COMP9417 Project Report

Group Name: BugMaker

Group Members:

Shuonan Wang Z5158229

Xin Cai Z5108236

Yukang Yan Z5158298

Yunfan Wang Z5171928

Zhihan Qin Z5290141

1. Introduction

1.1. Aim

This project aims to apply machine learning algorithms to predict specific outputs in a dataset which is a set of pre-processed articles form 10 different topics. The final output will be the recommendation list of the top 10 articles for each topic from the test dataset at most. The project focuses on 2 main parts -- feature extraction models and classification models. Details about the model selection and hyper-parameter tuning will be presented in the next chapter.

1.2. Dataset

Figure 1.1 demonstrates the distribution of each topic in the training set. There are 9500 articles in total, 49.93% of which are irrelevant articles. Besides, the distribution of 10 topics is also imbalanced. **Figure 1.2** shows word counts in each article which is categorized by topic and most articles are around 1000 words.

Based on the above observations, we mainly pay attention to the scale and weight of features, which are the words in the training set, and also anticipate that those methods which give more weights on rare topics will achieve better performance. To validate our training result, we split the training set into 2 parts which are a training set with 9000 articles and a validation set with 500 articles on the basis of the portion.

Additionally, based on the fact that the test set only has 500 articles probably with an extremely

nonuniform distribution, predicted probability matrix with a threshold is used to give out the recommended articles.

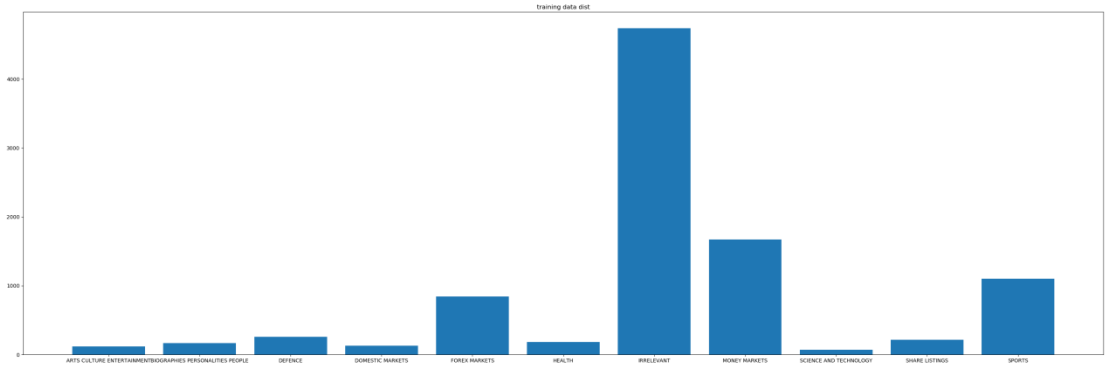


Figure 1.1

Table 1.1

Topic name	Articles numbers
ARTS CULTURE ENTERTAINMENT	117
BIOGRAPHIES PERSONALITIES PEOPLE	167
DEFENCE	258
DOMESTIC MARKETS	133
FOREX MARKETS	845
HEALTH	183
MONEY MARKETS	1673
SCIENCE AND TECHNOLOGY	70
SHARE LISTINGS	218
SPORTS	1102
Irrelevant	4734

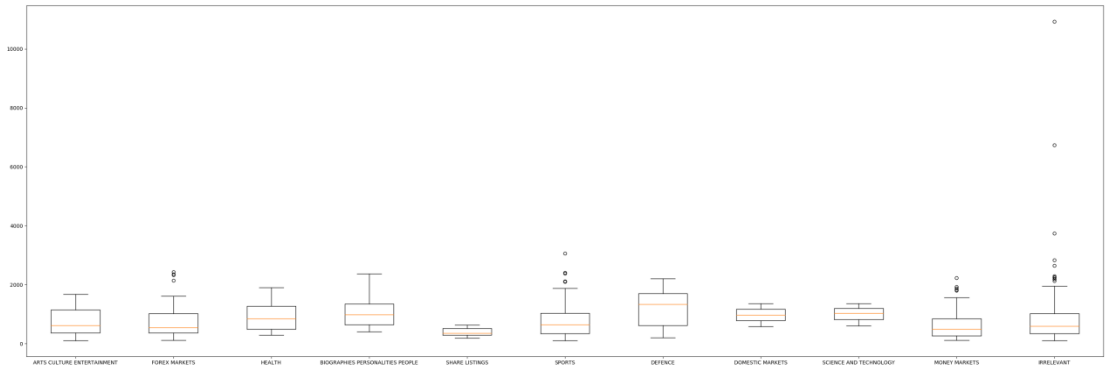


Figure 1.2

2. Methods

2.1. Method choice

To deal with the problem, there are two steps. First, all models are selected and compared under the principle of controlling variable. We compare extraction models with logistic regression

