The following report is about analyzing the dataset which is realized by the Economic and Social Research Council (ESRC).

We take into consideration 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 we are using a set of 5553 English words, we want to analyze 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.

**1.1 DATA SEMANTICS**

In this section we represent 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.

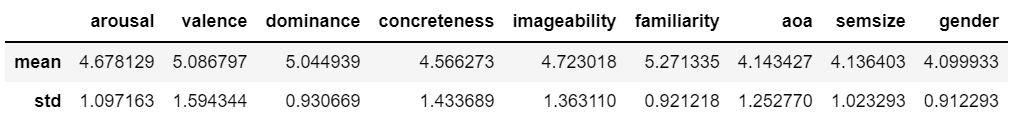
The first three dimensions (arousal, valence, dominance) are used to measure the emotional and psycological empact of a word. All the other variables indicate the grade of knowing the word and the use of it.

|  |  |  |  |
| --- | --- | --- | --- |
| **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 the following:

Table

Description automatically generated



The following is the table of the dataset (2) – ‘Words Polysemy’, where in total we have 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 this section we represent the distribution of variables with the help of histograms and charts.

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.

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**

This 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**

For this part of the project we decided to apply some transformations in order to represent data in a scale considered more suitable.

First transformation that we decided to apply was 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 tried to fill remained missing values with mean and median, as we did not face any significant difference in these two ways of substitution, we filled the missing values with the mean of the attribute.

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.

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

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

Chart, bar chart

Description automatically generated

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).

**Goal**

Gain insight in your data

1 with respect to your project goals

2 and general

**Find answers to the questions**

1 What kind of attributes do we have?

2 How is the data quality?

3 Does a visualization helps?

4 Are attributes correlated?

5 What about outliers?

6 How are missing values handled?

**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 before we apply any of those, we need to go through some preprocessing steps.

First of all, we need to normalize the ranges of our 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.

Chart, box and whisker chart

Description automatically generated

**Minmax Scaler**

Scaled between the range [0, 1].

Chart, box and whisker chart

Description automatically generated

**Robust Scaler**

Works good on outliers and consideres interquartile ranges instead of std.

Chart, box and whisker chart

Description automatically generated

For the further clustering analysis we remove all the categorical variables and work with only continuous ones.

**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 our case we chose 3 as an initial number for clusters.

In order to find optimal number of clusters we can use 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. In cluster analysis it is used to determe the number of clusters in a data set.

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.

The Most common measure is Sum of Squared Error (SSE). For each point, the error 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.

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:

Shape

Description automatically generated with low confidence Chart, line chart

Description automatically generated

Finally, we can sum up that 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.

PUT THE CLUSTER PLOTS

We also checked the number of records in each cluster and numbers were almost equally separated.

**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.

**2.2 Analysis by density-based clustering**

In the analysis by density-based clustering we dont have to choose the clusters, but we have to choose eps (radius of analysis) and min. of samples (minimum number of neighbours of the point withing the radius).

With DBscan we can see the actual clusters and the noise points.

We used this clustering method to compare detected noise points to the outliers explored previously by boxplots.

**2.3 Analysis by hierarchical clustering**

Four different methods: Average, Complete, Single & Ward

Chart, bar chart

Description automatically generatedChart

Description automatically generated

Chart

Description automatically generatedChart, histogram

Description automatically generated

Average:

Average is between single and complete,

good against noise and outliers, bad with clusters of different number of elements

Ward:

good against noise and outliers, good for spherical/rounded clusters.

Single:

Complete:

**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**

*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.

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.

Chart, line chart

Description automatically generated**A picture containing chart

Description automatically generated**

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

**CONFUSION MATRIX**

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:

A picture containing text, screenshot, electronics

Description automatically generated

Figura : Train Set

A picture containing text, screenshot, electronics

Description automatically generated

Figura : Validation Set

**ROC CURVE**

ROC curve with GINI index and criterions:

Line chart

Description automatically generatedChart, line chart

Description automatically generated

ROC curve with Entropy criterion:

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

**LEARNING CURVE**

GINI:

Chart

Description automatically generated

With ENTROPY:

Chart

Description automatically generatedA picture containing chart

Description automatically generated

**3.2 CLASSIFICATION BY OTHER ALGORITHMS**

**3.2.1 KNN CLASSIFIER**

**3.2.2 RANDOM FOREST CLASSIFIER**

Table

Description automatically generated

**3.2.2 DUMMY CLASSIFIER**

• Try to use another classification algorithm (KNN or Random Forest, suggested) or use a

baseline model for the comparison;

**3.3 FINAL DISCUSSION**

Which is the best algorithm? Best can be studied w.r.t. the performance evaluation or

other preferred point of view;

UNBALANCED:

A picture containing PowerPoint

Description automatically generated

BALANCED: