IS assignment 2
text classification report

Janez Ignacij Jereb, Stevan Stević

January 2024

1 Introduction

This is the report on the second assignment for Intelligent systems. For this assignment we needed to create a text classification model. We focused mostly on training many different models and testing their learning on different datasets (generated with resampling) and different vectorization methods (TF-IDF, Word2Vec). We also provide an exploratory analysis of the dataset.

2 Exploratory analysis of the dataset

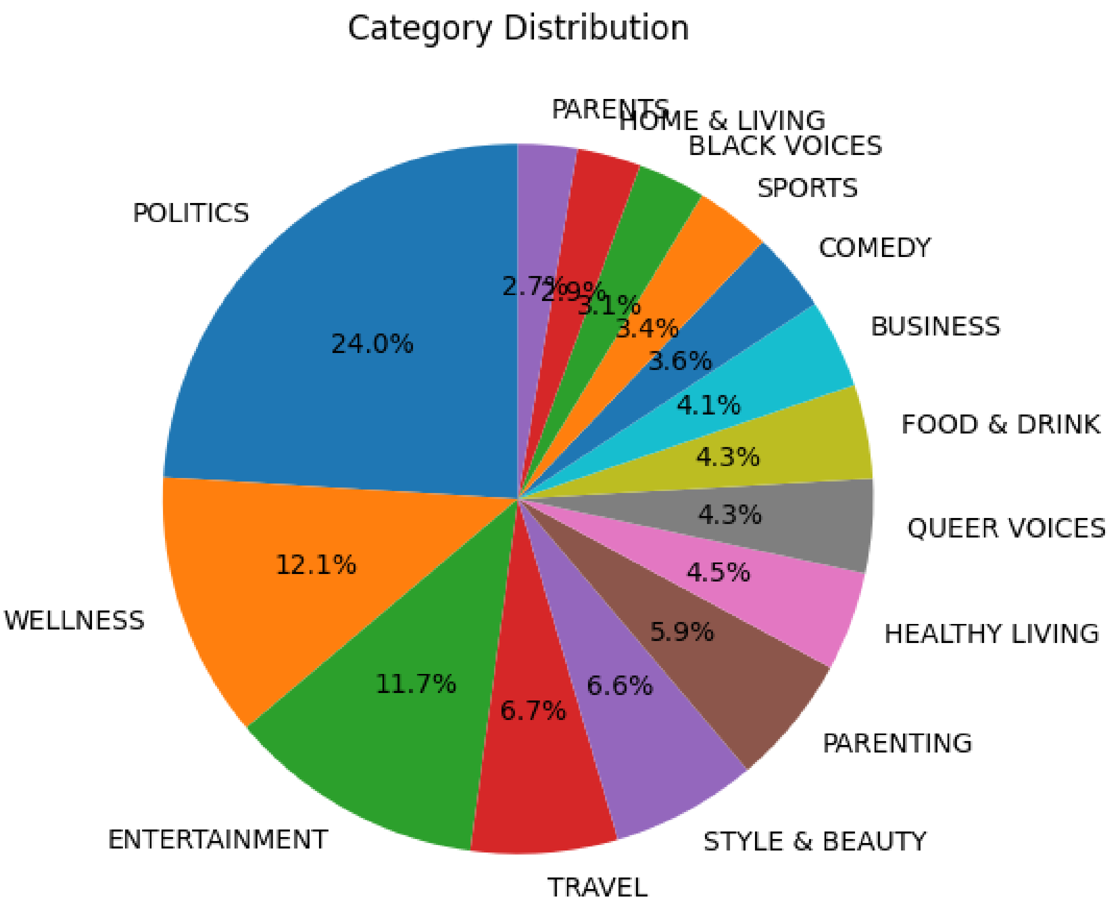
2.1 Dataset presentation

In this work, we utilized a portion of the News Category Dataset comprising 148,122 news headlines spanning from 2012 to 2022, sourced from HuffPost. The dataset also includes useful metadata. This portion was (probably) obtained by removing 27 least occurring categories (from 42). Includes 6 attributes: news category, headline, authors, link, short description and publication date. More information about the dataset including an exploratory data analysis and various applications are provided by Rishabh Misra[2].

2.2 Simple statistics

2.2.1 Category distribution

The dataset mostly consists of news of category Politics, with wellness, entertainment consisting more than half of the dataset.