classifier and compare different classifiers with document frequency (DF) extraction model. All hyper-parameters are set with a relatively good performance during the initial test phase. The reason to choose those 2 models is that their computational speed is fast, and they can achieve a relatively satisfying result at this stage. At the second step, top two feature extraction models and top two classifiers will be chosen for further adjustment.

A brief explanation of all methods we have experimented and the process to choose the final model will be illustrated in this chapter.

2.1.1 choice of feature extraction

This part will introduce all feature extraction methods we compared, some detailed results of comparison will be shown in **Chapter 3.4**

Df: Df is a basic method which counts the frequency of words in each article and form a word-frequency matrix. The method usually tends to overweight the words with more frequency. Considering the fact that some topic in the training set only have a small portion, this method performed well among all these choices.

PCA: Principal component analysis decreases dimensions of features. This method has subtle increase in some cases, however the process is very time-consuming, so we did not deploy this method eventually.

TF-IDF: TF-IDF is a commonly used weighting technique for data mining, tf-idf(i,j) = tf(i,j) * idf(i), we found if a word appears more in a certain topic and less in others, IDF value would be relatively large. The performance is not as good as we expected, this might because in a certain topic, if a feature only appears in some individual texts and rarely occurs in others, these individual texts are not excluded as special cases in this topic. Therefore, such feature items are not representative.

TF-IDF method based on variance: TF-IDF is weak at selecting some feature which takes possession of the most category discernibility. As a result, we applied a TF-IDF method based on variance, the weight value of the feature word wi in document dj is weight (wi, dj) = TF-IDF * var (wi).

Chi-square: Chi-square calculates the actual deviation of the actual value and the theoretical value to verify the correctness of the theory. The basic equation is $x^2 = \sum \frac{(O-E)^2}{E}$. The disadvantage might be that it only counts whether a word appears in a document, regardless of how many times it appears, this would exaggerate the role of low-frequency words. However, in our comparison, it has a higher f1 score than other methods.

After comparing the above methods of feature extraction, we finally apply chi-square to train the model.

2.1.1. Classifier choice

The main classifiers tested are based on the lecture and comparison of results is in **Chapter 3.3**.

KNN algorithm: KNN algorithm finds the k nearest neighbors of x among training instances

and scores the category candidates based on the class of k neighbors. The similarity of x and each neighbor's document could be the score of the category of the neighbor documents. **[1]**

Ada-boost: Ada-boost is an ensemble learning algorithm to solve the classification problem. Adjusting the sample weight and the weight of the weak classifier, the weak classifier with the smallest weight coefficient is selected from the trained weak classifier into a strong classifier with higher prediction accuracy.

SVM: Kernel SVM is traditionally used for the binary classification. In this project, we generate a multi-class SVM with a "one-vs-rest" approach for multi-class problems. And the probability SVM calculates the probability that the sample belongs to its category by using Sigmoid function.**[2]**

Random forest: Random forest is an algorithm that integrates multiple trees using the idea of Ensemble Learning. Its basic unit is the decision tree. We select Gini measure $G(Xm) = \sum_{k} p_{mk}(1 - p_{mk})$ in random forest algorithm, where p_{mk} is the proportion of class k samples in node m **[3]**.

Logistic regression: This algorithm is to map the result of a linear function to sigmoid function $y = \frac{1}{1+e^{-x}}$ **[4]**. This algorithm is a kind of classification measure. It will establish a cost function, and then iteratively replace the optimal model parameters through optimization methods, and then test to verify the quality of our transformed model.

Soft Voting Classifier: It takes all model prediction samples' the average probability of a certain class as the standard, and the corresponding type with the highest probability is the final prediction result

2.1.2. Justification of the method choice

KNN algorithm: The performance of KNN is dependent on finding a meaningful distance function, and text classification needs more feature values. Similar articles are easily misclassified in the same topic by distance function because the distance of the same topic article is too far away from each other.

