Faculty of Engineering

COMP9417 Machine Learning and Data Mining

2020 Term 1

Project

Group member

ZID	Name
Anant Krishna Mahale	z5277610
Hongyi Gao	z5183218
Nan Wu	z5238997
Raghavendran Kasinathan	z5284284
Yeqin Huang	z5175742

Contents

1. INTRODUCTION

This project is about text classification. As the description of the project specification. The task is using 9500 classified articles to predict the classes of other 500 articles. Also, there are 10 topics that each one requires 10 most relevant articles from the 500 articles. In this project, we used 4 models to accomplish the classification task and compared their accuracy results, they are Linear Support Vector Machine classifier, Multinomial Naïve Bayes classifier, Multi-layer Perceptron learning, and Random Forest with SMOTE classifier. We choose the one with highest precision and recall, which is Random Forest along with SMOTE technique, to classify and suggest 10 relevant articles. The challenge in this project is the unbalanced distribution of categories number of articles, and we get an ideal result by applying SMOTE (Synthetic Minority Over-sampling Technique) to deal with problem.

Our group name is: Renamed_Group_42, and we 5 members in our group, the group members information could be seen from the cover page. All the tables and graphs are list in the Appendix.

2. DATASET

2.1 Training set

The data given to build the machine learning topic has three columns with following data types. The ***article_words*** includes all the pre-processed key words in one article, and the ***topic*** is the class of that article. The ***article_number*** is just an indexing column to return most relevant articles. We need to make use of ***article_words*** and ***topics*** to train the machine learning model.

Given 9500 articles and relevant categories, we need to understand the distribution based on the topics. To do that we apply the counter. Pictures speaks better than words. Hence, to better understand the distribution, we plotted a pie chart, and this can be seen from **figure 1.**

We found that the distribution of categories in the dataset is not average, and irrelevant topis take over half of the articles, following topic money markets. Science and Technology is the least one, which only take over 1% of all the articles.

To understand better, we checked the most common words across all the topics, they are shown in figure 2.

The distribution of key words in the articles is not average, either. The most common one is 'percent', following 'year', 'trad', and 'dollar'.

2.2 Test set

This is an unseen data that will be used to suggest most relevant articles to the users using the Machine Learning model built using Training dataset. The same with training data, also have ***article_number***, ***article_words*** and ***topic*** attributes. Test set is used for the prediction of the articles that only have 500 instances. Some of the topics that in test set only has limited number of articles. For the requirement of 10 articles for each topic, there would be some topics that have less than 10 articles. For example, for topic 'Arts Culture Entertainment', there is only 3 articles that belong to this topic.

3. Methods

3.1 Models

For this project, we selected four models for classification. They are Linear Support Vector Machine classifier, Multinomial Naïve Bayes classifier, Multi-Layer Perceptron learning, and Random Forest Classifier. We have provided code for each of the model discussed in the Code Submission part (in Discussion_codeFiles).

For each model, we use accuracy, recall, and f1-score to justify their performance. Except that, cross-validation value is also considered as one standard. Other than that, the macro average and weighted average are applied to measure the performance for prediction in different topics.

3.1.1 Linear Support Vector Machine Classifier

SVM (Support Vector Machine is a classification method that once defeated neural networks. Since the late 1990s, it has been an important application in many fields. In recent years, due to the rise of deep learning, the usage of SVM began to decline, but it is still a classic classification method. Chervonenkis [1] noticed that many of the ideas now being developed in the framework of Support Vector Machines were first proposed by V. N. Vapnik and A. Ya. Chervonenkis (Institute of Control Sciences of the Russian Academy of Sciences, Moscow, Russia) in the framework of the "Generalised Portrait Method" for computer learning and pattern recognition. The development of these ideas started in 1962 and they were first published in 1964. The common method implemented today is proposed by Cortes and Vapnik. [2] They applied the Lagrange duality to solve the constrained optimization problem in the SVM problem.

In this project, we used linear SVM to make classification of the train sets and make predictions. To make it simple, we take two classification as example, as shown in the figure 5, let the training set be D blue dots as one type, and the red squares as another type. The goal of classification is to find a hyperplane to separate the two types of data. In the two-dimensional plane, the classification hyperplane is a straight line. As is shown in the figure, here are many possibilities for the hyperplane that can separate the training samples (green dotted line in the figure). In addition to separating the data in the training set, the hyperplane also needs to have better generalization performance and the data in the test set needs to be divided. Intuitively, the solid green line is a better division, because the straight line is farther from the two types of data points, and it is more tolerant to local disturbances in the data and can classify the data with greater confidence.