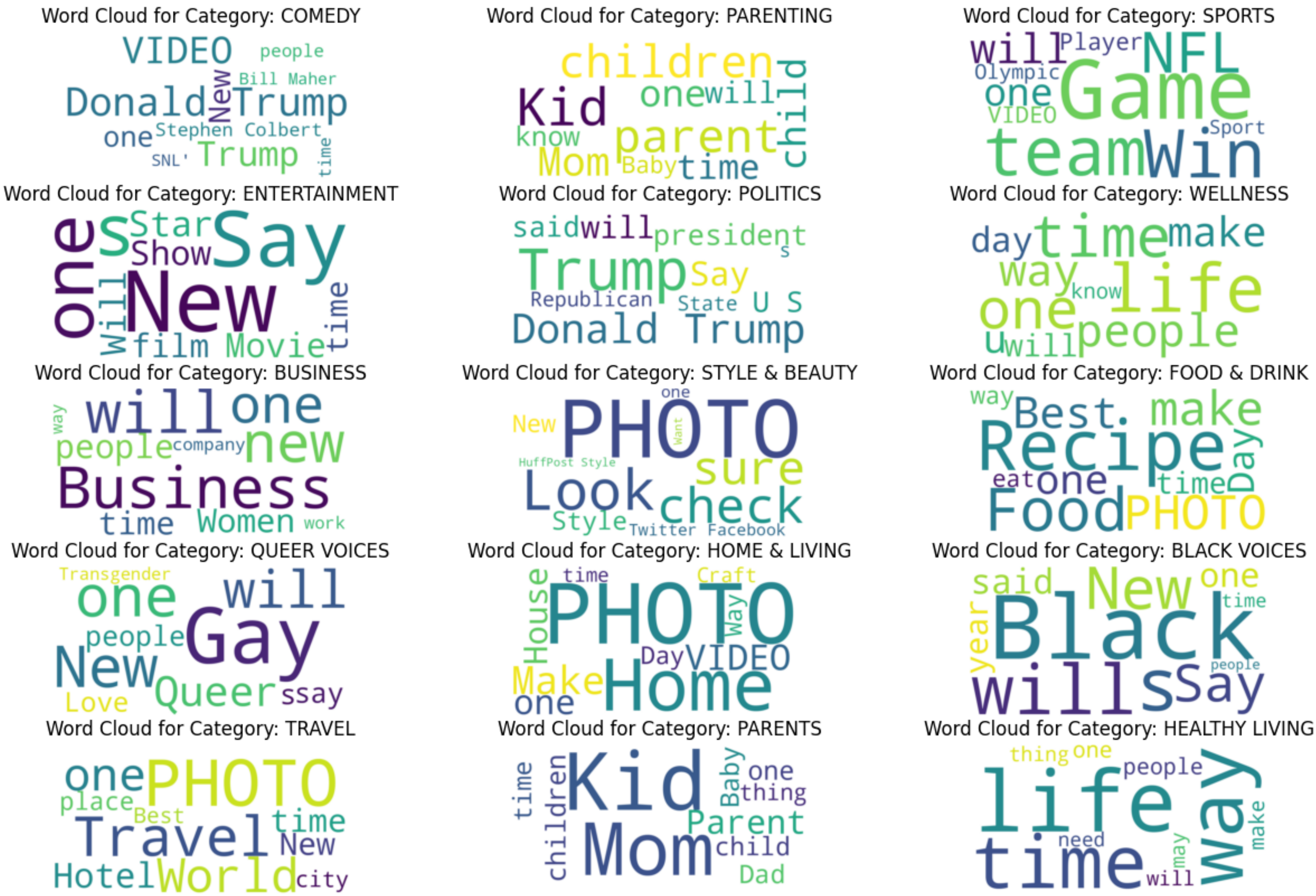
2.2.2 Missing values

Only headline and short description contain null values. These do not overlap.

Column name	Missing value count
headline	731
category	0
authors	0
date	0
short_description	736
short_description & headline	0