Ada-boost: Ada-boost is based on a simple decision tree with one node to improve weak classifiers through continuous training. Because there are 11 topics, it is difficult to classify successfully. And if the tree is too complicated, it would lose the simplicity of Ada-boost which lead to bad effect as well.

SVM: Compared with other methods above, SVM has a good performance in terms of data and results. It is available for data set with multi-features by using a kernel function. The kernel function is calculated on the low dimension in advance, and the virtual classification effect is shown on the high dimension, which avoids the complex calculation directly in the high dimensional space. It can also avoid overfitting, learning a form of decision boundary

called the maximum margin hyperplane. Detailed experimental results are shown in the discussion section below.

Random forest: After the experiment, we got the performance of random forest is unsatisfactory. Through analysis, we found that irrelevant articles are scattered, which will interfere with the Random forest classification. Additionally, the random forest will produce overfitting. Finally, some articles have a very low weight. After random separation, even if we can predict them correctly, their probabilities are very low. So, we give up this classifier.

Logistic regression: We select the OVR method as multi-class, which is more suitable for a linear model to conduct binary classification. Furthermore, Logistic regression can easily update the model to absorb new data. At last, we can get the weight of articles directly.

Soft Voting Classifier: Compared with other classifiers, we found the performance of Soft Voting Classifier is better. We vote between Logistic Regression and SVC. The soft voting classifier uses the sum of the probabilities of each category to make predictions. The accuracy of soft voting is slightly higher than that of hard voting. Because it considers the difference of each model. We will talk about more details in Results and Discussion part.

2.2. Hyper-parameter tuning

To tune the hyper-parameters of the chosen model, the project follows the rule of control variable and grid-search. For hyper-parameters which are hard to judge the impact and importance between them, we use grid-search to find the best value. For those which might consume too much time and relatively independent over others, the method of control-variable is applied to find the hyper-parameter. The tuning process and result will be presented in **Chapter 3.3**.

2.3 Evaluation metrics

During the process of building a machine learning model, it is based on some evaluation indicators to evaluate the performance of both the learning and prediction, thus it is an important part of machine learning.

For classification, the common evaluation indicators include Precision, Recall and F1-measure. In most cases, the calculation of Precision and Recall should build the confusion matrix at first which is shown below.

	the number of articles practically belonging to this topic	the number of articles practically not belonging to this topic
the number of articles which are	TP	FP

classified as this topic		
the number of articles which are not classified as this topic	FN	TN

Precision is the proportion of the articles that truly belong to a topic of all the articles which are classified as this topic:

$$P = TP / (TP + FP)$$

Recall is the proportion of the articles that are classified to a topic of all the articles that truly belong to this topic:

$$R = TP / (TP + FN)$$

F1 is a trade-off between Precision and Recall:

$$F1 = 2PR / (P+R) = 2TP / (2TP + FP + FN)$$

3. Results

3.1. Presenting

Finally, we apply chi-square and voting method to get the result. **Table 3.3.1** is the score over all test instances and **Table 3.3.2** is the score based on suggested articles.

Table 3.3.1

Topic name	**Precision**	**Recall**	**F1**
ARTS CULTURE ENTERTAINMENT	**0.50**	**0.67**	**0.57**
BIOGRAPHIES PERSONALITIES PEOPLE	**0.71**	**0.33**	**0.45**
DEFENCE	**0.90**	**0.69**	**0.78**
DOMESTIC MARKETS	**0.33**	**0.50**	**0.40**
FOREX MARKETS	**0.62**	**0.44**	**0.51**
HEALTH	**0.64**	**0.50**	**0.56**
MONEY MARKETS	**0.57**	**0.72**	**0.64**
SCIENCE AND TECHNOLOGY	**0.25**	**0.33**	**0.29**
SHARE LISTINGS	**0.67**	**0.57**	**0.62**

SPORTS	**0.95**	**0.98**	**0.97**

Table 3.3.2

Topic name	**Suggested articles**	**Precision**	**Recall**	**F1**
ARTS CULTURE ENTERTAINMENT	**9703,9789,9952,9830,9933**	**0.40**	**1.00**	**0.57**
BIOGRAPHIES PERSONALITIES PEOPLE	**9940,9988,9758,9933,9878, 9896**	**0.83**	**0.62**	**0.71**
DEFENCE	**9559,9770,9773,9616,9670, 9576,9987,9904,9706,9713**	**0.90**	**1.00**	**0.95**
DOMESTIC MARKETS	**9994,9989**	**0.50**	**1.00**	**0.67**
FOREX MARKETS	**9986,9977,9551,9875,9718, 9671,9588,9682,9786,9548**	**0.50**	**1.00**	**0.67**
HEALTH	**9661,9937,9807,9810,9929, 9873,9926,9609,9887,9833**	**0.60**	**0.67**	**0.63**
MONEY MARKETS	**9765,9618,9755,9602,9871, 9998, 9723,9737,9516,9967**	**0.80**	**0.62**	**0.70**
SCIENCE AND TECHNOLOGY	**9617,9722,9982,9621**	**0.25**	**0.50**	**0.33**
SHARE LISTINGS	**9518,9601,9666,9667,9972, 9715**	**0.67**	**1.00**	**0.80**
SPORTS	**9848,9596,9813,9942,9858, 9752,9620,9541,9656,9574**	**0.90**	**1.00**	**0.95**

3.2. Result evaluation using appropriate metrics

Precision reflects the accuracy of each topic whereas recall means the ratio of the true positive number of each topic to the total number of the topic. F1 is the general evaluation combining the two. As the result shown below, this method can make good suggestions on 'BIOGRAPHIES PERSONALITIES PEOPLE', 'DEFENCE',' MONEY MARKETS' and 'SPORTS' and the precession rate is very low on the topic of 'SCIENCE AND TECHNOLOGY' and there are only 2 recommendations of 'DOMESTIC MARKETS'. Figure 3.2.1 shows the distribution of test set, which indicates that the actual articles of these 2 topics are only 3 and 2.

Generally, the precision is higher than recall and has a good score in most topics, which means the final predict can distinguish the true articles well. And the relatively low recall means there are some irrelevant articles misclassified. To summarize, we think the final result is passable

as all topics can suggest at least one right article and most topics perform well, considering the actual distribution.

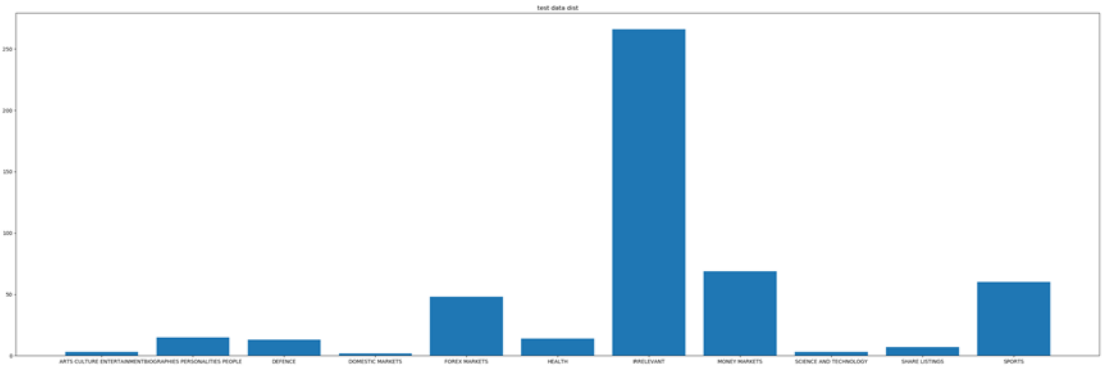


Figure 3.2.1

Table 3.2.1

Topic name	Articles numbers
ARTS CULTURE ENTERTAINMENT	3
BIOGRAPHIES PERSONALITIES PEOPLE	15
DEFENCE	13
DOMESTIC MARKETS	2
FOREX MARKETS	48
HEALTH	14
MONEY MARKETS	69
SCIENCE AND TECHNOLOGY	3
SHARE LISTINGS	7
SPORTS	60
Irrelevant	266

3.3. Evaluation of various hyper-parameters and design choices

After experimenting with different classifiers shown in Figure 3.3.1, it is easy to notice that logistic regression and SVM have a better performance over others and they are complementary at some degree, so the final decision is to combine them into a voting classifier. As it is shown in the figure, the default voting classifier has a similar performance comparing to the two classic classifiers. The x-axis in the figure is the encoded topic with alphabetical order and 6 is irrelevant articles. The hyper-parameters are set with a relatively good result during the laboratory phase.

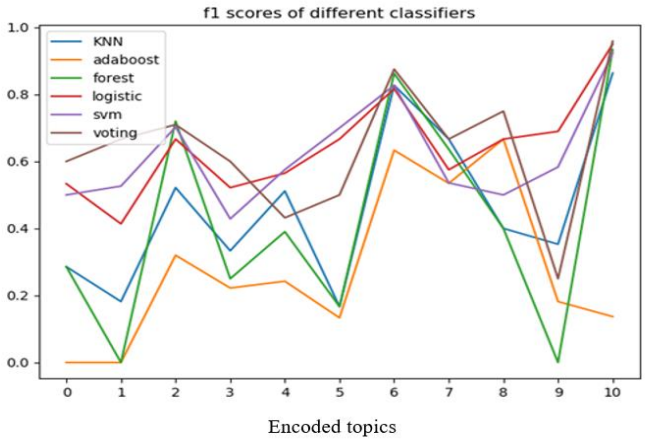


Figure 3.3.1

Once the model is chosen, the next step is to tune its hyper-parameters. In this model, there are 3 classifiers need to be adjusted -- logistic regression, SVM and voting. We deploy a grid search and apply GridSearchCV imported from Scikit-learn with 5-fold cross-validation to complete this phrase. For logistic regression, we adjust the max-iter, multi-class and solver to find the best combination. Then, considering the fitting time of SVM, we first observe the effect of iterations on the f1-score with 10-fold cross-validation as shown in **Figure 3.3.2** and conclude that the iteration time is very subtle after 1500. Then we use grid-search to tune the C and gamma.

At last, we compare the overall score to decide the best weights for voting. All the tuning log are presented in **AP3.3.1** in Appendices.

After confirming the final model and hyper-parameters, we observe the output result based on the probability matrix on the validation set and set the probability threshold to 0.4, so that only the article whose probability is bigger than 0.4 and which ranked in top 10 will be suggested.

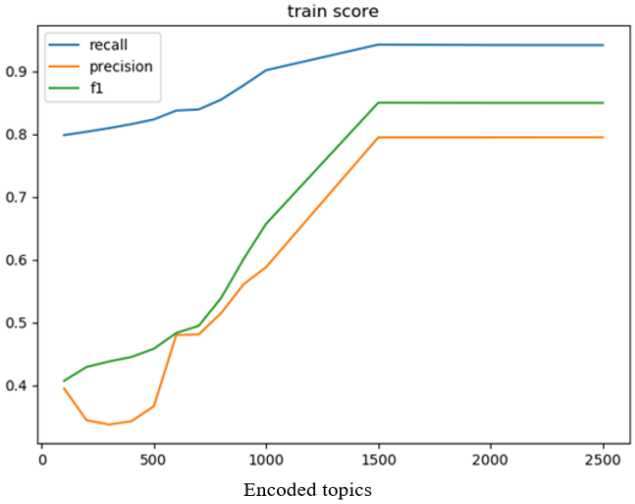


Figure 3.3.2

3.4. Feature importance analysis

Firstly, with the use of control variables, we test the performance of each extraction method with the same number of features. As it is shown in the figure below, chi2 generally performs well based on the performance of all 10 article topics and thus, we choose chi2 as our best extraction method and then focus on the analysis of the optimal number of chi2 features used.

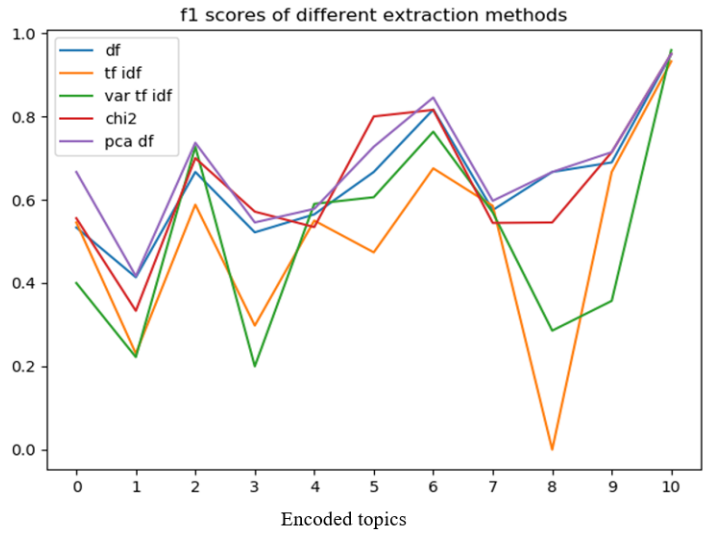


Figure 3.4.1

Figure 3.4.2

The feature number of chi2 determines the number of category variables selected. By testing the iterative num between 1000 and 2400, it can be observed that the highest accuracy is 0.747 when iteration num is equal to 1600 from the above figure. Therefore, we choose 1600 as the best feature number of chi2 and combine it with vote classifier to get a result of classification.

