Group Project Report

COMP9417 Machine Learning and Data Mining

T1 2020, UNSW

Group Name: Henry Erzi

Raymond Lu	z5277884
Ziyang Liang	z5420778
Yunze Shi	z5244619
Yinong Wang	z5219512
Yangqi Zhang	z5235062

1. Introduction

In this assignment we implemented a news article recommendation system based on machine learning algorithms, the main function of this program is to predict a topic for each of the 500 unseen articles and suggest up to 10 of the most relevant articles in each topic to 10 users. The training set contains 9500 articles with each article being categorised into one topic.

Paper highlights:

- We suggest a feature engineering method that only keeps the most frequent words in each topic as keywords and discards other words with lower frequency. TFIDF and Word Count vectorizer are then used to convert the bag of words into numerical features space.
- We proposed a 2-layer model in which the first layer only classifies irrelevant articles and other articles and the second layer focuses on classifying other articles into the rest of topics.
- In both first and second layers, we trained and tuned four different classifying models: Logistic Regression, Decision Tree, Naive Bayes and Support Vector Machine. A final model that ensembles these four models according to their performance in the validation set are suggested. The overall accuracy of the model is **0.81**.
- The recommendation is based on the predicted probability by the ensemble model.

2. Exploratory Data Analysis

2.1 Class distribution

There are 10 different topics plus one noise category 'irrelevant'. The first concern of ours is whether the distribution of articles is balanced or not over the 10 topics we are interested in. Another concern is the proportion of 'irrelevant' articles in the training set. Figure 2.1 shows the distribution of articles over topics in the training set, it can be observed that topics are highly unbalanced, with a gap of over 1600 articles between the largest size relevant class 'MONEY MARKETS' and the smallest size relevant class 'SCIENCE AND TECHNOLOGY'. Also, articles that are irrelevant account for nearly 50% of the total collection of articles.

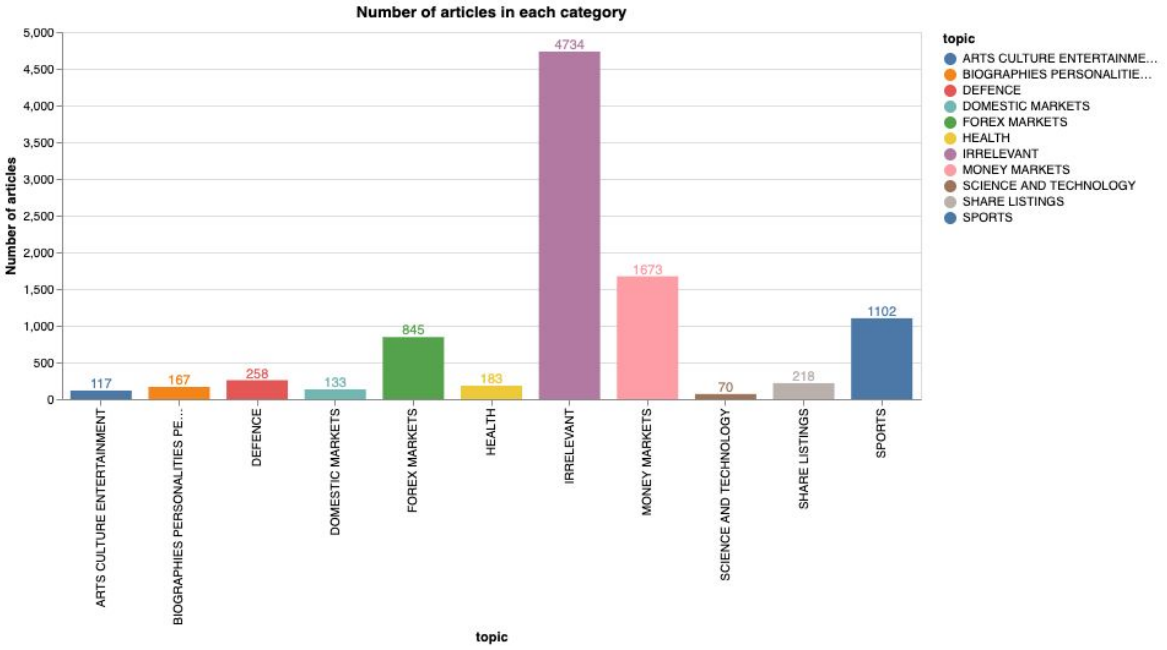


Fig 2.1 Distribution of articles in the training set

Another factor that was considered during the development of the models is the length of articles, as longer articles tend to have larger probabilities of covering a wider range of words. Figure 2.2 shows the number of words in an article in each topic. The plot suggests an obvious imbalance of article lengths in each topic, and a considerable amount of outliers in each topic.