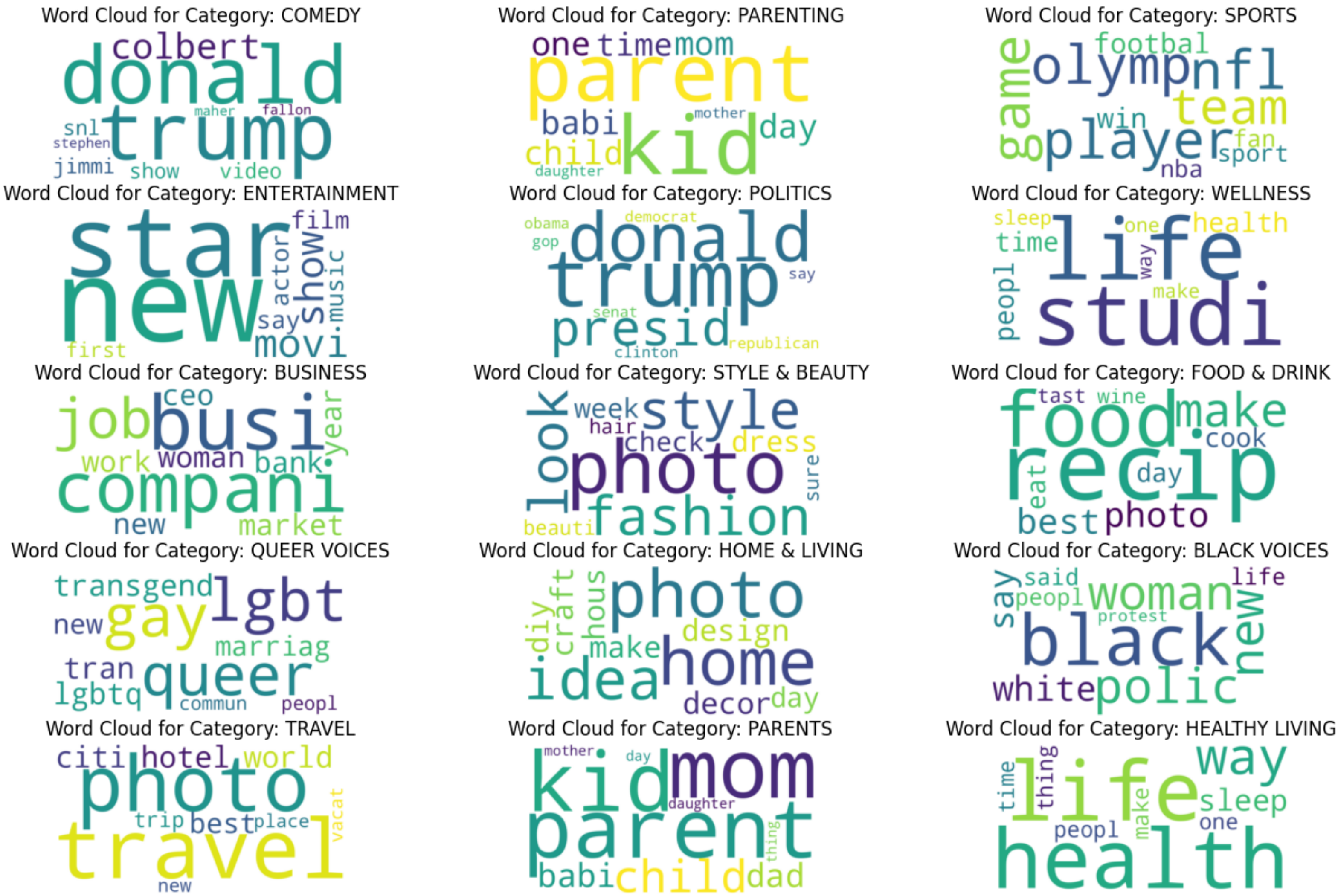
2.2.3 Word frequency

This word clouds show that there are defining words for categories such as new, business, Gay, Game, Black. But there are many which are ambiguous such as Kid, Photo, Life, Trump.



2.2.4 Analysis using average TF-IDF vectors

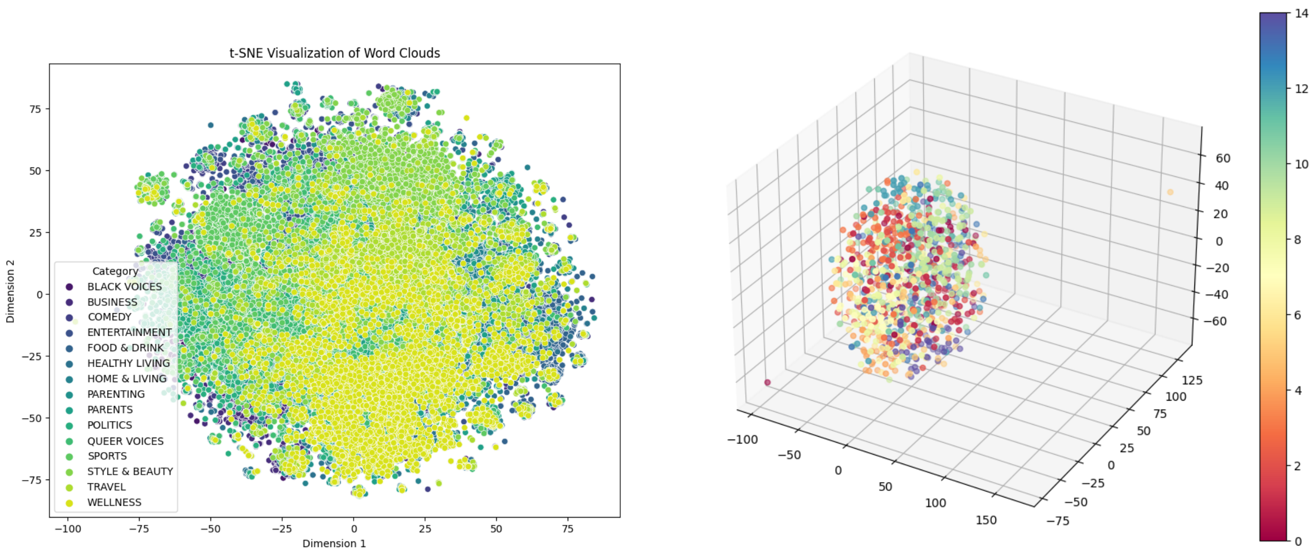
To have a more accurate view of word occurrence we also analysed average TF-IDF values. Most words from word clouds persist, but their importance changed.



2.2.5 Visualizing word space

2.2.5.1 Using t-SNE

We tried to visualize semantic word space using pretrained vectorizer LexVec[1] and t-SNE for dimension reduction. We tried using a sample of the original dataset and an undersampling, both returned disappointing results.



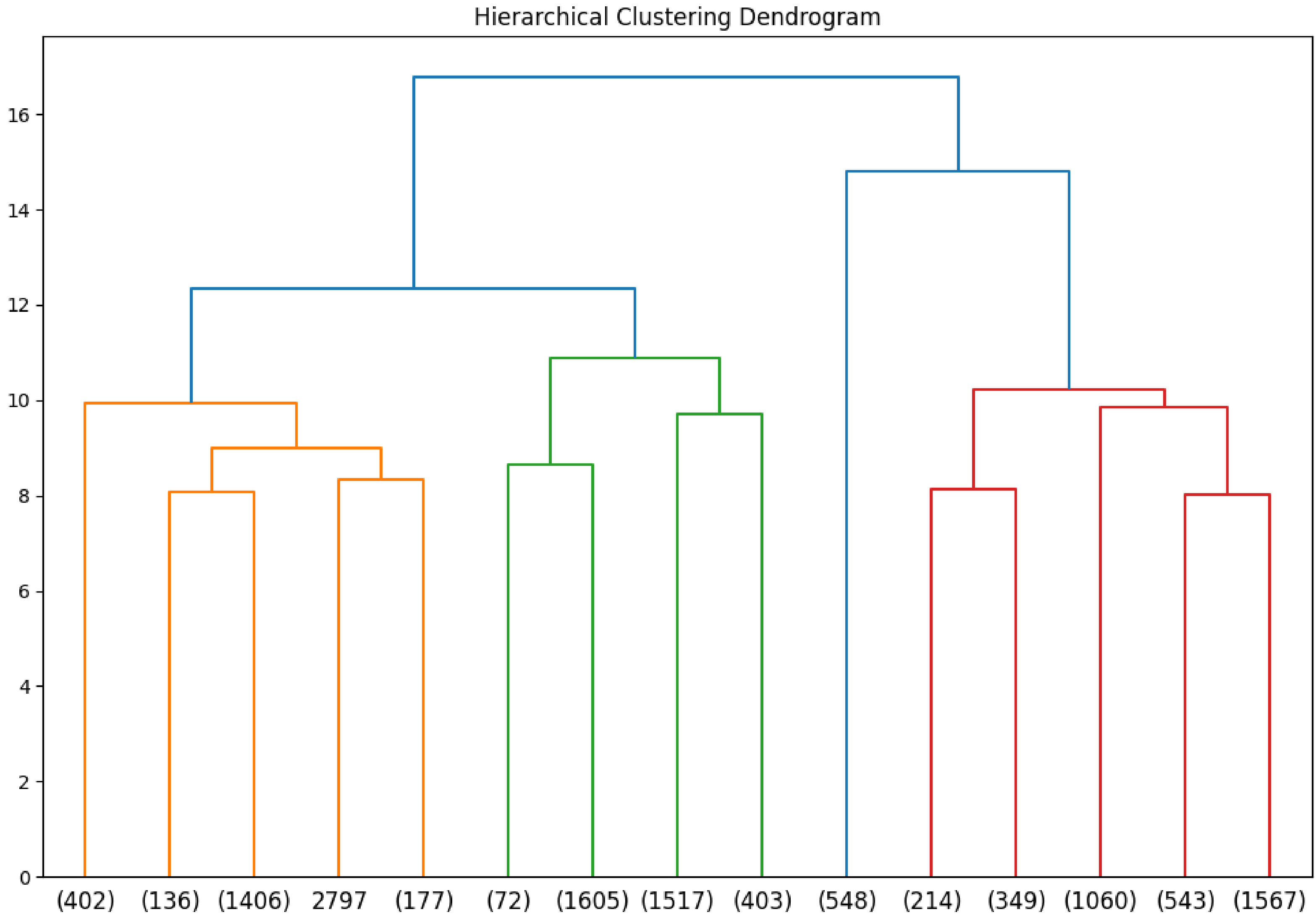
(a) 2D projection

(b) 3D projection

Figure 1: Visualizations generated with t-SNE

2.2.5.2 Using Hierarchical clustering on the undersampled dataset

Hierarchical clustering provided some interesting insights. It shows that there are four main clusters the bigger three are mixed almost uniformly while the smallest consists mostly of food & drink. If we increase the number of clusters to 15 we see that one cluster contains only comedy; sports and food & drink have almost an entire cluster. We see which categories overlap such as politics and comedy, politics and home & living, wellness and healthy living, business and entertainment.



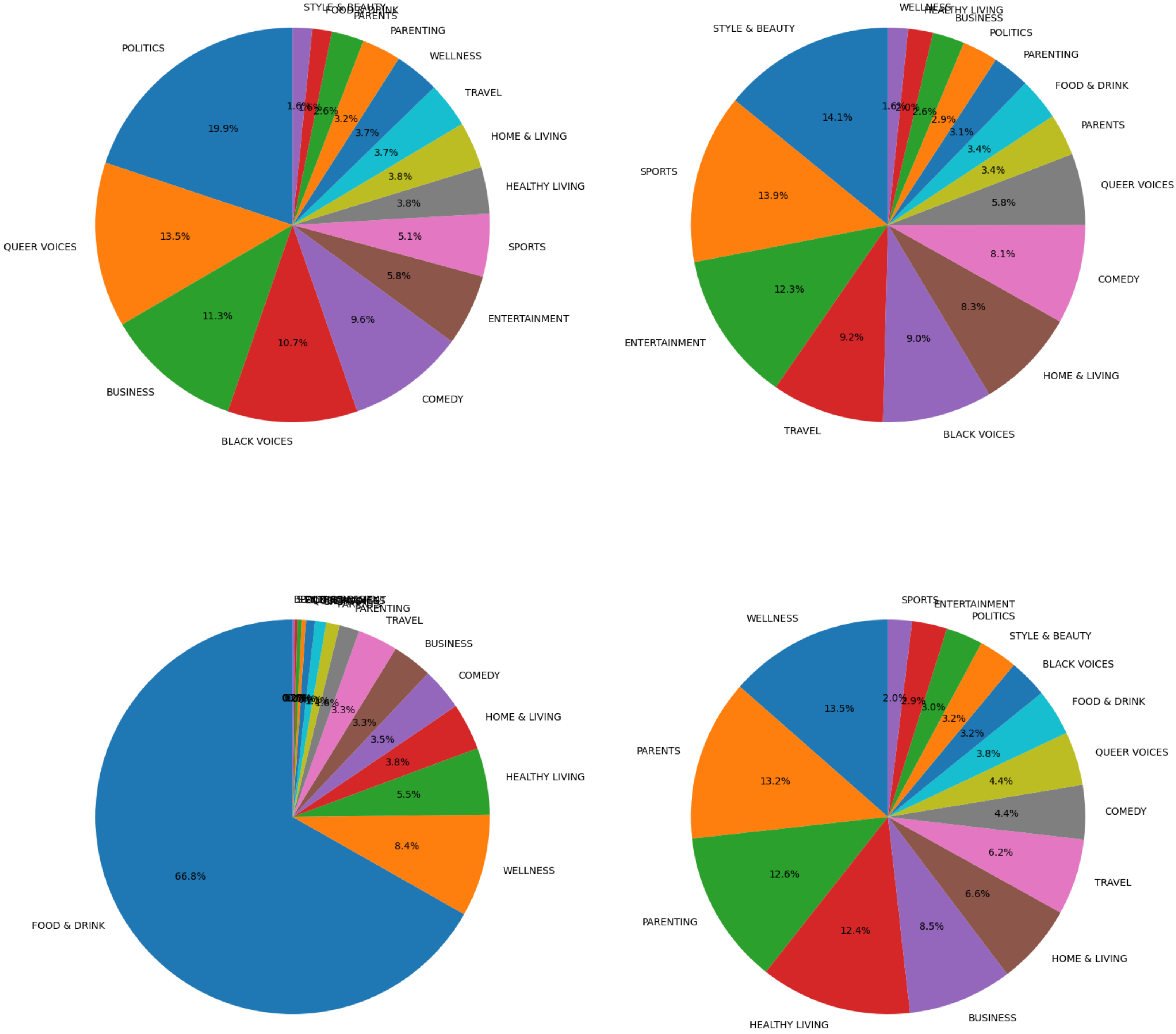


Figure 2: Category distribution of the four main clusters

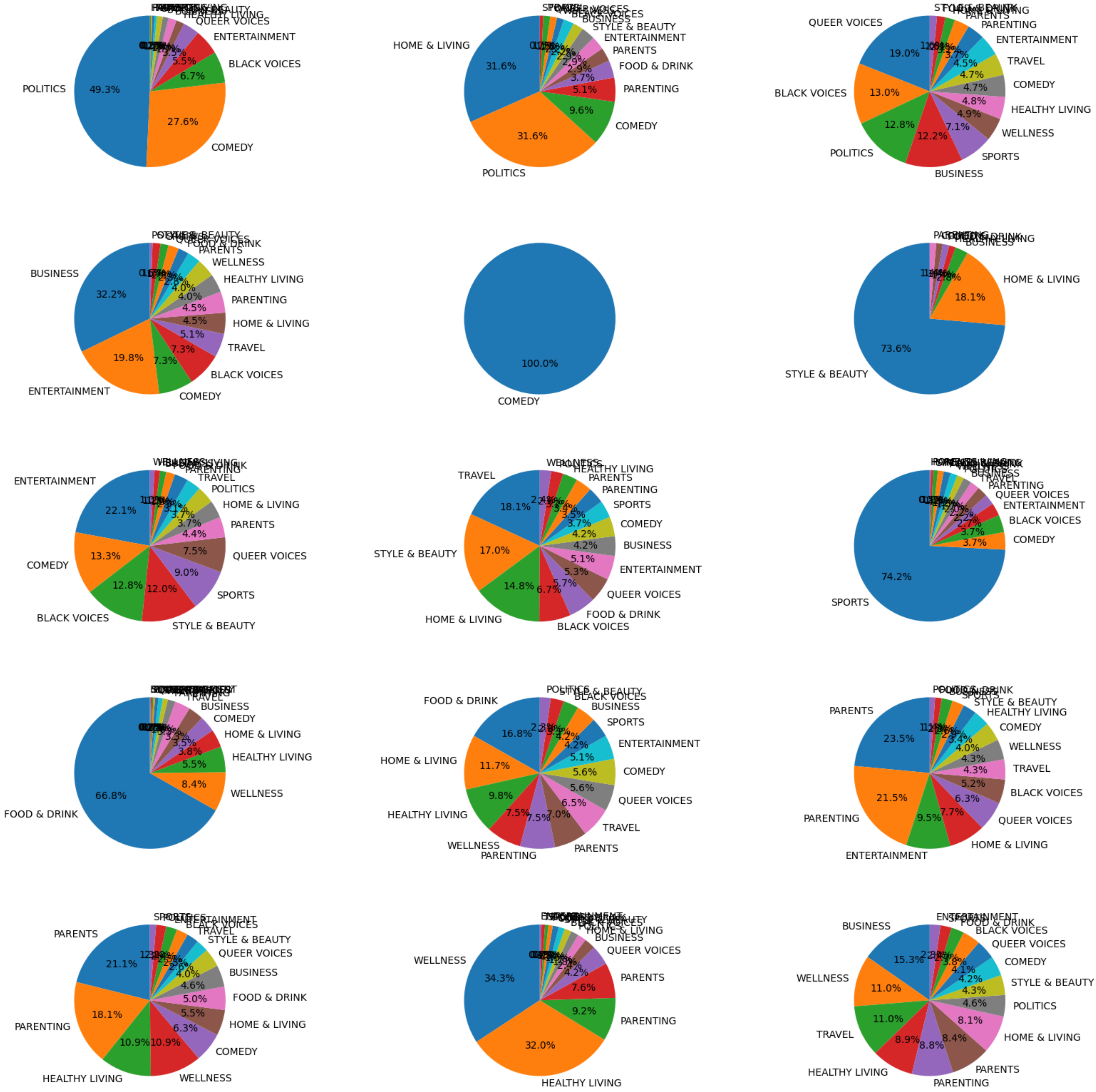


Figure 3: Category distribution of the 15 main clusters

2.2.5.3 Using Dummy classifiers

To see the baseline for models we "trained" a "random" classifier which chooses randomly with the distribution of the dataset and a majority classifier which chooses the most frequent class (politics). Results show that the base precision is 0.11, base recall is 0.24, base specificity is 0.88 and base f1 is 0.11.

Random classifier	precision	recall	specificity	f1	geom. mean	ind. bal. acc.	support
BLACK VOICES	0.0325	0.0329	0.9688	0.0327	0.1785	0.0289	4530.0000
BUSINESS	0.0444	0.0438	0.9603	0.0441	0.2050	0.0382	5940.0000
COMEDY	0.0318	0.0314	0.9637	0.0316	0.1740	0.0274	5349.0000
ENTERTAINMENT	0.1144	0.1139	0.8830	0.1142	0.3172	0.0929	17185.0000
FOOD and DRINK	0.0374	0.0375	0.9569	0.0374	0.1893	0.0325	6274.0000
HEALTHY LIVING	0.0457	0.0459	0.9547	0.0458	0.2093	0.0398	6623.0000
HOME and LIVING	0.0328	0.0325	0.9712	0.0327	0.1778	0.0286	4272.0000
PARENTING	0.0602	0.0602	0.9407	0.0602	0.2379	0.0516	8708.0000
PARENTS	0.0255	0.0252	0.9735	0.0254	0.1568	0.0222	3922.0000
POLITICS	0.2410	0.2426	0.7584	0.2418	0.4290	0.1745	35233.0000
QUEER VOICES	0.0388	0.0384	0.9575	0.0386	0.1917	0.0334	6280.0000
SPORTS	0.0385	0.0383	0.9660	0.0384	0.1924	0.0336	5036.0000
STYLE and BEAUTY	0.0673	0.0676	0.9334	0.0674	0.2512	0.0576	9731.0000
TRAVEL	0.0696	0.0705	0.9324	0.0700	0.2565	0.0601	9810.0000
WELLNESS	0.1232	0.1225	0.8799	0.1228	0.3282	0.0996	17762.0000
avg precision	0.1112	0.1112	0.1112	0.1112	0.1112	0.1112	0.0000
avg recall	0.1114	0.1114	0.1114	0.1114	0.1114	0.1114	0.0000
avg specificity	0.8889	0.8889	0.8889	0.8889	0.8889	0.8889	0.0000
avg f1	0.1113	0.1113	0.1113	0.1113	0.1113	0.1113	0.0000
avg geom. mean	0.2898	0.2898	0.2898	0.2898	0.2898	0.2898	0.0000
avg ind. bal. acc.	0.0864	0.0864	0.0864	0.0864	0.0864	0.0864	0.0000
total support	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	146655.0000

Majority classifier	precision	recall	specificity	f1	geom. mean	ind. bal. acc.	support
BLACK VOICES	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	4530.0000
BUSINESS	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	5940.0000
COMEDY	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	5349.0000
ENTERTAINMENT	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	17185.0000
FOOD and DRINK	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	6274.0000
HEALTHY LIVING	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	6623.0000
HOME and LIVING	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	4272.0000
PARENTING	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	8708.0000
PARENTS	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	3922.0000
POLITICS	0.2402	1.0000	0.0000	0.3874	0.0000	0.0000	35233.0000
QUEER VOICES	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	6280.0000
SPORTS	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	5036.0000
STYLE and BEAUTY	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	9731.0000
TRAVEL	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	9810.0000
WELLNESS	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	17762.0000
avg precision	0.0577	0.0577	0.0577	0.0577	0.0577	0.0577	0.0000
avg recall	0.2402	0.2402	0.2402	0.2402	0.2402	0.2402	0.0000
avg specificity	0.7598	0.7598	0.7598	0.7598	0.7598	0.7598	0.0000
avg f1	0.0931	0.0931	0.0931	0.0931	0.0931	0.0931	0.0000
avg geom. mean	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
avg ind. bal. acc.	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
total support	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	146655.0000

3 Data preprocessing

3.1 Text Cleaning

In the text cleaning phase we keeped only headline, short_description and category columns. Tokenized headline and short_description by words. Lemmatized and stemmed tokens using WordNetLemmatizer, PorterStemmer from NLTK. We also removed stopword (which were obtained with NLTK). For model training and testing we used the concatenation of the columns headline and short_description. At last we removed all invalid rows from the dataset.

3.2 Vectorizations and Other Presentations

In the vectorization and presentation phase, various techniques were applied to transform and prepare the data for further analysis.