4. Discussion

4.1. Performance analysis

As mentioned in the method choice section, figure 3.3.2 and figure 3.4.1, we select the best extraction model and classifier, which is CHI2 + vote. Compared with other methods, combination of chi2 and voting classifier performs best. CHI2 is more suitable for the dataset of this project. Because we have 10 different topic and 1 irrelevant topic. The soft voting classifier uses the sum of the probabilities of each category to make predictions. After the experiment, the effect of this integration method is indeed excellent. This method can find the classifier, which is suitable for our data set to reach the greatest extent.

4.2. Evaluation metric analysis

As discussed before, the precision score reflects the accuracy of each topic whereas recall means the ratio of the true positive number of each topic to the total number of the topic. For this project, the reader wants to have as many as true positive articles presented on its suggested list, so the precision score in the second result table is more representative.

4.3. Further improvement

In the project, we mainly explore feature engineering, classifier modelling and evaluation methods of news classification. Feature engineering is based on Bag of Words and modelling is based on classifiers within machine learning. We have been devoted to developing different methods to deal with this problem and finally, the output performance has been effectively improved compared with the beginning. However, the performance still needs further promotion and will start from the following aspects.

1. For feature engineering, by reading the given articles of the dataset, we cannot get abundant semantic meaning from our perspectives. Without taking into account the semantic meaning, we have not tried word embeddings which are a common family of NLP techniques on feature extraction.
2. For building classifiers, we haven't tried deep learning classifiers such as CNN and RNN in that they are mainly based on context-awareness and need features indicating the position of each word. Deep learning is powerful at learning details and as a result, the promotion of performance can be achieved when it comes to complicated situations.
3. For data augmentation, basic techniques of up sampling and subsampling had been adopted due to the imbalanced distribution of the dataset. However, we haven't witnessed the improvement, so we gave up data augmentation at last. Further research and experiment should be done in this step to promote the robustness of the model.

5. Conclusion

In this project, we first have general experiments of different kinds of extraction methods and classification methods. With the comparison of the methods using relatively good hyper-parameters, we find out the top2 extraction methods which are DF and Chi2. Then we combine the top2 classifiers into the voting classifier. After that, we deploy grid-search and k-fold cross-validation on the training set for tuning the hyper-parameters. Next, we pick the threshold of recommendation articles by observing the probability matrix in the validation set. Eventually, we compare 2 sets of results based on 2 different extraction methods and choose the final result. Generally speaking, the influence of extraction methods and their parameters is obvious, Meanwhile, with all classic classifiers presented in chapter 2 logistic regression and SVM tend to have a better scope. However, given the fact that the distribution of articles is very irregular and the way to process features is very primitive, the overall result is just passable, and it is hard to have further improvement under the current structure of this assignment. To achieve more progress in text classification, more advanced algorithms, such as attention-based LSTM and other neural networks may be compulsory.

Reference List:

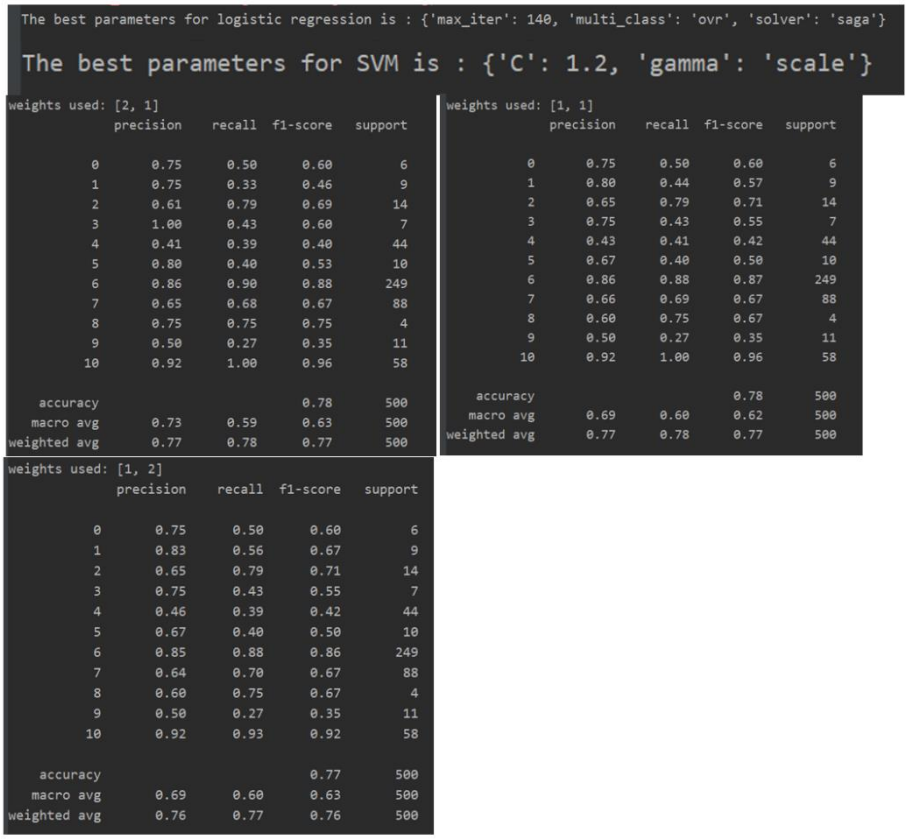
[1]. Text Classification Algorithms: A Survey Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura E. Barnes, Donald E. Brown (23 April 2019) DOI:10.3390/info10040150

[2]. Classification of News Dataset Olga Fuks Stanford University [2018]

[3]. Scikit-learn.org. n.d. 1.10. Decision Trees — Scikit-Learn 0.22.2 Documentation. [online] Available at: <https://scikit-learn.org/stable/modules/tree.html> [Accessed 19 April 2020].

[4]. Scikit-learn.org. 2020. 1.1. Linear Models — Scikit-Learn 0.22.2 Documentation. [online] Available at: <https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression> [Accessed 19 April 2020].

Appendices:



```
The best parameters for logistic regression is : {'max_iter': 140, 'multi_class': 'ovr', 'solver': 'saga'}

The best parameters for SVM is : {'C': 1.2, 'gamma': 'scale'}
```

weights used: [2, 1]

	precision	recall	f1-score	support
0	0.75	0.50	0.60	6
1	0.75	0.33	0.46	9
2	0.61	0.79	0.69	14
3	1.00	0.43	0.60	7
4	0.41	0.39	0.40	44
5	0.80	0.40	0.53	10
6	0.86	0.90	0.88	249
7	0.65	0.68	0.67	88
8	0.75	0.75	0.75	4
9	0.50	0.27	0.35	11
10	0.92	1.00	0.96	58
accuracy			0.78	500
macro avg	0.73	0.59	0.63	500
weighted avg	0.77	0.78	0.77	500

weights used: [1, 1]

	precision	recall	f1-score	support
0	0.75	0.50	0.60	6
1	0.80	0.44	0.57	9
2	0.65	0.79	0.71	14
3	0.75	0.43	0.55	7
4	0.43	0.41	0.42	44
5	0.67	0.40	0.50	10
6	0.86	0.88	0.87	249
7	0.66	0.69	0.67	88
8	0.60	0.75	0.67	4
9	0.50	0.27	0.35	11
10	0.92	1.00	0.96	58
accuracy			0.78	500
macro avg	0.69	0.60	0.62	500
weighted avg	0.77	0.78	0.77	500

weights used: [1, 2]

	precision	recall	f1-score	support
0	0.75	0.50	0.60	6
1	0.83	0.56	0.67	9
2	0.65	0.79	0.71	14
3	0.75	0.43	0.55	7
4	0.46	0.39	0.42	44
5	0.67	0.40	0.50	10
6	0.85	0.88	0.86	249
7	0.64	0.70	0.67	88
8	0.60	0.75	0.67	4
9	0.50	0.27	0.35	11
10	0.92	0.93	0.92	58
accuracy			0.77	500
macro avg	0.69	0.60	0.63	500
weighted avg	0.76	0.77	0.76	500

AP 3.3.1