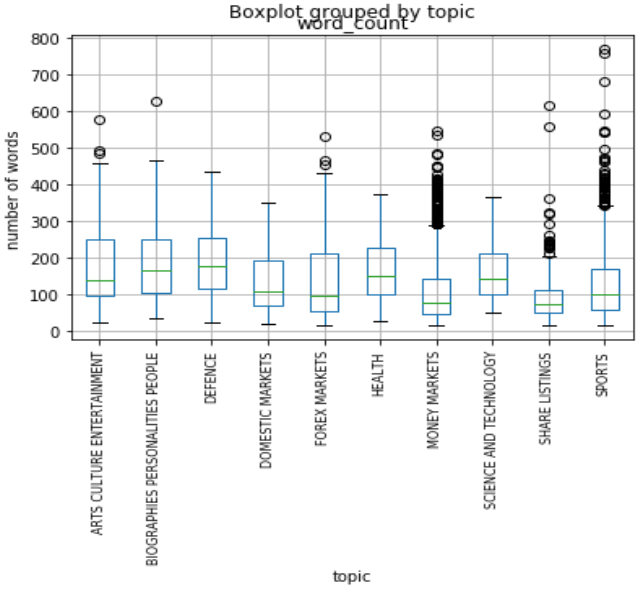


Fig2.2 Article length of each topic

There are measures for handling such imbalanced datasets. One approach is oversampling minor classes. Synthetic Minority Over-sampling Technique (SMOTE) [1] is adopted in this

project to perform the oversampling. Another approach is to take into account the length of an article and the size (or proportion) of a class when performing feature engineering. Term frequency-inverse document frequency or TF-IDF [2] is used in this project to compare with Countvectorizer, another feature representation method.

Apart from SMOTE and TF-IDF which are involved in the preparation stage for training our models, multiple metrics besides accuracy were used to evaluate the models learned, such as the precision score, the recall and the F1 score.

2.2 Words in different classes

It is natural to argue that terms used in articles would be distinguishable from topic to topic. Table 2.1 shows the top five frequently appearing words in each topic, which suggests that these frequent words in different topics vary significantly.

Table 2.1 Top 5 frequently used words in each topic

Most frequent words in each topic and its frequencies					
Topics	**1st frequent**	**2nd frequent**	**3rd frequent**	**4th frequent**	**5th frequent**
Arts	year (156)	film (153)	million (102)	stat(83)	work(81)
Biographies	year (249)	presid (198)	yeltsin (140)	stat (133)	hospit (124)
Defence	nato (511)	stat (410)	milit (406)	forc (381)	defend (357)
Domestic	import (541)	ton (324)	precent (265)	year (262)	oil (191)
Forex	dollar (3358)	bank (2178)	rate (1720)	trad (1671)	market (1662)
Health	year (253)	health (245)	stat (241)	diseas (184)	tobacc (169)
Irrelevant	precent (5945)	year (5237)	million (4738)	trad (3660)	market (3547)
Money	bank (4447)	rate (3684)	precent (3515)	dollar (3387)	market (2890)
Science	spac (141)	mir (110)	shuttl (83)	scient (77)	year (73)
Share listing	shar (740)	compan (391)	list (347)	million (295)	stock (259)
Sports	play (1491)	win (1054)	game (947)	year (938)	match (918)
				Unique frequent words	

Moreover, rather than the common frequent words like ‘percentage’, ‘year’, we are interested in the words that appear the most frequently in one topic than in the others as such words are highly descriptive. We then calculated the average frequency of each of these words in different topics. The resulting figures are attached to Appendix A. Most frequent words in one topic have very low frequencies in other topics, which inspires us to propose a feature selection method that only keeps the most frequent keywords in each topic in the bag of words and discards those words in lower frequencies. The details of this method are provided in section 3.1.1.

3. Methodology

3.1 Feature Engineering

Feature engineering is one of the core parts of this project. The following steps address three main steps of the process: feature selection, text representation and imbalanced class handling.

3.1.1 Feature selection

Based on our analysis in section 2, we now consider a feature selection strategy to keep the keywords in each topic and discard words in lower frequency. This strategy is of importance to reduce the noise in the sample and also to increase the computational efficiency.
For each topic in the training set, we extract the most frequent n words. Then we combine these keywords in each topic as a keyword list L and iterate all the article words and only keep those words occurring in the keyword list. After that, using the same keyword list L, we also iterate the article words in the test set to extract the high frequent words to ensure the consistency of feature space.

For instance, if we set $n = 10$ and have 11 topics (consider irrelevant as one topic), we will have at most $10 \times 11 = 110$ keywords in total. Since different topic groups can have the same keywords, the total keywords will always be less than 110. However, having 110 words to learn 9500 articles in 10 topics is insufficient. Therefore, we consider n = 10, 20, 50 and 100 to create different keyword feature space and for each model, search the optimal n with regards to accuracy performance.

3.1.2 Text representation

The data set was preprocessed in a way that stopwords, tense and tone were removed, we further removed some meaningless symbols such as "_" and then used two feature creation methods, often referred to as bag of words or BoW[3], to transform articles into vectors of numbers.

The first method is word count, that is, representing a term using it's frequency count in the corpus. This approach is potentially influenced by the size of a class and length of articles because larger classes with longer articles tend to dominate the input matrix. As discussed in section 2.2, our training set is class unbalanced with the length of articles varies considerably, which suggests this approach of text representation might not perform ideally.

Another approach that aims to address the above issue is the TF-IDF method, details of which is discussed in Appendix. With TF-IDF, the value of a term in a vector is penalised by the number of articles it appeared in.

3.1.3 SMOTE to solve imbalanced class

From the distribution plot above, we observe a significant imbalance issue between different topics and IRRELEVANT topic occupies nearly half proportion (49.8%) in the training data set. It is critical to solve this issue as this hinders learning characteristics from the other 10 topics. Since under-sampling will lead to loss in information, we propose a Synthetic Minority Over-sampling Technique (SMOTE) to deal with this issue. The details of SMOTE algorithm and its discussion is available at Appendix B.

3.2 Model Training

Four classifiers are trained in this assignment using decision tree, logistic regression, support vector machine and naive bayes. The training processes follow the same pattern: starting with a base model and then perform hyperparameter tuning on the basic model. 30% data points were sampled out as validation sets to avoid overfitting, the optimal hyperparameters of each model are presented in section 3.4. Since correctly predicting each class is equally critical for this task, accuracy is a proper metric for evaluating the models, the results are discussed in section 3.5.

3.2.1 Decision Tree

Decision tree will be a classification model with low computation cost and since we are able to extract keywords by topics, we expect the decision tree model will have a good performance by identifying the keywords in each article. However, decision trees are more preferred to predict binary values rather than multiclass and decision trees are easy to overfit so pruning should be considered in the tree model.

The following five parameters are tuned in the decision tree model.

- "*max_depth":* The maximum depth of the tree
- *"min_samples_split":* The minimum number of samples required to split an internal node
- *"min_samples_leaf"*: The minimum number of samples required to be at a leaf node
- *"splitter"*: The strategy used to choose the split at each node

- *"min_weight_fraction_leaf"*: The minimum weighted fraction of the sum total of weights (of all the input samples) required to be at a leaf node

After tuning the model, the decision tree reached an accuracy of 0.94 in the training set and 0.73in the validation set.

3.2.2 Logistic regression

The data matrix is very high dimensional (35817 raw features), and sparse (with most values being 0). For dataset like this, linear models like Logistic Regression often works well[6].