- **TF-IDF Vectorization:** Utilized the `TfidfVectorizer` to transform text data into TF-IDF vectors, limiting the feature space to a maximum of 5000 features.
- **Word2Vec Vectorization:** Employed `Word2Vec` to create word embeddings with a vector size of 100. This technique represents words as vectors in a continuous vector space. Vectorization is applied on each token and the final result is the average of vectors (denoted `avg w2v`).
- **MinMax Scaling for Word2Vec:** The `MinMaxScaler` was utilized to scale the average Word2Vec vectors to a range $[0, 1]$ (denoted `avg w2v scaled`). It was used for naive bayes classifiers.

These techniques provided diverse representations of the data, capturing different aspects for comprehensive analysis and model training.

3.3 Data Split

In the data splitting phase, the dataset was divided into three subsets: `orig`, `rus`, and `ros`. The split ratio was 80% for training and 20% for testing.

- **Original Dataset (`orig`):** - This subset represents the original, unaltered data.
- **Random Undersampled Dataset (`rus`):** - The `rus` subset is created through random undersampling. This technique involves reducing the number of instances in the majority class to balance class distribution.
- **Random Oversampled Dataset (`ros`):** - The `ros` subset is generated through random oversampling. This involves increasing the number of instances in the minority class to address class imbalance.

These subsets serve different purposes, allowing for the training and evaluation of models on various versions of the dataset with different class distributions.

4 Data vectorization and other data presentations

5 Overview of the models

Many different models were tested on different datasets and vectorizations to see how these influence model's performance. The selection of models and application of hyperparameter tuning were done under memory and time constraints imposed by Google Colab. We have chosen parameters for which there was no automatic selection for them and were not important to the model.

5.0.1 AdaBoost Classifier (`AdaBoostClassifier`):

- **Description:** AdaBoost (Adaptive Boosting) Classifier is an ensemble method that combines weak learners to create a strong classifier.
- **Vectorizations tried:** `TF-IDF vectorization, average word2vec vectorization`
- **Hyperparameters tried:** -
- **Optimal parameters:** -

5.0.2 Complement Naive Bayes (`ComplementNB`):

- **Description:** Complement Naive Bayes is an adaptation of the standard Multinomial Naive Bayes classifier for imbalanced data.
- **Vectorizations tried:** `TF-IDF vectorization, average word2vec vectorization (scaled)`
- **Hyperparameters tried:** α `for values 0.00013894954943731373, 0.281176869797423, 0.0002120950887920190 0.6551285568595508`
- **Optimal parameters:**
  - Original, TF-IDF: α=0.281176869797423
  - Oversampled, TF-IDF: α=0.6551285568595508
  - Original, avg w2v: α=0.00013894954943731373
  - Oversampled, avg w2v: α=0.00021209508879201905

5.0.3 Gaussian Naive Bayes (`GaussianNB`):

- **Description:** Gaussian Naive Bayes classifier assumes that features follow a normal distribution.
- **Vectorizations tried:** `TF-IDF vectorization, average word2vec vectorization (scaled)`
- **Hyperparameters tried:** `50 value from logarithmic scale from 1 to 10exp-9`
- **Optimal parameters: -**

5.0.4 Logistic Regression (`logregression`):

- **Description:** Logistic Regression is a linear model for binary and multiclass classification. Cross-validation included through RidgeClassifierCV.
- **Vectorizations tried:** `TF-IDF vectorization, average word2vec vectorization`
- **Hyperparameters tried: -**
- **Optimal parameters: -**

5.0.5 Multinomial Naive Bayes (`MultinomialNB`):

- **Description:** Multinomial Naive Bayes classifier is suitable for classification with discrete features.
- **Vectorizations tried:** `TF-IDF vectorization, average word2vec vectorization (scaled)`
- **Hyperparameters tried:** `Hyperparameters tried:  50 value from logarithmic scale from 1 to 10exp-9`
- **Optimal parameters:**
  - Original, TF-IDF: α=0.281176869797423
  - Oversampled, TF-IDF: α= 0.009540954763499934
  - Original, avg w2v: α=0.0007543120063354615
  - Oversampled, avg w2v: α=1.0481131341546853e-07

5.0.6 Random Forest Classifier (`RandomForestClassifier`):

- **Description:** Random Forest Classifier is an ensemble of decision trees with randomness introduced during training.
- **Vectorizations tried:** `TF-IDF vectorization, average word2vec vectorization`
- **Hyperparameters tried:** max_depth for values 2, 4, 10 (RandomGridSearchCV did crash on Google Colab, so it was done manually)
- **Optimal parameters:** max_depth=10

5.0.7 Ridge Classifier (`ridge`):

- **Description:** Ridge Classifier is a linear model that uses L2-norm for regularization. Cross-validation included through RidgeClassifierCV.
- **Vectorizations tried:** `average word2vec vectorization`
- **Hyperparameters tried: -**
- **Optimal parameters: -**

5.0.8 Support Vector Classifier (`svc`):

- **Description:** Support Vector Classifier (LinearSVC) is a linear SVM for classification.
- **Vectorizations tried:** `TF-IDF vectorization`
- **Hyperparameters tried:** `C for values 0.1, 1, 10`
- **Optimal parameters:**
  - Original, TF-IDF: C=0.1
  - Oversampled, TF-IDF: C=10

