided as an input data. We can say that both datasets are high-dimensional as the number of attributes is quite high.

Through the variable “Name” we indicate the name of the attribute/variable we are considering. In total we have 13 attributes (columns 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 - Contininuous (float) | [12770.0, 2022459848.0] |

Out of all features we distinguish 9 variables as 9 dimensions of each record in our dataset, which are: 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**

In the following section the distribution of variables are represent with the help of different kind 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). We observed that 4303 words has polymesy equal to 0, and 379 equal to 1. Figure 1.2.1 and 1.2.2 shows the distribution of words with respect to ‘Polysemy’ attribute.

A picture containing bar chart

Description automatically generatedChart, pie chart

Description automatically generated

Chart, histogram

Description automatically generatedChart, bar chart

Description automatically generatedChart, box and whisker chart

Description automatically generated

We visualised density plots, bar charts and boxplots, similarly as previous, in order to learn the statistics and distribution of all the variables with respect to our target variable ‘Polysemy’.

Table

Description automatically generated

**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.

First 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).

Graphical user interface

Description automatically generated with low confidenceChart, histogram

Description automatically generated

After 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.

We 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.

Chart, histogram

Description automatically generatedChart, histogram

Description automatically generated

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.

Chart

Description automatically generatedAfter 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, 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 only continuous values.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 all continuous attributes in dataset. Even though all three of the scalers have been examined, the outputs in the following sections are presented on 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 thats 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.

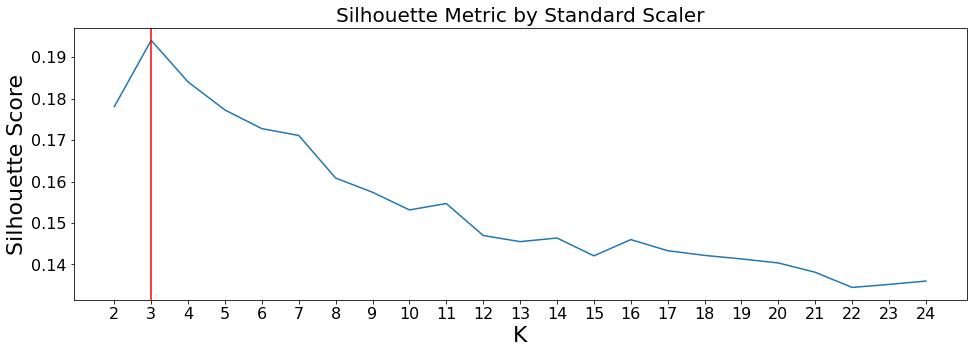
**Elbow Method** helps to plot the WCSS (Within Clusters Summed Squares) values ​​and selects the point where the parameter value falls more than the previous value. For each point, the Sum of Squared Error (SSE) is the distance to the nearest cluster. To get SSE, we square these errors and sum them. 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 w.r.t cluster’s number equal to 6 and Silhouette Coefficient w.r.t cluster’s number equal to 3.

The following table is a demonstration of Silhouette Score and SSE w.r.t number of clusters.

Table

Description automatically generated

Chart, scatter chart

Description automatically generated

PUT THE CLUSTER PLOTS

3 Clusters:

A picture containing window, building, green, different

Description automatically generated

6 Clusters:

A picture containing colorful, different, colors, painted

Description automatically generated

Table

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

Chart, line chart

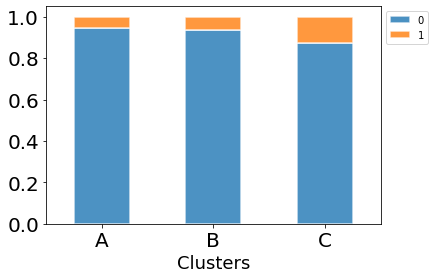
Description automatically generated

**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.

**Barchart**

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

 Chart

Description automatically generated Chart, bar chart

Description automatically generated

Chart, 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.

**2.3 ANALYSIS BY HIERARCHICAL CLUSTERING**

Hierarchical 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.

Text

Description automatically generatedThere 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 ‘ward’ and ‘average’ methods. However, silhouette scores have been measured in order to represent the best method for hierarchical clustering and on figure 2.3.4 we can see the highest score which is achieved by ‘average’.

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 generated

In 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 to say 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.

Steps:

* 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 our 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 our 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:

* **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}.

In the following table we represent 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.

A picture containing icon

Description automatically generated

After that we started to build the Decision Tree model. 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

DESCRIBE DECISION TREE !!!!!!

Icon

Description automatically generated with low confidence

For the unbalanced data we repeated same steps, using the same algorithms that were used to select the best hyperparameters (criterion = 'entropy', max\_depth = 12, min\_samples\_split = 10, min\_samples\_leaf = 5), measuring the importance of the features, and finally implementing Decision Tree model shown below (fig. ).

**3.1.2 PERFORMANCE EVALUATION OF THE ALGORITHM**

*Evaluate the performances of the algorithm(s) w.r.t. confusion matrix, accuracy, precision, recall, F1, ROC curve - 9 pts.*

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. By the comparison of the tables 4.2 and 4.3, as first impression, the model built in the balanced training set feels to perform better. It shows in the validation set an accuracy value (0.92) greater than the classifier built in 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.

**ROC CURVE**

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**

Therefore, in the following sections only the balanced case is reported since it performed better.

**CONFUSION MATRIX**

A picture containing text, screenshot, electronics

Description automatically generated

We 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.tr (train set). With Entropy:

Figura 2: Validation Set

**LEARNING CURVE**

A picture containing chart

Description automatically generated

**3.2 CLASSIFICATION BY OTHER ALGORITHMS**

**3.2.1 KNN CLASSIFIER**

KNN 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.

A picture containing chart

Description automatically generated

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

Description automatically generated

A picture containing text, electronics, screenshot

Description automatically generated

**3.2.2 RANDOM FOREST CLASSIFIER**

The Random Forest is an ensemble algorithm which was used in this case to combine different decisional

tree models. Randomized Search technique was used to determine the best parameters used into the

implementation. The parameters chosen were: 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 in training and validation sets as before.

A picture containing application

Description automatically generated

The figure 1.2 represents the confusion matrix on train set (D.tr) where we can see that Random Forest classifier predicted well 1943 records as polysemy = 1 (True Positive) and it misclassified 30 records as False Positive; on the other hand, it well classified 1922 records as Polysemy = 0 (True Negative) and misclassified only 9 records as False Negative.

A picture containing text, monitor, electronics, screenshot

Description automatically generated

On the validation set it predicted TP:823, FP:67, TN:770, FN:14.The accuracy of D.tr is 0.99, same for f1 score; the accuracy of D.val is 0.95 and so is f1 score.

Table

Description automatically generated

Line chart

Description automatically generated with medium confidence

**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 [Yes] equals to 0.95.

On the test set the following metrics were scored by the model, the results are shown in table 4.5 and

confusion matrix 4.6.

This model, applied to a set of data never seen before, performs well enough if we consider the correctly

classified record with the target variable ‘1’.

RESULTS:

Table

Description automatically generated

A picture containing chart

Description automatically generated

UNBALANCED:

A picture containing PowerPoint

Description automatically generated

BALANCED:

Chart, scatter chart

Description automatically generated