School of Computer Science and Engineering
Faculty of Engineering
UNSW Australia

Predicting Relevant News Articles Based on Topics

Using Machine Learning Techniques

By

EvansCrew2.0+QuarantineClub

Adam Stucci z5157372
Byron Chen z5210309
Jing Jing Fan z5214546
Peter Nguyen z5061984
Vivian Shen z5214750

Group Report submitted as requirement for
COMP9417: Machine Learning and Data Mining

Trimester 1 2020
Dr. G. Mohammadi

1. Introduction

Advancing into the digital age, the internet can provide users with an unlimited amount of information, choices and selections. To alleviate the stress and decrease the time an user might take to find relevant articles, a recommendation system can be implemented using machine learning techniques to efficiently and effectively provide the user with the most relevant articles.

The primary purpose of this research is to determine the most appropriate modelling technique to recommend a set of news articles to an user with an interest in a particular topic. The recommendation system should not suggest news articles that are unlikely to be relevant to the user's interest if there are less than 10 recommended news articles.

In order to achieve this, multi-class text classification will be performed using various different machine learning methods including Naive Bayes, Random Forest, Support Vector Machines and Logical Regression.

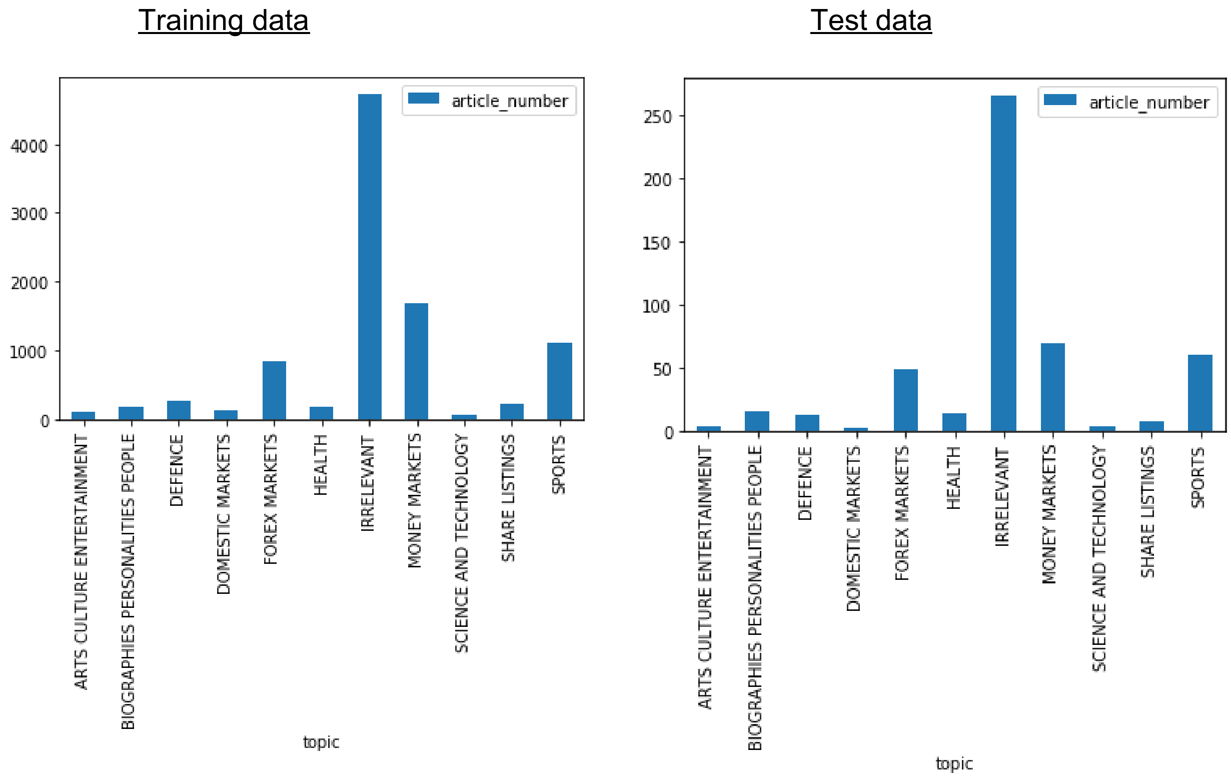
1.1 Exploratory Data Analysis

Prior to modelling, we conducted a basic Exploratory Data Analysis (EDA) to handle any data cleaning and to also examine any statistical complications.

Our training dataset consisted of 9500 categorical data with unique news article numbers and a selection of words from that article. Each article was either related to the user's topic of interest or was an irrelevant article (article that did not belong to any of the user's topics of interest). Additionally, our testing dataset consisted of similar 500 categorical data which followed a similar probability distribution with our training dataset which is important for classifying data, to eliminate additional factors when comparing model performances on training and test datasets.

Upon inspection, both our training and test datasets were not missing any values from any columns, and all our data within each column was consistent as all article_numbers were integers, all article_words were a list of words delimited by a comma and all topics are as listed in Figure 1 with the using the exact spelling and characters. Whilst there were words in article_words that were incomplete English words such as commun and also non-English words such as hoechst, we found they were either consistently incomplete or misspelt throughout the entire file. Overall, this means that there was no need for additional data cleaning on our behalf and our models in mind have to account for some form of text classification.

Figure 1. News Article Distribution on Topics for Training and Test Datasets


Training data

Test data

A more detailed breakdown of this distribution can be found in Appendix Figure 1.

In Figure 1 we can see that 48% of the training data consisted of news articles associated with “irrelevant” topics, this may create issues such as misclassifying articles as “irrelevant” instead of it’s true topic. However, it can also serve a positive function as generally common words (regardless of article) can be identified more easily.

From Figure 1, we found that there was not a uniformed distribution of data, some topics have a larger list of articles whilst some topics are extremely limited. This is important to note since increased distribution of news articles for a particular topic means that our model is better informed on how to classify certain categories. We predict that our model would be able to classify news articles in our test dataset for Money Markets topic more accurately compared to articles with the Science And Technology topic. We can see that these smaller samples are harder to train within the training set, but are also under-represented in the testing set comparatively. This is likely to create issues within testing as a slight misclassification can severely impact the results, and is more likely to occur as it is harder to learn on without weighting.

2. Methodology

2.1 Experiment Design

We began by performing EDA to gain insight into this particular datasets properties, identifying potential limitations and pitfalls with certain algorithms and other choices. Following this, we conducted literature reviews to gain insight into how different algorithms and models tend to perform on similar problems. Using our theoretical basis and prior experimental results to inform our model justifications and construction. As performance is highly problem and data specific we will build several models to test them and examine their ‘performance’ under criteria that fits the problem the best.

2.2 Feature selection

TF-IDF scoring method was chosen as this metric weighs down words that are common throughout multiple articles as they have little discriminating power and weighs up words that are more specialised in a particular article. This is particularly important for our dataset since article_words has words that are common throughout multiple articles as well as words that are only used in a specific article.

Each unique word was set as a separate feature and ‘topic’ was set to be the target. Then a TF-IDF value was computed for each word with respect to the article that it is in. Each TF-IDF value is a measurement of how relevant the word is in a specific document. It is calculated using this formula.

Figure 2. TF-IDF Value Calculation

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

$tf_{i,j}$ = number of occurrences of i in j
df_i = number of documents containing i
N = total number of documents

Thus, this is helpful since if an article with similar “relevant” words as an article that has already been fitted was encountered, it can easily be deduced that it will most likely belong to the same topic as the fitted article.

Another key feature of TF-IDF is that it normalises the distribution of words, i.e. The relevancy between a word and a topic is not solely dependent on how frequently that word appears in articles of the mentioned topic, successfully resolving another complication in our dataset.

2.3 Dealing with Noise

Some discussion developed over whether articles categorised as ‘Irrelevant’ should be considered as ‘noise’. Especially since Irrelevant topics account for the large proportion of our dataset, and might misclassify articles as “irrelevant” instead of it’s true topic. However, the capability to predict an article as ‘Irrelevant’ was also necessary. Any attempt to increase prediction accuracy by reducing the weight of irrelevant articles would be overfitting the model. Based on this criteria, it was decided that irrelevant articles would be treated as another topic as opposed to noisy data.

2.4 Machine Learning Models

These following models were chosen and explored as they supported text classification well and also performed well with large quantities of data. The results of these exploratory models were ranked and later used to determine which model suggested the ‘best’ top 10 news article ranks.

2.4.1 Naive Bayes

The Naive Bayes (NB) classification performs well on categorical data and is commonly used for text classification and often outperforms other models.
The NB model calculates the posterior probability of the given article to be associated with all possible topics. The topic with the highest posterior probability classifies the particular article with that topic.

Since NB is a probabilities classifier, these probabilities can also be used to help rank the news articles and consequently suggest the ‘best’ top 10 articles. Additionally, assuming that attributes have an equal and independent contribution to the model, NB enables low computational costs even with a large training set. This is applicable in the context of this project given the nature of TF-IDF.

2.4.2 Random Forest

Decision Tree classifiers create a system of class separation points based on key data points or features presented in a database. Predictors essentially ‘flow down’ the tree on branches determined by these separation points until they reach the leaf node that determines their classification. A random forest model (RF) builds individual decision trees which splits the data based on feature, generating a random subset of features. This split is chosen randomly which ensures accurate variable rankings even if there is a strong correlation in our dataset.

This model is useful as it can handle categorical features and a lot of data with a low computation cost since we only consider a subset of features. As we only use a subset of features, this will allow us to diversify in the trees. From the RF, the topic the news article is associated with is chosen by taking the average over all trees.

2.4.3.a Support Vector Machines

Support Vector Machine models (SVM) aims to classify our news articles by finding a hyperplane that maximises the distance between the data points of our topics on 10 dimensions (10 topic features). An SVM model is a classification method as SVM works well in high dimensionality and for cases where we find linearly separated data, which is commonly the case for text classification.

Whilst SVM training time for larger data sets are slow, with a sparse data set like ours, the time can be decreased dramatically. Here, the SVM will generate hyperplanes to classify the news articles into their respective topics:

2.4.3.b Stochastic Gradient Descent

The Stochastic Gradient Descent (SGD) allows for the discriminative learning of linear classifiers such as the Linear Support Vector Machines and the Logistic Regression classifier. It iteratively does a stochastic approximation of gradient descent optimisation replacing the actual gradient by an estimate from randomly selected data subsets training the Linear Classifiers.

This is useful for large scale data sets which are reflective of the training and test data set and since it only uses a subset of the data which does not require kernel methods.

2.4.4 Logistic Regression

Logistic Regression is a linear classification model whose predictive model is based on a logistic (sigmoid) function. Training data is mapped on a range of 0 and 1, and from this a logistic function, scaled by the values of present features, is fitted to describe the probability of either ‘true’ or ‘false’ based on an input.

The main advantage of using Logistic Regression is how informative it can be. Among other things, Linear Regression distinctly measures the relevancy of a predictor. This makes it a simple task to identify the most relevant articles for each topic, more efficiently, and definitively in comparison to other models. This model may be best for finding correlations between features, but suffers in accuracy and efficiency.

2.5 Hyper-parameter

The classifiers that had the most ideal classification time in terms of performance were then set up for hyper-tuning to determine the parameters which resulted in the best accuracy values. The classifier model parameters were loaded into GridSearchCV to find the best estimated classification model with the best mean test score which is the accuracy. Only two models were able to show significant increase in accuracy from hyper-tuning within a reasonable time, the Multinomial Naive Bayes classifier and Stochastic Gradient Descent classifier.

For the Stochastic Gradient Descent (SGD) classification model, the shuffle parameter was initially set to false to consistently maintain the high accuracy value that would’ve otherwise returned varying accuracy scores of lower values than the maximum values obtainable. The following parameters that actively affected the accuracy values were iterated over to find the best score in the GridSearchCV operation.

For the hyper-parameters, the following features were selected for hyper-tuning as they affect the accuracy scores for the stochastic gradient descent classifier:

1. The stopping criterion (tol):
    - This parameter helps changes the values as the tolerance dictates when the iterations will stop when the hinge values is greater than the best loss value subtracting the tolerance value which affects the final accuracy scores
2. The learning rate (learning_rate):
    - The parameter dictates the rate of the learning rate of the classifier be able to classify the words at the optimal level within the iteration time
3. The initial learning rate for 'constant' (eta0):
    - The constant starting rate for the learning parameter to determine how fast the classifier will perform at the start of the training of the data

Figure 3. Stochastic Gradient Descent (SGD) Hyper-Tuning

Parameters	**Values iterated over**
The stopping criterion (tol)	1e-1, 1e-2, 1e-3, 1e-4, 1e-5, 1e-6, 1e-7, 1e-8
The learning rate (learning_rate)	constant, optimal
The initial learning rate for 'constant' (eta0)	0.05 - 0.15 (in increments of 0.01)

2.6 Pipelining

The pipeline was established to determine which classification model was the best to perform the final classification of the topics with respect to the words in the respective articles. The metrics that were used to determine which metric had the best performance is determined by each of the classification's accuracy score. The classification models that had optimal classifying capabilities were all placed within their own respective pipelines that is setted up with all their classification models with the parameters that was previously hypertuned.

The list of pipelines of each of the classification models that is evaluated is then created. The pipes in the pipelines are iterated through where the model of each pipe is fitted. Then the model of the most accurate classifier in the pipeline, with their hyper-tuned parameters, is returned ready to be used in determining the top recommended articles for each topic.

2.7 Evaluation Metrics

The main evaluation metric used between models was the accuracy score which was a crucial measure for evaluating how effectively the classification model performed while training the dataset.

Precision measures the classification model's ability to detect the ratio of True Positives to True and False Positives. Recall measures the classification model's ability to accurately predict positive instances (True Positives and False Negatives). F1 Score is commonly used to compare classification models' performance as it takes the weighted harmonic mean of precision and recall ranging from 1.0 as the best and 0.0 as worst. Support indicates the number of actual classifications of the dataset.

The top 10 articles for each topic was selected by computing a calibrated predicted probability score for each article with respect to the topics. The test set is used to calibrate the probabilities This is done by using k-fold cross validation to return the average of the k-folds which is representative of the predicted probabilities.

3. Results

Figure 4. Accuracy Score Comparison for all Models

	Multinomial Naive Bayes	**Random Forest**	**Support Vector Machine**	**Logistic Regression**
Training accuracy score	0.825	0.989	0.871	0.812
Test accuracy score	0.752	0.74	0.79	0.746

Figure 5. Final classification model results and hyper parameters used

Stochastic Gradient Descent Classifier (SGD) with Hyper-Tuning using the 5-fold cross-validation grid-search over the parameter grid of the training and feature sets:

Parameters	**Final Hyperparameters**
The stopping criterion (tol)	1e-5
The learning rate (learning_rate)	optimal
The initial learning rate for 'constant' (eta0)	0.05
Accuracy Score (test):	0.79
Accuracy Score (train):	0.871

The training data has a higher accuracy (by 8%) than the test data but not to the extent of overfitting with faster run-time over large amounts of data set.

Figure 6. Classification Report using Stochastic Gradient Descent Classifier (SGD)

Topic name	**Precision**	**Recall**	**F1**	**Support**
ARTS CULTURE ENTERTAINMENT (ACE)	0.33	0.67	0.44	3
BIOGRAPHIES PERSONALITIES PEOPLE (BPP)	1.00	0.20	0.33	15
DEFENCE	0.88	0.54	0.67	13
DOMESTIC MARKETS	0.50	0.50	0.50	2

FOREX MARKETS	0.41	0.27	0.33	48
HEALTH	0.69	0.64	0.67	14
MONEY MARKETS	0.52	0.68	0.59	69
SCIENCE AND TECHNOLOGY	0.00	0.00	0.00	3
SHARE LISTINGS	0.40	0.29	0.33	7
SPORTS	0.95	0.97	0.96	60

Figure 7. Classification Report for List of Suggested Articles using Stochastic Gradient Descent (SGD)

Topic name	**Suggested articles**	**Precision**	**Recall**	**F1**
ARTS CULTURE ENTERTAINMENT	9830, 9952, 9789. 9703. 9933. 9526	0.33	0.67	0.44
BIOGRAPHIES PERSONALITIES PEOPLE	9940, 9988, 9878	0.80	0.27	0.40
DEFENCE	9616, 9559, 9842, 9670, 9576, 9773, 9770, 9607	0.88	0.54	0.67
DOMESTIC MARKETS	9994, 9640	0.333	0.50	0.40
FOREX MARKETS	9551, 9588, 9682, 9632, 9798, 9986, 9772, 9786, 9529, 9671	0.50	0.10	0.17
HEALTH	9661, 9873, 9833, 9926, 9929, 9947, 9609, 9621, 9911, 9978	0.70	0.50	0.583
MONEY MARKETS	9618, 9871, 9755, 9761, 9998, 9835, 9769, 9602, 9840, 9707	0.70	0.10	0.18
SCIENCE AND TECHNOLOGY	9617, 9982	0	0	N/A
SHARE LISTINGS	9518, 9601, 9715, 9666, 9668	0.60	0.43	0.50
SPORTS	9857, 9760, 9848, 9922, 9569, 9574, 9997, 9849, 9787, 9887	1.00	0.17	0.286

3.1 Result Evaluations

From Figure 4, we see that whilst RF had the highest accuracy for training dataset, SGD tuned from SVM performed the best on test dataset of interest by at least ~4% compared to the other models.

As expected, topics that had small numbers of training and test samples were not able to be predicted correctly as the mean value and estimated probabilities are not as close to their true value. Furthermore, we also notice problems amongst classes with very similar predictors, e.g

Forex Markets and Money Markets as their sets are quite close in the feature space. Thus distinguishing between them is rather sensitive to the predictors in a document.

Unexpectedly, our precision for BPP was the highest whilst only having a smaller pool of 15 articles. This contradicts our priori expectation that topics with higher support values such as Money Markets should have higher precision rates. However, Sports, which has the 2nd support values and precision rates and similarly for ACE and Science & Technology still aligns with our priori expectations. This would indicate that selected keywords summaries played a larger role in helping our model predict topics compared to the number of articles it gets trained on.

Higher precision values are extremely valuable to us in this case since False Negatives are excluded, suggesting a possible loss of information for our user since these articles will not be included in our recommendation. We found that higher precision values in topic classification led to higher precision for article rankings, which is expected. Our model generated higher precision values in article rankings which is expected with the exception of Domestic Markets whose low support value can account for this discrepancy in results. Ultimately, the precision rate for article rankings is more relevant to our findings since our aim is to create a recommendation and hence it is more important to recommend articles closer to the suggested articles as opposed to ensuring that we are classifying our articles by topics correctly.

Low recall values are ideal since our cost of False Negative is high as it results in information loss for the user. Here, most of our recall values are below 0.5 for classification, however, our recall values for article suggestion are comparatively lower, indicating that the information loss for the users is not as dire as initially thought.

Since F1 indicates that our best prediction is for the topic Sports and our worst prediction is for Science & Technology which can be explained by the low support value.

3.2 Feature Importance

Words in the training data were transformed into a vector representation with each unique word as a feature. Therefore features with the most importance are the words that carry the highest weight (ie TF-IDF value). These words usually distinguish the topics from another as they only frequently appear in particular articles. Features that are less important are common words found through multiple articles as they have little power to distinguish between topics

Further statistical measures like mutual information and chi2 were used to assess feature importance to each class label. Domain specific terms contributed the most to class differentiability. However from trialing different values of using the best k features, model performance was typically worse if not the same. This suggests that even features with low information are informative enough to be useful in classification.

4. Discussion

4.1 Method Comparison

Hyper-tuning

Compared to SGD classifier, the pipelining of the MNB Classifier was not as effective since the hyper-tuned CountVectorizer and TfidfTransformer was instantiated for the data set in the code that primarily performed the solution with the MNB Classifier whilst the code with the pipeline used the non-hyper-tuned TfidfVectorizer that was initialised. However, the SGD Classifier would have performed better than MNB Classifier regardless which was used to return the classification report and return the top 10 articles anyway.

Naive Bayes Classifier

The baseline model of this classifier is a lot less accurate than other discriminative methods. Naive Bayes also has nearly no hyper-parameters to tune. After tuning the features, Naive Bayes was much more competitive than previously and had the second best accuracy score amongst our tested models.

Naive bayes is a simpler model and tends to have a low variance and is quick to compute and thus is a suitable model. Compared to the other methods, our Naive Bayes Classifier obtained higher F1 and precision values compared to RF and Logistic Regression and relatively lower values compared to the SGD model.

As we care about giving the user relevant articles with little noise, the precision is particularly important. Naive Bayes tended to perform poorer here compared to the SGD classifier when there were more samples available for the class. However, when the class had less samples available, Naive Bayes tended to have better precision.
With this tradeoff we ultimately elected the model with higher accuracy i.e. the SGD classifier.

Support Vector Machine

Support Vector Machine usually has high training time for large data sets. However, with the usage of Linear SGD Classifier, it has significantly faster training time with very low performance requirements that returns one of the higher accuracy values compared to other methods, given that it has the right hyper-tuned parameters.
Hyper-parameter tuning has proven to be a significant time investment and challenge to correctly produce optimal results. A large portion of time was consumed manually adjusting parameters and identifying which of those exhibited active effects on the score.
SVM scales worse with noise. As such, given the huge number of words whose relevance overlaps across multiple topics, there may be inaccuracies caused by variance.
SVM also does not natively calculate probability estimates. This can be remedied with option methods, however this incurs a significant time-cost. This disadvantage diminishes our capability to reliably identify the most relevant articles for each topic.

Random Forest

Despite the fact that random forest is designed to improve with overfitting, our results showed that it still overfitted more compared to the other models as test data has an accuracy of 73% whilst train data has accuracy of 98%. Decision trees within the random forest will find it to determine splits of words if they are commonly shared between a number of articles of different categories. Additionally, although RF greatly improves the performance of Decision Trees, it still correlates badly with a large number of features. In the context of this project, where each unique word is a feature, the costs of the model are significant.

4.2 Metrics

High precision rate is an appropriate metric to use in the suggestion of articles as this is the only thing the user will be presented with. As a result, the article recommendation is more important than ensuring that the classification of our articles were correct. Additionally, we also valued a lower recall value for our suggested articles. This is appropriate as False Negative would suggest potential loss of information for the user. To ultimately determine which model would be a better fit for our recommendation system, we compared the accuracy of the test dataset and F1 score as it takes the harmonic mean of our precision and recall values.

4.3 Complications and Future Improvements

The GridSearchCV function that was used for hyper-tuning is very computationally expensive with $O(n^{\text{no. of features}})$ time complexity. Therefore not all ranges of hyperparameters could be tested and the final chosen hyperparameters may not have been the most optimal. To improve upon this, better hyper-tuning methodologies could have been used instead. The quantities of values and types of parameters to test could also be cut down through more extensive research and experimentation.

The article distribution between topics are imbalanced, meaning that a significant word that distinguishes a topic may appear less due to there being less articles of that topic. Using tf-idf would undermine the significance of such unique words. A way to improve upon this would be to calculate weights for each word dependent on its frequency throughout topics rather than through the articles. Another way to improve this is to weigh the words in topics with limited articles higher in an attempt to 'balance out' the distribution. We could also try oversampling minority classes and undersampling majority classes.

The model can be also improved by considering only the most informative features.
As the current matrix representation of the features have extremely large dimensions. One method for doing this is estimating common information for each feature with respect to the topics to measure the informativeness of the feature to the class label and then only using the top K features to fit the model. We could also consider more accurate weighting metrics other than TF-IDF.

5. Conclusion

We were able to successfully develop a modelling technique that recommends the top 10 best articles for an user with an interest in a particular topic. This was done through transforming article words into a representation of numerical vectors then fitting it with various models. Best models were then hypertuned to produce better results. Despite running into many obstacles, we discovered many different models and methodologies suitable for text classification with the SVM model using SGD classifier being the best. This had the best performance with our final SGD model obtaining a 0.79 accuracy score on our test dataset and the highest precision and F1 values. However, there are still many ways we could have improved upon our methodologies which we can take into consideration for future reimplementations of the classification model solution.

6. Reference

- https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/
- https://www.kdnuggets.com/2018/08/wtf-tf-idf.html
- https://towardsdatascience.com/natural-language-processing-feature-engineering-using-tf-idf-e8b9d00e7e76
- https://towardsdatascience.com/an-implementation-and-explanation-of-the-random-forest-in-python-77bf308a9b76
- https://towardsdatascience.com/machine-learning-nlp-text-classification-using-scikit-learn-python-and-nltk-c52b92a7c73a
- https://medium.com/datadriveninvestor/an-introduction-to-grid-search-ff57adcc0998
- https://muthu.co/understanding-the-classification-report-in-sklearn/
- https://elitedatascience.com/overfitting-in-machine-learning#how-to-detect
- https://medium.com/@manoveg/multi-class-text-classification-with-probability-prediction-for-each-class-using-linearsvc-in-289189fbb100
- https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74
- https://towardsdatascience.com/a-guide-to-svm-parameter-tuning-8bfe6b8a452c
- https://towardsdatascience.com/how-to-make-sgd-classifier-perform-as-well-as-logistic-regression-using-parfit-cc10bca2d3c4
- https://chrisalbon.com/machine_learning/naive_bayes/calibrate_predicted_probabilities/
- https://scikit-learn.org/stable/modules/sgd.html#classification
- https://en.wikipedia.org/wiki/Stochastic_gradient_descent
- https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html
- https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9

7. Appendix

Appendix A. Detailed Percentage Distribution of Training and Test Dataset

	A,C&E	BPP	Defence	DM	FM	Health	MM	S&T	SL	Sports	Irrelevant
Train	1.3%	1.4%	2.7%	1.4%	8.8 %	1.9%	17.6 %	0.74 %	0.74 %	11.6%	48%
Test	0.6%	0.4%	2.6%	0.4%	9.6	2.8%	13.8	14%	1.4%	12%	52%

Appendix B.

	Multinomial Naive Bayes			Random Forest			Support Vector Machine (SGD)			Logistic Regression		
Topic Name	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Art Culture Entertainment	1.00	0.67	0.80	1.00	0.33	0.50	0.33	0.67	0.44	0.00	0.00	0.00
Biographies Personalities People	0.80	0.27	0.40	0.00	0.00	0.00	1.00	0.20	0.33	0.00	0.00	0.00
Defence	1.00	0.31	0.47	1.00	0.23	0.38	0.88	0.54	0.67	1.00	0.46	0.63
Domestic Markets	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.50	0.50	0.00	0.00	0.00
Forex Markets	0.45	0.10	0.17	0.56	0.19	0.28	0.41	0.27	0.33	0.48	0.31	0.38
Health	0.83	0.36	0.50	0.00	0.00	0.00	0.69	0.64	0.67	0.50	0.07	0.12
Money Markets	0.50	0.87	0.64	0.57	0.74	0.64	0.52	0.68	0.59	0.54	0.65	0.59
Science & Technology	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Share Listings	0.00	0.00	0.00	0.00	0.00	0.00	0.40	0.29	0.33	0.50	0.14	0.22
Sports	0.94	1.00	0.97	0.92	0.92	0.92	0.95	0.97	0.96	0.94	0.97	0.95