Therefore, the hyperplane with the largest distance between the two types of data points is the best classification. The margin between the two types of data points from the hyperplane is as shown in the figure 6. Assume that the expression of the two dashed lines in the figure is $\omega^{\top}x + b = -1$ and $\omega^{\top}x + b = 1$. So, the expression of middle classification line is $\omega^{\top}x + b = 0$. For the convenience of calculation, the labels of the two types of data are set to blue dots (y = 1) and red squares (y = -1). If the classification hyperplane can correctly classify the two types of data, we can get the constrained expression: $\begin{cases} \omega^{\top}x_i + b = 0 & y_i = 1 \\ \omega^{\top}x_i + b = 0 & y_i = -1 \end{cases}$. And the sum of the distance between the two types of data to the hyperplane, that is, the margin is $\frac{2}{||\omega||}$.

Eventually, it becomes a constrained optimization problem. Under the constrained expression, we need to get the max margin. Cortes and Vapnik [2] proposed that the Lagrange duality can efficiently solve the problem and will always find the optimal result.

For this project, we set all the hyper-parameters for linear SVM as default, but for the model selection part, the training and test split of the training set with 'random state=42'.

3.1.2 Multinomial Naïve Bayes

As for this project, we need to implement a text classifier, which is a discrete features classification task, and Naïve Bayes is an ideal algorithm to deal with this question.

We learn the multinomial Naïve Bayes in week 3 from the lecture. For supervised learning algorithm Naïve Bayes classifiers, probability is used on the classification. The Bayes rule can be described as, [4]

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

If the target is V, and the target fuction is $f: X \to V$, then the distances x described by attributes is $< x_1, x_2, \dots, x_n >$.

The attribute with the highest probability would be predicted result. Which can be described as,

$$v_{MAP} = arg \max_{v_j} P(x_1, x_2, \dots, x_n \mid v_j)P(v_j)$$

For Naïve Bayes, the probability of $P(x_1\ , x_2, \dots, x_n |\ v_j)$ is described as,

$$P(x_1\ , x_2, \dots, x_n |\ v_j) = \prod_i P(x_i|v_j)$$

Then the Naïve Bayes classifier can be described as, [5]

$$v_{MAP} = arg \max_{v_j} P(v_j) \prod_i P(x_i|v_j)$$

As for the real world has many situations that the values of features are not consecutive, rather discrete. Then the multinomial classifier is the one that could be applied on these questions.

For multinomial Naïve Bayes classifier, the features in the attributes would be discrete and that for some attributes, the features quantity could be 0, in order to solve this problem, the Laplace smoothing is introduced, [6]

$$\hat{\theta} = \frac{x_i + \alpha}{N + \alpha d} \qquad (i = 1, \dots, d)$$

α is the smoothing parameter, and for most situations α would equal to 1.

For this project, we already have the split training and test dataset. The training set and test set have the same attributes, ***article_number***, ***article_words***, and ***topic***, for the model selection process, we chose 11 topics as the features, and ignored the ***article_number***, as it is not useful for temporary.

The beneficial of multinomial Naïve Bayes is that it could be applied on most text classification tasks, it is easy to debug and check the result. When the distribution of the dataset is average, the performance of it could be ideal. However, the disadvantage for it is when the distribution for the dataset is not average, the classifier could be confused and make the wrong prediction. Then for multinomial Naïve Bayes classifier, dealing with dataset that without average distribution could be a problem.

For this project, the hyper-parameters of MNB are also set as default.

3.1.3. Multi-Layer Perceptron Classifier

A perceptron takes a single or multiple input including a bias and an activation function to produce a single output. It multiplies the inputs by some weight, and then passes them into an activation function to produce an output. This model can be seen from figure 7.

In our case, the inputs are the text words that are to be classified based on some logarithmic functions to produce the article label output.

Once we run the perceptron to produce the output Article label, we can compare it to a known Article label from the training set and adjust the weights accordingly. We keep repeating this process until we have reached a maximum number of allowed iterations which is around 100 in our case, or an acceptable error rate.

In this neural network, we simply begin to add layers of perceptron together, creating a multi-layer perceptron model of a neural network, which is implemented in the SKLEARN NLP CLASSIFIER LIBRARY.

Once we chose MLP for classification, we had to form the Count Vector out of the text input keywords that are given for each article label. With this count vector as input to the Neural Network, they continuously iterate in-between the hidden layers to get the Output Article label based on some weighted score-based processing.

The layers of the neural network are what makes the Machine learning model unique. We could keep it short by reducing the number of iterations but for better accuracy, it is best to keep an optimal number of iterations minimum.

3.1.4. Random Forrest Classifier

The shortness of all the above models are obvious, if the samples of one category is much less than others, the accuracy of that class would be less than others. This would decrease the performance for the whole dataset. Even for the other algorithms, the unbalanced distribution would lead to a failure on classification. To solve this problem, we used SMOTE technique to transform the dataset. In that way, each word would be considered into all the categories, not like before that for each class, the classifier only considers all the words appeared in that class.

Synthetic Minority Over-sampling Technique (SMOTE) is a method that enlarge the minority dataset to enhance the performance of model. Oversample is originally used in signal processing. For data science, it is used to enlarge the minority dataset to balance the wrong prediction possibility. [7]

For this project, Random Forest Classifier with SMOTE applied on the Training Dataset. This is a method that shuffle all the dataset in the minority part and enlarge it with random order data in the original one. [8]

Random Forest is using Bagging method of ensemble learning. For this algorithm, it does not rely on the previous tree, which means each tree is independently. It is the decision tree been replaced with random subset of features. It usually cost low and calculate efficient. We combine Random Forest and SMOTE together and got a better performance. [9]

To improve the performance of the model, we reset some hyper-parameters. That is,

```
RandomForestClassifier(bootstrap=True,ccp_alpha=0.0,class_weight
=None,criterion='gini',max_depth=None,max_features='auto',max_le
af_nodes=None,max_samples=None,min_impurity_decrease=0.0,min_imp
urity_split=None,min_samples_leaf=1,min_samples_split=2,min_weig
ht_fraction_leaf=0.0,n_estimators=100,n_jobs=None,oob_score=Fals
e,random_state=42,verbose=0, warm_start=True)
```

4. Results

4.1 Linear Support Vector Machine

There is one built in function in ‘sklearn’ which named ‘cross_val_score’. We used this function implemented the cross validation.

From the table 1, we can see the metrics of each topic on the test set. The accuracy of the whole test is 0.724 which is acceptable. It performs better when the number of the topic is big, such as IRRELEVANT and SPORTS. They got 0.83 and 0.91 for f1-score respectively, which means the classification of these two topics are effective when we are doing classification. However, the prediction is not reliable for some topics which has less number in the test sets, especially for the SCIENCE AND TECHNOLOGY, which got triple 0 in the linear SVC model.

The cross-validation value is 0.72. This is an acceptable value for us to believe the prediction result.

The macro average is around 0.48 for precision value, for recall and f1-score, that is around 0.35, and for weighted average, the result is around 0.7. This reveals that for SVM, it performs well with majority topics, but poor for minority topics.

4.2. Multinomial Naïve Bayes

The metrics of each topic on the test set for multinomial Naïve Bayes can be seen from table 2. For multinomial Naïve Bayes, the performance has decreased compared to linear SVM, and for Domenstic Markets, the prediction is still wrong for each one, also for Science and Technology. For most topics, the prediction accuracies are slightly lower than linear SVM, the same with recall and f1-score values.

The cross-validation value for this model is 0.74, better than linear SVM. The macro average is 0.5, and weighted average around 0.75. This means the model performs average on each topic.

4.3 Multi-Layer Perceptron Classifier

Table 3 shows the metrics of each topic on the test set for multi-layer perceptron classifier. For SIENCE AND TECHNOLOGY, all the articles are wrongly classified. But for topics with large amount of articles, the performance of MLP is good, it seen that for SPORT, the precision, recall and f1-score values are all achieved 0.95, which means for 60 articles, there are 57 of them are classified correctly. The accuracy for MLP is 0.75, which is a good model in this aspect. But would take a long time to calculate.

The cross-validation value for MLP is 0.98, which is the highest among all the models. For macro average, it is around 0.55, and for weighted average, it is around 0.75. Half and Half good and bad performance for single topic.

4.4. Random Forest Classifier

Finally, we chose random forest classifier along with SMOTE technique to classify the articles under each topic. The metrics of each topic on the test set for this model is shown on table 4, it can be seen that only DOMESTIC MARKETS got no article classified correctly, and for the rest of the topics , each topic has at least some articles are classified correctly. This might due to the amount of DOMESTIC MARKET is too little, only 2 articles are here. The topic is hard to be classified with FOREX MARKETS and MONEY MARKETS. The average accuracy is 0.75, which is the best among all the models.

Before we developed the RFC with SMOT, we did tried RFC without SMOT, and the results were not appealing, we do not write it on this report because we think that is not the key point to describe. But the RFC with SMOT performed the best among all other models with SMOT, this is what we did not expected.

The suggested article index is shown in table 6. For we do not wish users to read articles that they are not interested in, there are some blanks on the table. To get the article list, we checked the recall value of RFC with SMOT, and then use the value to multiple with the accuracy to generate the threshold. The threshold is set as table 7 shows.

The cross-validation value is 0.75, macro average is around 0.57, and weighted average around 0.75. This is slightly better than MLP.

The selection of RFC of SMOT is based not only the precision accuracy, but also the cross-validation value and macro, weighted average value. Due to RFC with SMOT achieved the highest precision accuracy and second cross-validation value, and the performance of prediction on each topic of macro and weighted average value are ideal. Though MLP did achieved the highest cross-validation value, it did slightly poor performance for each topic comparing with RFC with SMOT, besides, MLP took too much time that cannot be endured. (Unless we can use GPU accelerating, but SKLEARN is not a module that designed for GPU, thus this is not possible.)

5. Discussion

We made a brief comparison of all the models we selected, and this can be seen from table 6.

For the two tables for measurement, they are all describing the whole performance of the algorithm, especially when the number of articles in one topic is large. However, as for there are some topics that are less than 10 articles, the performance of the model is hard to be described in the statistical way, instead, the second one, which including the result of the prediction is better for measurement. As for programmers can see for minority topics, what wrong classification did the model predicted.

For all the algorithms been used in our project, they all performed the flaw that have poor performance when the topic is the minority one. For linear SVM,

To further enhance the performance of classification, we think the next steps is filtering the key words in each topic. For some words in one topic, that might not be the essential one to do the prediction, while might confusing the models with other topics. The processing with these key words would optimize the classification result.

Instead of that, we can try ensemble learning that combine several good but different models together to improve the performance for whole dataset. The ensemble learning would accept the result from different models for the one that with most votes. [10] We have tried several combinations of models by using ensemble learning module in SKLEARN, while the result did not improve and compared to performance of the single model. This is due to the poorly selection of ensembled models. To improve this, we should try more different model combinations and generate the best one.

We see that a lot of articles are not suggested at all to the users, keeping in mind that having the right suggestions is of priority and hence it is best to not suggest a wrong article. The RFC model is meant to keep it intact in this hold and hence will work best in the long run to predict the test cases to give right suggestions.

6. Conclusion

This project is using machine learning models do the text classification. The model training is based on a 9500 instances database, the prediction is based on another database with 500 instances. We need to classify these articles into 11 topics. (IRRELEVANT is considered as one topic, too.) We have tried a list of machine learning models whose classification reports have been tabulated in appendix. As we can clearly see, the Machine learning models Multinomial Naive Bayes and RandomForest classifier gives a pretty good accuracy and Recall value for the training set after applying SMOT OverSampler method. With the generated training set, when using Cross validation, we see that although RandomForestClassifier gives a distributed probability values across articles, it is much better in terms of predicting the right article and hence the RECALL value for most articles are pretty good and its False negative value is low. Hence, we conclude to use the RandomForestClassifier with tuned parameters, to predict our Articles based on Text Classification. To improve the performance in the future, we think using ensemble learning and key words filter would be a good choice.

7. reference

[1] Chervonenkis, Alexey. (2013). Early History of Support Vector Machines. 10.1007/978-3-642-41136-6_3.

[2] Cortes C, Vapnik V. Support-vector networks. Machine learning. 1995 Sep 1;20(3):273-97.

[3] jteng. (2018). 支持向量机(SVM)之线性分类 [graph]Retrieved from https://blog.csdn.net/jteng/article/details/74905923

[**4]** Wikipedia contributors. (2020, April 25). Bayes' theorem. In Wikipedia, The Free Encyclopedia. Retrieved

from https://en.wikipedia.org/w/index.php?title=Bayes%27_theorem&oldid=953039771

[5] Gelareh Mohammadi, (2020) ‘Classification (2)’ [PowerPoint presentation]. COMP9417: Machine Learning & Data Mining. Available at: https://moodle.telt.unsw.edu.au/pluginfile.php/5454210/mod_resource/content/1/Week3_Classification2_Part2.pdf (Accessed: 24 April 2020)

[6] Wikipedia contributors. (2020, January 22). Additive smoothing. In Wikipedia, The Free Encyclopedia. Retrieved

from https://en.wikipedia.org/w/index.php?title=Additive_smoothing&oldid=937083796

[7] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research, 16, 321-357.

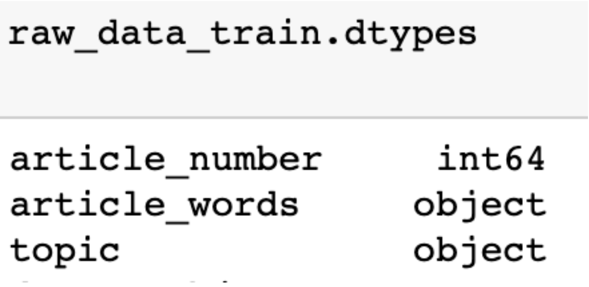
[8] Wikipedia contributors. (2020, April 20). Oversampling and undersampling in data analysis. In Wikipedia, The Free Encyclopedia. Retrieved from https://en.wikipedia.org/w/index.php?title=Oversampling_and_undersampling_in_data_analysis&oldid=952129679

[9] Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. R news, 2(3), 18-22.

[10] Robi Polikar (Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. R news, 2(3), 18-22.2009) Ensemble learning. Scholarpedia, 4(1):2776.

Appendix

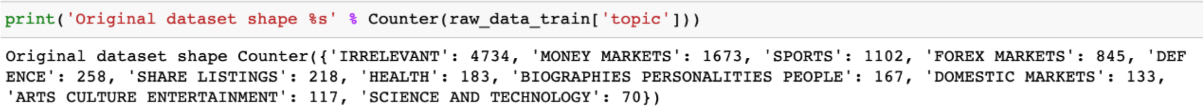
All the graphs and tables are in the order that appeared on the main part.



```
raw_data_train.dtypes

article_number      int64
article_words      object
topic              object
```

Figure 1. The data types in the training dataset



```
print('Original dataset shape %s' % Counter(raw_data_train['topic']))

Original dataset shape Counter({'IRRELEVANT': 4734, 'MONEY MARKETS': 1673, 'SPORTS': 1102, 'FOREX MARKETS': 845, 'DEF
ENCE': 258, 'SHARE LISTINGS': 218, 'HEALTH': 183, 'BIOGRAPHIES PERSONALITIES PEOPLE': 167, 'DOMESTIC MARKETS': 133,
'ARTS CULTURE ENTERTAINMENT': 117, 'SCIENCE AND TECHNOLOGY': 70})
```

Figure 2. The article statistics for each topic in training set

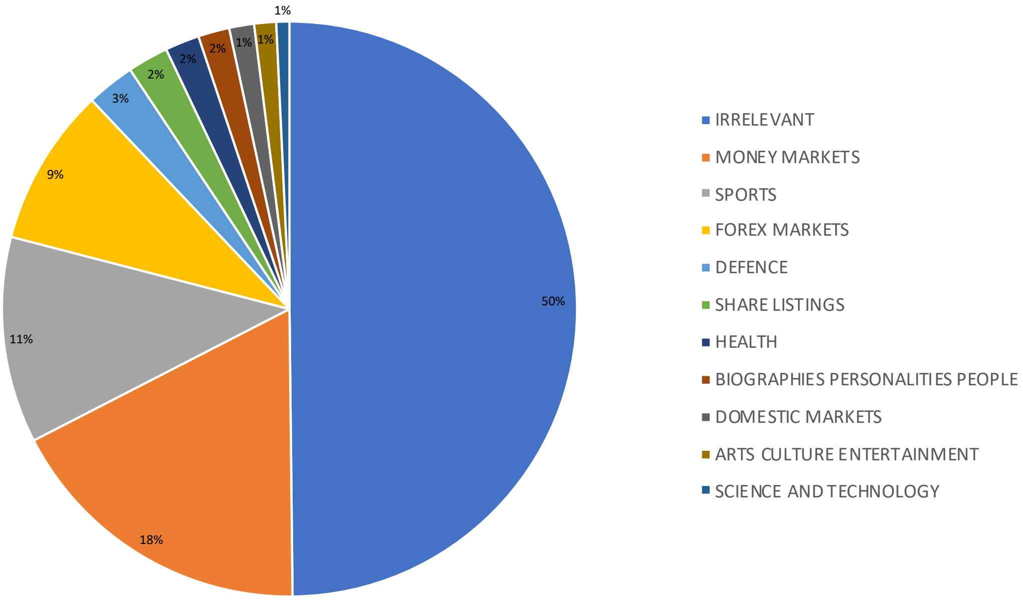


Figure 3. Distribution of article topics

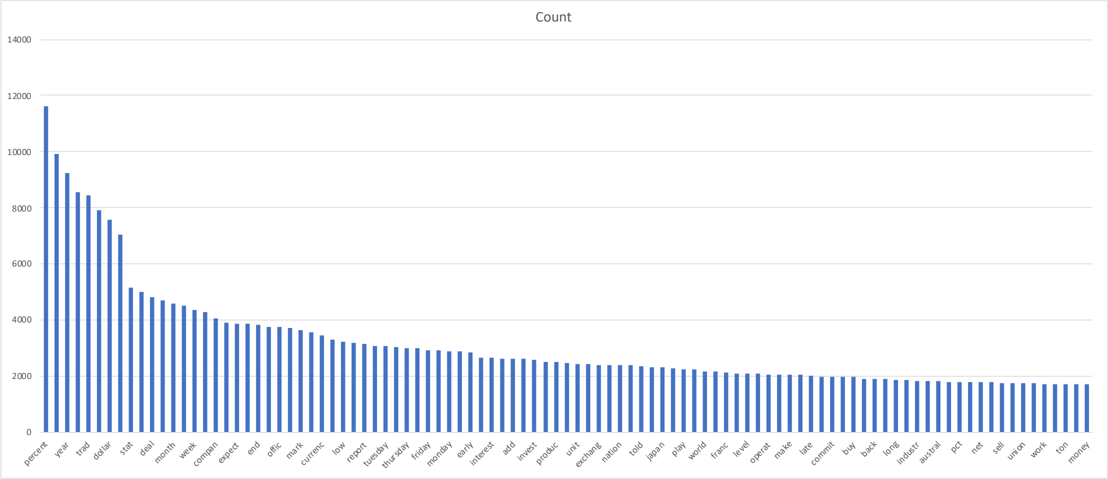


Figure 4. The words count in all the articles

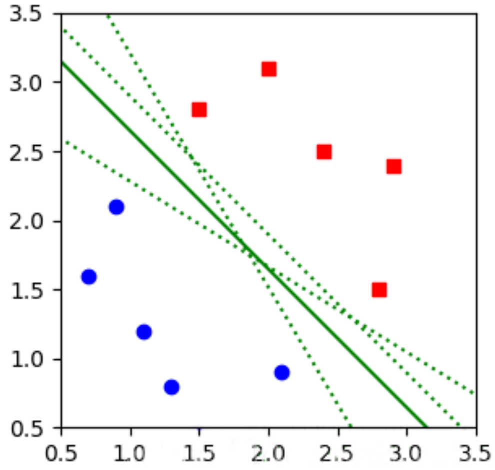


Figure 5. The hyperplane classification example 1. [3]

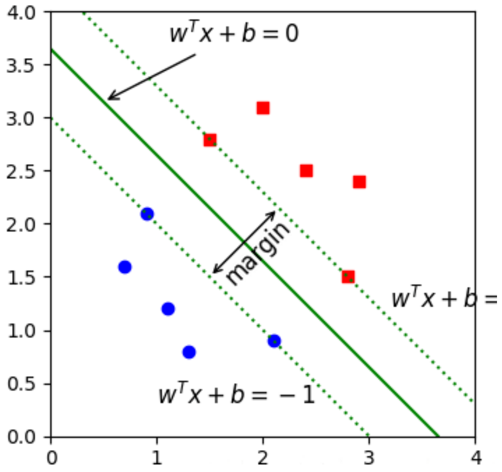


Figure 6. Margin of hyperplane example. [3]

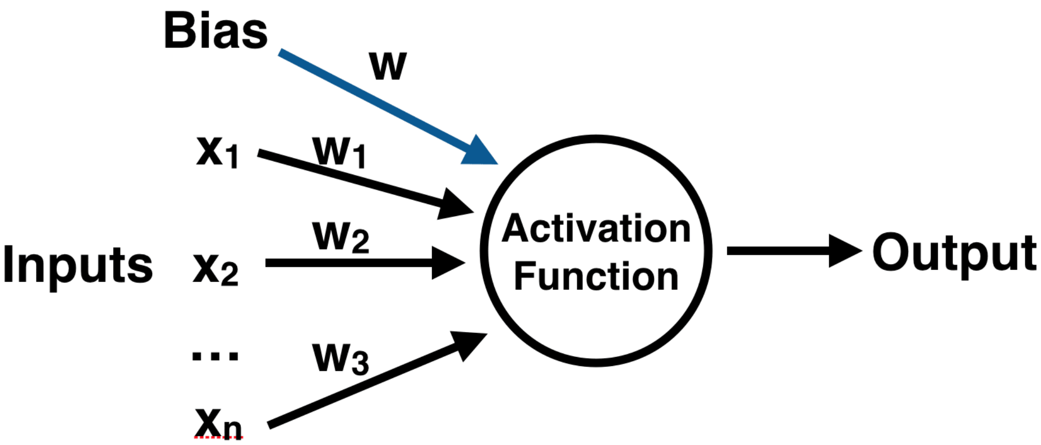


Figure 7. The perceptron classifier model

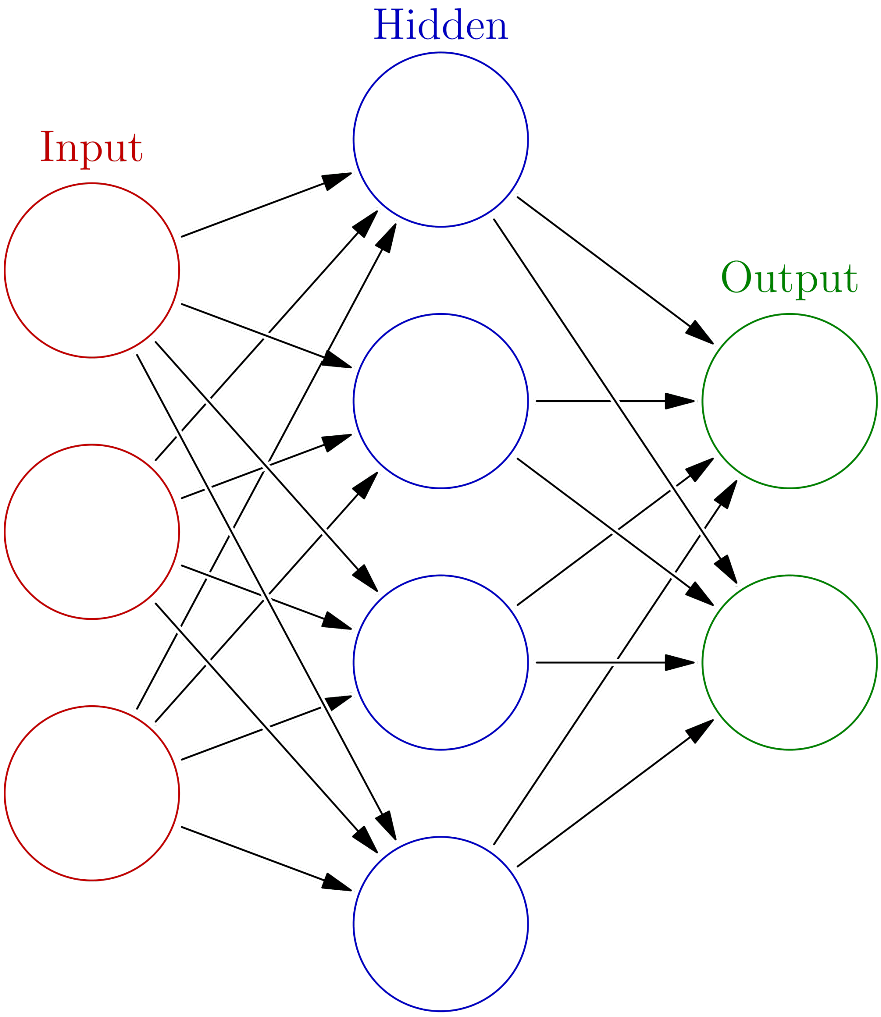


Figure 8. The hidden layers in the neural network

Topic name	Precision	Recall	F1
ARTS CULTURE ENTERTAINMENT	0.00	0.00	0.00
BIOGRAPHIES PERSONALITIES PEOPLE	0.50	1.00	0.67
DEFENCE	0.43	0.20	0.27
DOMESTIC MARKETS	0.67	0.46	0.55
FOREX MARKETS	0.40	1.00	0.57
HEALTH	0.80	0.57	0.67
MONEY MARKETS	0.85	0.80	0.83
SCIENCE AND TECHNOLOGY	0.52	0.67	0.59
SHARE LISTINGS	0.00	0.00	0.00
SPORTS	0.71	0.71	0.71
Marco average	0.48	0.34	0.37
Weighted average	0.72	0.75	0.72

Table 1. The metrics table for linear SVM

Topic name	Precision	Recall	F1
ARTS CULTURE ENTERTAINMENT	0.4	0.67	0.5
BIOGRAPHIES PERSONALITIES PEOPLE	0.78	0.47	0.58
DEFENCE	0.60	0.69	0.64
DOMESTIC MARKETS	0.00	0.00	0.00
FOREX MARKETS	0.41	0.31	0.35
HEALTH	0.83	0.71	0.77
MONEY MARKETS	0.44	0.80	0.57
SCIENCE AND TECHNOLOGY	0.00	0.00	0.00
SHARE LISTINGS	0.50	0.14	0.22
SPORTS	0.95	1.00	0.98
Marco average	0.53	0.51	0.49

Weighted average	0.76	0.73	0.73

Table 2. The metrics table for Multinomial Naïve Bayes

Topic name	**Precision**	**Recall**	**F1**
ARTS CULTURE ENTERTAINMENT	0.20	0.33	0.25
BIOGRAPHIES PERSONALITIES PEOPLE	0.60	0.20	0.30
DEFENCE	0.86	0.46	0.60
DOMESTIC MARKETS	0.67	1.00	0.80
FOREX MARKETS	0.47	0.38	0.42
HEALTH	0.70	0.50	0.58
MONEY MARKETS	0.51	0.57	0.53
SCIENCE AND TECHNOLOGY	0.00	0.00	0.00
SHARE LISTINGS	0.60	0.43	0.50
SPORTS	0.95	0.95	0.95
Marco average	0.58	0.52	0.53
Weighted average	0.74	0.75	0.74

Table 3. The metrics table for MLP

Topic name	**Precision**	**Recall**	**F1**
ARTS CULTURE ENTERTAINMENT	0.40	0.67	0.50
BIOGRAPHIES PERSONALITIES PEOPLE	0.40	0.13	0.20
DEFENCE	0.64	0.69	0.67
DOMESTIC MARKETS	0.00	0.00	0.00
FOREX MARKETS	0.52	0.48	0.50
HEALTH	0.82	0.64	0.72
MONEY MARKETS	0.51	0.64	0.57
SCIENCE AND TECHNOLOGY	0.50	0.33	0.40
SHARE LISTINGS	0.47	1.00	0.64

SPORTS	0.93	0.93	0.93
Marco average	0.55	0.58	0.54
Weighted average	0.76	0.75	0.75

Table 4. The metrics table for RFC with SMOT

Topic name	**Suggested articles**	**Precision**	**Recall**	**F1**
ARTS CULTURE ENTERTAINMENT	9952	0.40	0.67	0.50
BIOGRAPHIES PERSONALITIES PEOPLE		0.40	0.13	0.20
DEFENCE		0.64	0.69	0.67
DOMESTIC MARKETS		0.00	0.00	0.00
FOREX MARKETS	9525, 9550, 9588, 9682, 9708, 9798, 9986	0.52	0.48	0.50
HEALTH		0.82	0.64	0.72
MONEY MARKETS	9516, 9534, 9571, 9583, 9586, 9587, 9600, 9602, 9618, 9634	0.51	0.64	0.57
SCIENCE AND TECHNOLOGY		0.50	0.33	0.40
SHARE LISTINGS		0.47	1.00	0.64
SPORTS	9513, 9514, 9520, 9536, 9569, 9573, 9574, 9580, 9596, 9608	0.93	0.93	0.93

Table 5. The result metrics table for RFC with SMOT on the test set

	Advantages	Disadvantages
SVM	1. Fast computing 2. Easy debug	1. Not ideal performance 2. Not suitable for unbalanced dataset
MNB	1. Fast computing	1. Performance is not ideal enough

	2. Easy debug 3. Better performance on u;	2. Still need to be improved for small dataset
MLP	1. High performance 2. Average performance on each topic	1. Corresponding horrible calculation time 2. Slow, slow, slow, unbearable when it with SMOT
RFC With SMOT	1. Higher accuracy performance 2. Friendly for small dataset	1. The dataset was enlarged that increased the compute cost.

Table 6. The comparison table for 3 models

Topic name	**Threshold**
ARTS CULTURE ENTERTAINMENT	0.7452
BIOGRAPHIES PERSONALITIES PEOPLE	0.75
DEFENCE	0.75
DOMESTIC MARKETS	0.75
FOREX MARKETS	0.7275
HEALTH	0.75
MONEY MARKETS	0.6975
SCIENCE AND TECHNOLOGY	0.75
SHARE LISTINGS	0.75
SPORTS	0.74

Table 7. The threshold for the selection of articles