University of New South Wales

COMP9417

Text Classification Group Report

Group 42

Authors	**Student Numbers**
David Wu	z5209802
Simon Hanly-Jones	z5149715
Jessie Shen	z5305064
Jorge Andres Esguerra Alarcon	z5297157
Alexander Nguyen	z5059289

Introduction

We have been given two sets of labelled data, a training set and a testing set. These sets of data, respectively, contain 9500 and 500 instances of news articles represented as a set of words categorised under a variety of different topics; we also have 10 readers, each interested in a specific topic. Our objective will be to find the 10 most relevant articles for each topic out of a test set of 500 articles for each “interested reader.” As we are dealing with human language (written text in news articles) this problem falls under the branch of **Natural Language Processing** (NLP) in artificial intelligence.

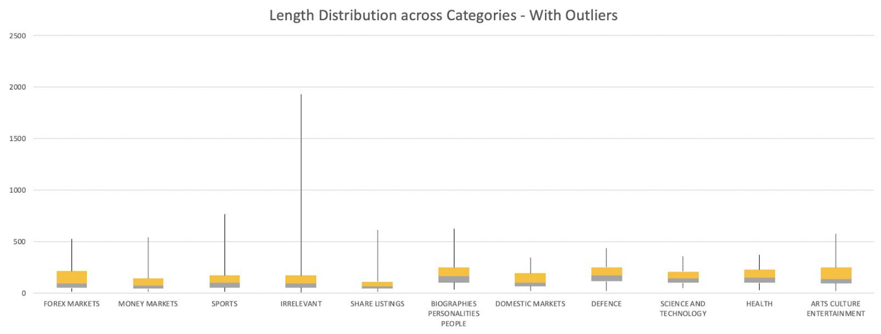
The main issue with NLP scenarios is that computers do not understand text and the human language very well. So to accomplish this challenge of recommending articles to 10 different readers, we preprosessed each instance of data so that it could be interpreted more easily. Then, a series of text classification models were implemented and trained with this preprocessed training data. These models were evaluated and fine-tuned before the final model was ultimately chosen to recommend the articles.

For this project we have explored several different classification models: a decision tree, a logistic regression, and a support vector machine model. Each was evaluated according to the Accuracy, Precision, Recall, and F1 statistics of each model as well as its confusion matrix, we then chose the one with the best performance for our final recommendation.

Exploratory Data Analysis

An exploratory analysis of the data revealed significant differences in average article length by category, meaning that models which do not normalise for this discrepancy will likely be biased towards categories which are generally longer (Appendix A).

Figure 1. Distribution of article length by subject



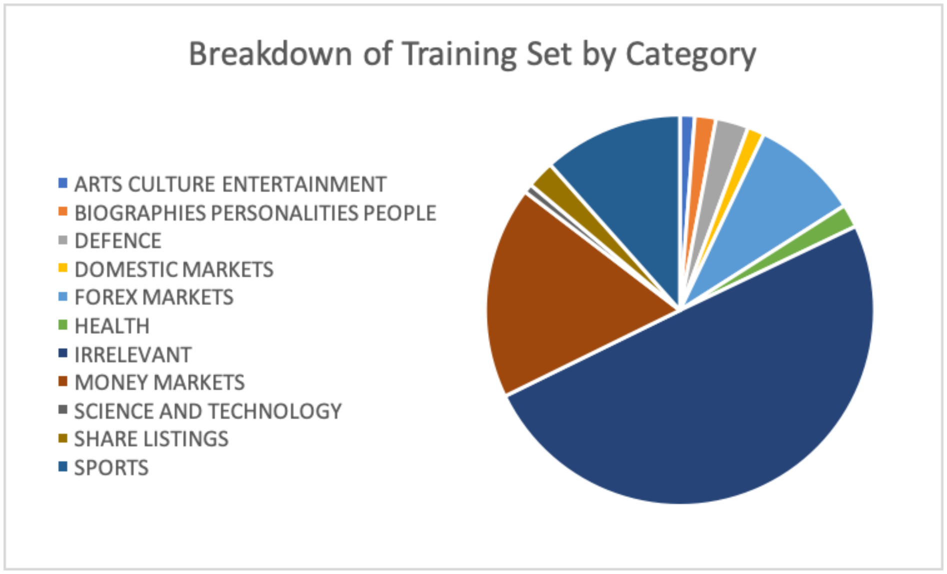
In terms of the data ranges we can see that the range of lengths for the topics vary greatly, with the Irrelevant topic having the greatest range between the maximum and minimum (though the maximum length is a significant outlier), whilst Science and Technology have the lowest range.

If we look at the interquartile ranges for each category, they were actually quite small relative to the overall range for each category - meaning that for the most part, the data is quite normally distributed with the exception of some significant outliers. From this we can interpret that the variability of the data is quite low with some categories having much less variability than others (for example, the Share Listing category).

Because of these observations, we cannot draw any definite conclusions on whether to use a high-variance and low-bias model or high-bias and low-variance model for our text classification and instead we decided we should test one of each to gain some empirical evidence.

In addition, in order to figure out if any undersampling or oversampling of different groups was needed, we also looked into the approximate split by category of the training data set.

Figure 2. Distribution of training set by category



As we can see in the above figure, irrelevant articles make up the bulk of the training set. However, even within the relevant articles, certain categories like Money Markets, followed by Sports and then followed by Foreign Exchange Markets, occupy a larger share of the training set than the other categories do.

Because our data set is not balanced with regards to different categories, our model may need to oversample the categories that are underrepresented and undersample the categories that are overrepresented; otherwise, our model could be biased towards classifying articles into the over-represented categories.

Furthermore, since there is such an imbalance in the categories of our training set it is also imperative (even if it was not set out in the assignment specifications) that we use other evaluation metrics such as Precision, Recall, and F1 to measure the “success” of our predictive models, as they account for imbalanced data rather than computing the pure accuracy statistics of the model on the test set.

Methodology

Pre-processing

The first step of any machine learning pipeline is the Data Cleaning and Pre-Processing. With the given data set, much of this step has already been completed for us. From our exploratory data analysis, we were unable to find any instances of missing data; the set of words representing each article had already been converted to lower-case, removed of punctuation and stopwords, undergone tokenization and stemming. To feed the data into Sci-kit Learn's feature extraction libraries, we replaced the commas (",") and underscores ("_") with spaces (" ") of the article_words column. SKLearn's feature extraction libraries tokenize words separated by spaces, not commas, which is why the first change was needed; the second change was added after an initial processing of the data, upon which it was noticed that several nonsense words containing underscores were present.

Feature Extraction

We originally decided to use the Bag of Words model to represent the data, as the data came in the form of sets of words to represent each article. By using SKLearn's CountVectorizer class we extracted and transformed the data into vectors of words with corresponding token counts as features, and the resulting category of the articles as our labels.

But, as some features appear numerous times throughout all the articles, they might be less indicative of features of specific categories; we decided to extract features with SKLearn's TfidfVectorizer and TfidfTransformer to create features based on the Term-Frequency Inverse-Document-Frequency values of each document, which is a measure of how frequently each word appears in each document compared to the overall body of text. We then used these two forms of extracted features to train and develop the classification models below.

An alternative we considered using for the vectorization of the words in every article was to process every N-gram instead of each word separately, when vectorizing. This is a strategy suggested by Kapadia (2019), in which the model is trained based on the consecutive n-sized grams that appear in the text. When dealing with separate words, we are in fact training our model with all the 1-Grams in the articles. Training with N-grams with N >= 2 helps our text classification models differentiate between the usage of words depending on context. However, we could not apply this strategy as the initial input was already preprocessed, holding only the words that appeared in every article, but not in the real order they appeared, to build coherent n-grams.

Choice of Classification Models

We decided to explore four different classification models to have more to compare and contrast; this way we could choose the highest-performing model for our final analysis. We chose to develop a Decision-Tree model, Logistic Regression model, Support Vector Machine (SVM) model and Multinomial Naive Bayes (MNB) model. We chose these four because Decision-Tree and SVM models are both high variance but low bias, whilst Logistic Regression and MNB models are both low variance and high bias; as we saw in the exploratory data, because each category has articles that are roughly the same length, we decided to try models on both ends of the bias-variance tradeoff spectrum.

Cross Validation - Choice of Metric

For each model, five-fold cross validation was used to obtain the optimal hyperparameters. However, given the disproportionately high number of irrelevant articles in the dataset and the specifics of the assigned task, it was necessary to select a cross validation evaluation metric that selected parameters weighted to the model's performance per class. For this reason, we chose the weighted F1 score as it allowed each class to be treated equally. This was chosen to maximise the chance that each user would get more relevant recommendations for their chosen subject, rather than maximising absolute overall accuracy. The consequences of this choice will be explored in detail in the discussion section of this report

Construction of the Machine Learning Pipeline and Hyperparameter Tuning:

Despite each model being vastly different, we were able to create a machine learning pipeline that enabled us to treat each model as a black-box when we were fine-tuning hyper parameters. We were able to achieve this by having a standardised inputs (Eg. Bag of Words or TFIDF Vectors) that would work for all models along with a consistent performance metric that we would evaluate all the models on.

We ultimately decided to use SKLearn's GridsearchCV function as our main form of tuning in order to reduce both over and under-fitting. This is because it was the most efficient method that worked on cross-validation results.

Results

Table 1. Classifier Models Training Set Cross Validation Results Comparison

Model	**‘Best’ Parameters**	**Weighted F1**
Logistic Regression	Feature Extraction: TFIDF (min_df = 5, C=10, max_iter=10000, solver='saga')	0.76
Multinomial Naive Bayes	Feature Extraction: TFIDF alpha=0.01, class_prior=None, fit_prior='True'	0.75
Support Vector Machine	Feature Extraction: TFIDF C = 5, kernel = ‘linear’	0.76
Decision Tree	Feature Extraction: TFIDF class_weight=None, criterion='gini', max_depth=70, max_features=None, min_impurity_decrease=0.0, min_impurity_split=None, min_samples_split=2, min_weight_fraction_leaf=0.0, splitter='best'	0.68

An analysis and comparison of the results can be found in the Discussion section.

Final Model Decision

Ultimately, we chose the **Logistic Regression** model with hyper-parameters {(min_df = 5, C=10, max_iter=10000, solver='saga')}. While the results of the SVM and the Logistic Regression models tied in performance, we ultimately chose to implement the Logistic Regression model because it managed to better differentiate and classify minor categories, while the SVM model would overlook recommending articles for minor categories. This is because the original objective was to provide **all** users with the best list of recommended articles.

The hyperparameters chosen above were optimised with this objective in mind. Having C set to 10 meant that there was significantly less regularisation compared to the default model's value of C = 1. As C represents the inverse of regularisation strength (λ), this represents very minimal penalisation. By increasing C, we actually increase the model's tendency to overfit; however, this seemed to actually counteract the imbalance between recommended topics and allow the model to perform better when classifying topics with smaller samples. In terms of the 'solver' parameter, 'saga' was selected as 'liblinear' is generally only reserved for binary classification on small data sets, whilst 'saga' better handles multiclass problems focussing on minimising 'multinomial loss'.

In terms of feature selection, we chose to provide this model with TFIDF extracted features. Previously, we had tested this model using Bag of Words inputs, which marginally reduced accuracy with no benefit. A more detailed explanation of why TFIDF was better for all cases can be found in the discussion section. We chose a minimum limit of 5 for the number of documents that a word must appear in before it was considered as a feature to exclude words that were too uncommon to be informative.

Final Result Selection

The final article recommendations made were based on the probability assigned by the Logistic Regression model for each article. For each topic, a maximum of 10 articles were chosen. We gave some consideration to imposing some arbitrary probability threshold or cost function to favour exclusion rather than inclusion, however, this was found to be too detrimental to the already limited article recommendations that could be made for the smaller topics, such as Domestic Markets. Ultimately, it was determined to rely on raw classification, sorted by probability.

Figure 3. Confusion Matrix of Logistic Regression on Training Set

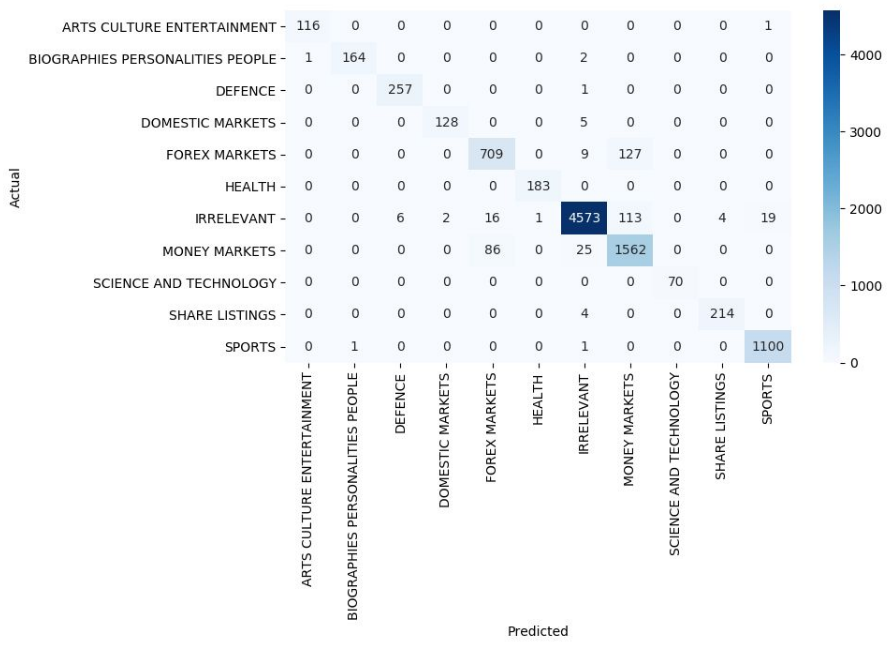


Table 2. Model Evaluation results of the Logistic Regression model on the test set.

Topic Name	**Precision**	**Recall**	**F1**
ARTS CULTURE ENTERTAINMENT	0.33	0.67	0.44
BIOGRAPHIES PERSONALITIES PEOPLE	0.80	0.27	0.40
DEFENCE	0.89	0.62	0.73
DOMESTIC MARKETS	1.00	1.00	1.00
FOREX MARKETS	0.46	0.38	0.41
HEALTH	0.73	0.57	0.64
MONEY MARKETS	0.54	0.64	0.58
SCIENCE AND TECHNOLOGY	0.00	0.00	0.00
SHARE LISTING	0.60	0.43	0.50
SPORTS	0.95	0.97	0.96

Table 3. Articles recommended by the Logistic Regression Model from the Test Set

Topic Name	Articles	Precision	Recall	F1
ARTS CULTURE ENTERTAINMENT	9526, 9604, 9703, 9789, 9830, 9952	0.33	0.67	0.44
BIOGRAPHIES PERSONALITIES PEOPLE	9854, 9896, 9933, 9940, 9988	0.8	0.4	0.53
DEFENCE	9559, 9576, 9607, 9616, 9670, 9770, 9773, 9842, 9987	0.89	0.8	0.84
DOMESTIC MARKETS	9796, 9994	1	1	1
FOREX MARKETS	9529, 9551, 9588, 9625, 9632, 9682, 9772, 9875, 9977, 9986	0.4	0.4	0.4
HEALTH	9609, 9661, 9810, 9833, 9873, 9911, 9926, 9937, 9947, 9978	0.7	0.7	0.7
MONEY MARKETS	9516, 9618, 9707, 9755, 9761, 9765, 9769, 9835, 9871, 9898	0.8	0.8	0.8
SCIENCE AND TECHNOLOGY	9617, 9982	0	0	0
SHARE LISTING	9518 ,9601, 9666, 9715, 9972	0.6	0.43	0.5
SPORTS	9569, 9573, 9620, 9752, 9760, 9787, 9857, 9886, 9942, 9997	1	1	1

Result Analysis: (See Appendix D. for Confusion Matrix)

Overall our model did a reasonable job at providing recommended articles to most of the classes for the exception of a few classes. We were incredibly surprised at our model's ability to classify the only two Domestic Market articles given the fact that this topic would have a strong overlap with any other financial topic (Eg Money Markets, Share Listing). Other categories like Sport and Defence also performed very well, most likely due to their distinct wording within the articles. However, our model's difficulty in classifying the three Science and technology articles, as it classified them as two as Irrelevant and one as Arts Culture. Most likely this was due to the fact that the three articles had all very different wording and showed no strong trend towards the training data articles. Ultimately we believe even though our model tried to minimise as much bias towards larger topics as it could it was still inherently biased towards these larger categories.

Discussion

Observations and Comparison

- Logistic Regression showed solid performance, along with the fact that it classified minority clases, such as Domestic Markets, better than all other models. (See *Appendix D.* for confusion Matrix on Testing Set.)
- Multinomial Naive Bayes showed decent performance and quick processing, but did not achieve a weighted F1 score as high as the top two models.
- Support Vector Machine showed solid performance in terms of weighted F1 score; however, it also seemed to also misclassify minority classes and bias towards topics with more articles. This was also the most time consuming model to do the Grid Search hyperparameter tuning on.
- Decision Tree showed the weakest results over the test set, scoring the lowest F1_score. Decision Tree models tend to overfit for the training set, and when tested against unseen data, the overly complex models lose a lot of precision. We decided to scrap the tuning of this model early on, as we could not make it achieve an accuracy greater than 73%. We concluded that Decision Tree was not a very suitable model for this task.

Generally, most models had difficulty with classes that had a very small number of articles, such as Science and Technology. Also, there was considerable difficulty in distinguishing between Forex Markets and Money Markets. In a real situation it would perhaps be more appropriate to combine the two very similar categories. Alternatively, one could consider training a subordinate model to reconsider and re-classify both of these categories only.

A more general observation was that feature extraction using TFIDF seemed to perform fractionally better than CountVectorizer (Bag of Words), most likely due to the imbalance between the topics. We also found that the models which had high-variance and low-bias took longer to run compared to models with high bias and low variance; the SVM model took the longest to run, most likely due a greater degree of computational processing power required due to weaker assumptions. Another observation was that high-variance-low-bias models tended to overfit significantly more than the low-variance and high-bias models; this was not surprising as high bias models over simplify, whilst high variance models tended to overcomplicate.

Discussion of Metrics

As referred to in Methodology, models were optimised with respect to F1 score weighted equally to each class, and not weighted according to proportion of articles in the training set. This choice is not without its compromises: using the weighted F1 score detracts slightly from the overall accuracy of a given model. For instance, a Multinomial Naive Bayes model with parameters trained by reference to accuracy achieves 76% accuracy, while one trained with reference to class weighted F1 score achieved only 75%. However, if one assumes that we want to provide every reader with the most relevant recommendations, then this compromise is worthwhile. Otherwise, if the model was trained in reference to accuracy, it would be biased towards recommending articles for readers whose preferred categories had a greater number of training set articles; in the worst case scenario this model could be biased towards optimizing for the 'Irrelevant' topic which provides no solution to the original machine learning problem.

In terms of model evaluation, consideration of the weighted F1 score of each topic should be given priority. This is for similar reasons to those set out above.

Future Improvements:

Further improvements could be made in the manner of data collection as well as model usage. The provided data has been preprocessed thoroughly to reduce words to their stems and to remove stop words. However, it has also been sorted and the order of the words was lost. This can be seen on inspection of the training and test sets where words are repeated several times in an article, then never again. As such, it is not possible to use any models that consider features that encode contextual data such as n-gram strategies which look for frequent, informative combinations of words based on their original order. In a practical situation, this would be a way to improve the performance of the classification.

Furthermore, finding a training set with a more balanced distribution among all the classes to which articles belonged could help us greatly improve the score metrics we proposed for our models. These Machine Learning models rely heavily on having enough examples so that the results can be generalized well, so having a short supply of articles of some of the categories definitely hindered performance.

Another improvement would be to train a subordinate model to classify only Forex Markets and Money Markets, due to the difficulty we had in segregating suggestions among these 2 categories. This stems from the similarity of the words used in both types of articles, as the word choice is very similar in both.

Conclusion

The goal of this machine learning task was to provide reading recommendations based on category selection. This task was accomplished with use of carefully tuned traditional machine learning methods and yielded interesting results.

Logistic Regression was ultimately selected as the best performing model, which was tuned with great care to return appropriate recommendations for as many topics as possible — including those underrepresented in the dataset. This achieved strong results when measured against the test dataset as a whole and also relatively strong results when measured against the top 10 selections from each topic, with potential users being presented with a considerable number of relevant articles for each topic aside from one. However, when this topic is examined in detail, one can see that the testing test contains only two articles for certain categories, which is not sufficient to obtain a realistic view of the models performance for this topic.

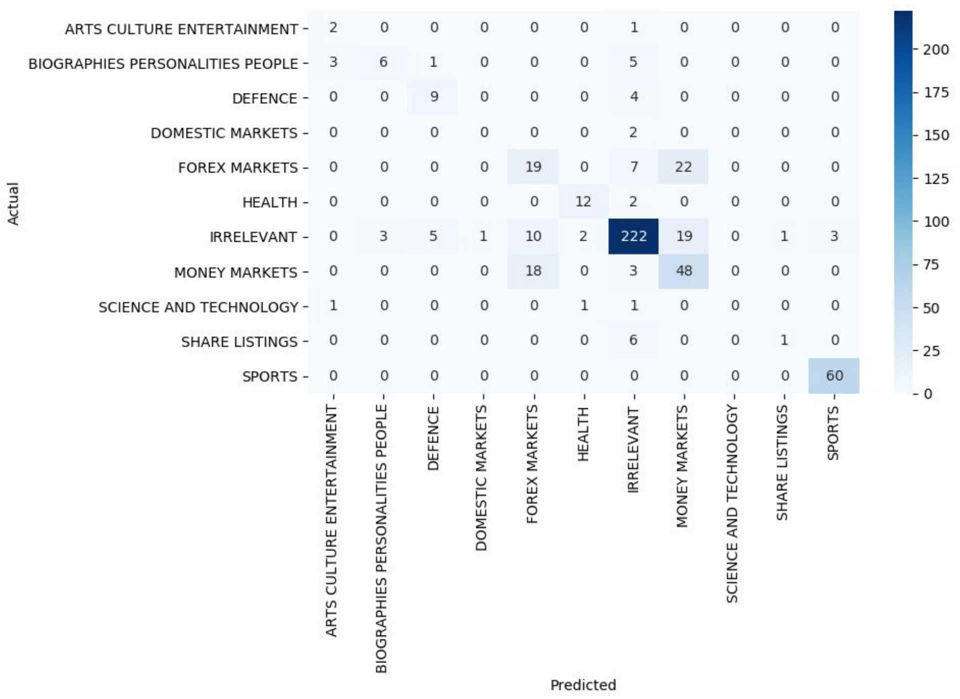
Further progression of this classification problem would be interesting and would require re-evaluation of the data pre-processing, in that it should remain unsorted to retain contextual information. A careful re-evaluation of the topics and whether they were truly distinct would also be beneficial to further development of this predictive task.

Appendix

Appendix A. Average article length in words by category of the Training Set.

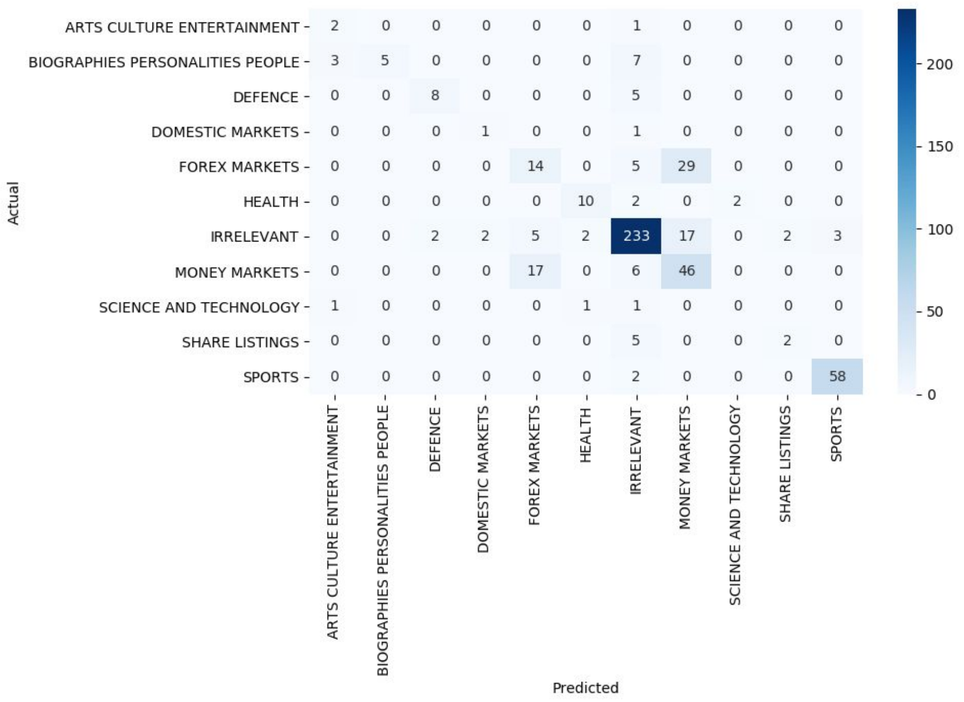
Category	Average Number of Words
ARTS CULTURE ENTERTAINMENT	178.8034188
BIOGRAPHIES PERSONALITIES PEOPLE	185.3113772
DEFENCE	186.3449612
DOMESTIC MARKETS	129.3458647
FOREX MARKETS	136.0295858
HEALTH	162.6885246
IRRELEVANT	125.2940431
MONEY MARKETS	110.8995816
SCIENCE AND TECHNOLOGY	164.2428571
SHARE LISTINGS	95.38990826
SPORTS	127.3892922

Appendix B. Multinomial Naive Bayes Confusion Matrix



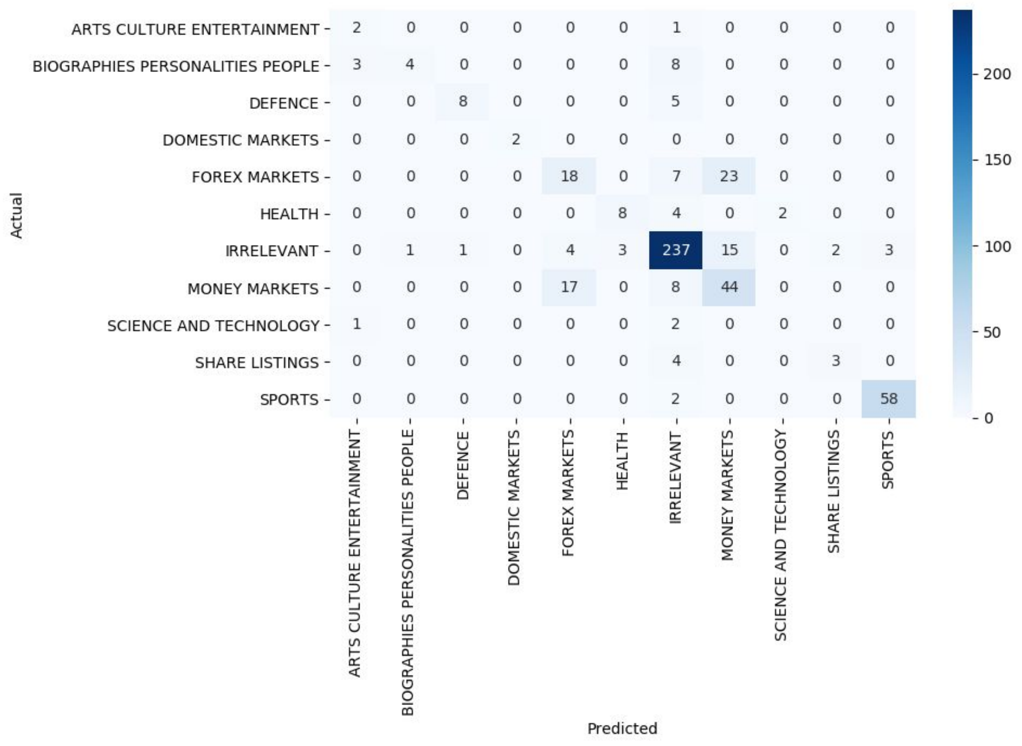
Actual \ Predicted	ARTS CULTURE ENTERTAINMENT	BIOGRAPHIES PERSONALITIES PEOPLE	DEFENCE	DOMESTIC MARKETS	FOREX MARKETS	HEALTH	IRRELEVANT	MONEY MARKETS	SCIENCE AND TECHNOLOGY	SHARE LISTINGS	SPORTS
ARTS CULTURE ENTERTAINMENT	2	0	0	0	0	0	1	0	0	0	0
BIOGRAPHIES PERSONALITIES PEOPLE	3	6	1	0	0	0	5	0	0	0	0
DEFENCE	0	0	9	0	0	0	4	0	0	0	0
DOMESTIC MARKETS	0	0	0	0	0	0	2	0	0	0	0
FOREX MARKETS	0	0	0	0	19	0	7	22	0	0	0
HEALTH	0	0	0	0	0	12	2	0	0	0	0
IRRELEVANT	0	3	5	1	10	2	222	19	0	1	3
MONEY MARKETS	0	0	0	0	18	0	3	48	0	0	0
SCIENCE AND TECHNOLOGY	1	0	0	0	0	1	1	0	0	0	0
SHARE LISTINGS	0	0	0	0	0	0	6	0	0	1	0
SPORTS	0	0	0	0	0	0	0	0	0	0	60