6 Overview of results

Model	Vectorization	Fit Time (s)	Test Time (s)	Avg Precision	Avg Recall	Avg Specificity	Avg F1
AdaBoost orig	avg w2v	122.84558200836182	0.5259435176849365	0.5442	0.5756	0.9447	0.5439
AdaBoost orig	tfidf	378.5199770927429	10.508871078491211	0.4734	0.4251	0.8584	0.3577
AdaBoost ros	avg w2v	109.74340319633484	0.4973893165588379	0.4513	0.4564	0.9612	0.4444
AdaBoost ros	tfidf	377.17439317703247	10.9009850025177	0.5945	0.3697	0.9548	0.4015
ComplementNB orig	avg w2v scaled	33.01250147819519	0.015288114547729492	0.4728	0.5227	0.9141	0.4261
ComplementNB orig	tfidf	159.33163046836853	1.1400470733642578	0.6703	0.6722	0.9529	0.6357
ComplementNB ros	avg w2v scaled	31.378465175628662	0.022382259368896484	0.5230	0.4783	0.9629	0.4518
ComplementNB ros	tfidf	163.45824480056763	1.1326518058776855	0.6472	0.6516	0.9751	0.6414
GaussianNB orig	avg w2v scaled	0.44372105598449707	0.20717692375183105	0.6145	0.5941	0.9654	0.6009
GaussianNB orig	tfidf	9.845949172973633	12.109414100646973	0.5245	0.2057	0.9704	0.1853
GaussianNB ros	avg w2v scaled	0.45205140113830566	0.2182445526123047	0.5621	0.5417	0.9672	0.5419
GaussianNB ros	tfidf	9.718861818313599	12.114492416381836	0.5396	0.4352	0.9595	0.3952
logregression orig	avg w2v	352.5325565338135	0.06722331047058105	0.6618	0.6742	0.9574	0.6518
logregression ros	avg w2v	324.3756229877472	0.07582592964172363	0.6076	0.6097	0.9721	0.6080
MultinomialNB orig	avg w2v scaled	31.032079458236694	0.011812210083007812	0.2439	0.2383	0.7660	0.0958
MultinomialNB orig	tfidf	155.75294733047485	1.0726559162139893	0.6870	0.6800	0.9503	0.6494
MultinomialNB ros	avg w2v scaled	29.743656873703003	0.01397395133972168	0.5460	0.5263	0.9661	0.5242
MultinomialNB ros	tfidf	148.98659086227417	1.1193251609802246	0.6858	0.6843	0.9774	0.6844
RandomForest orig	avg w2v	115.16363644599915	0.6198108196258545	0.6782	0.6027	0.9342	0.5395
RandomForest orig	tfidf	53.965874910354614	1.2370004653930664	0.5439	0.2894	0.7828	0.1828
RandomForest ros	avg w2v	103.47948789596558	0.6246218681335449	0.6560	0.6426	0.9744	0.6414
RandomForest ros	tfidf	53.62057900428772	1.2318823337554932	0.5849	0.5114	0.9650	0.5242
Ridge orig	avg w2v	3.0185704231262207	0.08026123046875	0.6260	0.6331	0.9412	0.5783
Ridge ros	avg w2v	3.0338809490203857	0.09703421592712402	0.5740	0.5853	0.9704	0.5712
SVC orig	tfidf	270.16023111343384	0.974750280380249	0.7168	0.7228	0.9625	0.7020
SVC ros	tfidf	305.685373544693	0.9919228553771973	0.7485	0.7515	0.9822	0.7495

Table 1: Model Performance Metrics

6.1 Results

Result interpretation was done using the provided appendix. Initially we expected for resampling to make a greater impact on learning but it mostly depends on the model. Oversampling could amplify the risk of overfitting but results do not vary enough to imply that.

6.1.1 Vectorization methods

Vectorization method did not have an universal effect on the quality of predictions. Classifiers based on naive bayes did not show any common preference. AdaBoost, GaussianNB and RandomForests did better with word2vec while ComplementNB, MultinomialNB with TF-IDF.

6.1.2 Sampling methods

Different sampling methods also did not provide any universal improvements of prediction. AdaBoost, MultinomialNB, RandomForest, SVC did train better with the oversampled dataset while GaussianNB, LogisticRegression did perform better on the original split.

6.1.3 Types of models

The best performing model is LinearSVC and the worst performing is AdaBoost. With the right vectorization and sample most models achieved f1 score of around 65%. Ensemble methods performed surprisingly poorly.

6.1.4 Interpretation of confusion matrices

Most models had problems distinguishing between categories parenting, parent and healthy living, wellness. Many combinations manifest pathological behavior:

- RandomForest, oversampled, TF-IDF: mostly predicts business
- GaussianNB, original, TF-IDF: "chaotic" behavior (there are many outlying miscategorizations)
- RandomForest, original, TF-IDF: mostly acts as a majority classifier
- MultinomialNB, original, word2vec: mostly acts as a majority classifier
- AdaBoost, original, TF-IDF: partially acts as a majority classifier

7 Discussion

One of our regrets is using Google Colab non-locally because of its great limitations. While the goals of this assignment mostly imply searching for the best model our approach of trying many different models on different datasets and vectorizations does not provide highly performant models. The strength of our work is mostly on the exploratory data analysis and analysis of the effects of different vectorizations and samplings. For a more succesful search for the best model it would be better to focus on only best models and not on compiling information of the tested models. Most interesting findings were presented in sections hierarchical clustering and result interpretation.

References

[1] *LexVec*. `https://github.com/alexandres/lexvec`. Accessed: 2024-01-13.

[2] Rishabh Misra. *News Category Dataset*. 2022. arXiv: `2209.11429 [cs.CL]`.