Appendix C. SVM Confusion Matrix



Actual \ Predicted	ARTS CULTURE ENTERTAINMENT	BIOGRAPHIES PERSONALITIES PEOPLE	DEFENCE	DOMESTIC MARKETS	FOREX MARKETS	HEALTH	IRRELEVANT	MONEY MARKETS	SCIENCE AND TECHNOLOGY	SHARE LISTINGS	SPORTS
ARTS CULTURE ENTERTAINMENT	2	0	0	0	0	0	1	0	0	0	0
BIOGRAPHIES PERSONALITIES PEOPLE	3	5	0	0	0	0	7	0	0	0	0
DEFENCE	0	0	8	0	0	0	5	0	0	0	0
DOMESTIC MARKETS	0	0	0	1	0	0	1	0	0	0	0
FOREX MARKETS	0	0	0	0	14	0	5	29	0	0	0
HEALTH	0	0	0	0	0	10	2	0	2	0	0
IRRELEVANT	0	0	2	2	5	2	233	17	0	2	3
MONEY MARKETS	0	0	0	0	17	0	6	46	0	0	0
SCIENCE AND TECHNOLOGY	1	0	0	0	0	1	1	0	0	0	0
SHARE LISTINGS	0	0	0	0	0	0	5	0	0	2	0
SPORTS	0	0	0	0	0	0	2	0	0	0	58

Appendix D. Logistic Regression Confusion Matrix on Testing Set


Actual
Predicted
200
150
100
50
0

Actual \ Predicted	ARTS CULTURE ENTERTAINMENT	BIOGRAPHIES PERSONALITIES PEOPLE	DEFENCE	DOMESTIC MARKETS	FOREX MARKETS	HEALTH	IRRELEVANT	MONEY MARKETS	SCIENCE AND TECHNOLOGY	SHARE LISTINGS	SPORTS
ARTS CULTURE ENTERTAINMENT	2	0	0	0	0	0	1	0	0	0	0
BIOGRAPHIES PERSONALITIES PEOPLE	3	4	0	0	0	0	8	0	0	0	0
DEFENCE	0	0	8	0	0	0	5	0	0	0	0
DOMESTIC MARKETS	0	0	0	2	0	0	0	0	0	0	0
FOREX MARKETS	0	0	0	0	18	0	7	23	0	0	0
HEALTH	0	0	0	0	0	8	4	0	2	0	0
IRRELEVANT	0	1	1	0	4	3	237	15	0	2	3
MONEY MARKETS	0	0	0	0	17	0	8	44	0	0	0
SCIENCE AND TECHNOLOGY	1	0	0	0	0	0	2	0	0	0	0
SHARE LISTINGS	0	0	0	0	0	0	4	0	0	3	0
SPORTS	0	0	0	0	0	0	2	0	0	0	58

References

Kim, C. (2018, February 23). Retrieved from https://medium.com/machine-learning-intuition/document-classification-part-2-text-processing-eaa26d16c719

Madan, R. (2019, November 18). Retrieved from https://medium.com/analytics-vidhya/decisiontree-classifier-working-on-moons-dataset-using-gridsearchcv-to-find-best-hyperparameters-ede24a06b489

Muller, A. Guido, A. (2017) *Introduction to Machine Learning with Python, A Guide for Data Scientists,* O'Reilly Media Inc

Sanikamal. (2018, December 2). Text Classification With Python and Keras. Retrieved from https://www.kaggle.com/sanikamal/text-classification-with-python-and-keras

Zafra, M. (2019, June 15). Retrieved from https://towardsdatascience.com/text-classification-in-python-dd95d264c802

Kapadia, S. (2019, March 27) Language Models: N-Gram. Retrieved from: https://towardsdatascience.com/introduction-to-language-models-n-gram-e323081503d9