First, the basic logistic regression model achieved an accuracy of 0.75 on the training set using the word count representation of features. Then SMOTE was performed and TF-IDF was used for hyperparameter tuning. The hyperparameters are:

- *"min_df":[1-5]*. The minimum number of articles a word has to be appear in in order to be considered when representing the feature vector of an article
- *"C":[0.0001, 0.01, 0.1, 1, 10]*. Inverse of regularization strength, smaller values of C specify stronger regularization
- *"solver"*: Algorithm to use in the optimization problem
- *"class_weight": ["balanced", None]*. "balanced" will adjust weights inversely proportional to class frequencies, i.e penalise large class.
- *"multi_class": ['ovr', 'multinomial']*.
    - *"ovr"*: one-versus-rest, essentially a binary problem is fit for each label
    - *"multinomial"*: the loss function is the multinomial loss fit across the entire probability distribution
- *"Penalty":["l1", "l2"]*. Used to specify the norm used in the penalization.

After the tuning, logistic regression achieved an accuracy of 0.97 on both the training set and the validation set, which suggests that the model performs the same on unseen articles.

3.2.3 Naive Bayes

Multinomial Naive Bayes (MNB) classifier can handle arbitrary term frequencies and update posterior probabilities from training data observations, which fits the text classification well as posterior probabilities are associated with maximum-likelihood estimates and class-conditional probabilities. Moreover, MNB is computationally efficient when handling large datasets.

The initial accuracy of the model on the training set is 96.23%. We conducted TruncatedSVD on the input matrix to reduce the dimension, and achieved only an accuracy of 75% on the validation set. Hence, we gave up using TruncatedSVD for feature selection and instead continued with selecting "keywords" method. The hyperparameters tuned are:

- *"min_df"*: The minimum number of articles a word has to appear in in order to be considered when representing the feature vector of an article.
- *"alpha"*: Additive smoothing parameter.
- *"class_prior"*: Prior probabilities of the classes. Because we assume the topic classes are in uniform distribution, we set the prior probabilities identical for all topics.
- *"fit_prior":* Whether to learn class prior probabilities or not.

After tuning, the best MNB classifier achieved an accuracy of 96% on the validation set.

3.2.4 Support vector machine

Support vector machines (SVM) is a very suitable algorithm for classifying text materials because text data are high dimensional and sparse. SVM identifies the support vectors for the classification and then obtains the maximum margin hyperplane from the support vectors. This form of results and the deletion of non-support vectors will be useful to avoid overfitting in high dimensional data (Joachims 1998). Additionally, Kivinen et al (1997) also shows that algorithms with error bounded such as SVM, will be suitable for sparse dataset.

We have three parameters that need to be tuned: *Decision_function_shape*, *C* and *Gamma*. *Decision_function_shape* determines the form of hyperplane. If we use 'ovo' we will have 11 hyperplanes and if we use 'ovr', there will be 55 hyperplanes in total. *C* will be the regularization parameters on the optimization process. The larger the *C* value, the smaller strength of regularization, which implies SVM will accept smaller margins which classify the training set more correctly. *Gamma* determines the radius to select supporting vectors. With a larger gamma value, there will be less supporting vectors selected. The details of tuning is given in the section 3.3. After tuning, the best SVM model is having 0.96 accuracy in the SMOTE training set and 0.83 in the validation set.

Despite there are lots of advantages of SVM, there is one critical limitation: SVM may not be practical for the recommendation step because of the difficulties to map the prediction output to the probability. Since we are ensembling SVM with the other three models, we can not use other measures for recommendation such as the distance to hyperplane. Although

there might be inconsistency between probability and prediction, we will still enable the probability option in the SVM model which trained a sigmoid function to map output to probability according to Platt (1999).

3.3 Hyperparameter Tuning

3.3.1 Grid search using pipeline

Grid search over hyperparameters was performed on each of the above models to achieve the best performance on the training set. In order to ensure a valid cross validation result, we used the pipeline class of sklearn to chain text representation steps with model training steps inside of the cross validation loop.

3.3.2 Optimal hyperparameters

After obtaining the best parameter on the training set, each model was tested on their validation sets to avoid overfitting. The optimal hyperparameters for each model are shown below:

Model	**Text represent method**	**Classifier**
Decision Tree	Tf-idf: min_df = 1	Max_depth = 128, min_samples_split = 2, min_samples_leaf = 1
Logistic Regression	Tf-idf: min_df = 4	C=0.001, class_weight = None, multi_class: 'multinomial', penalty = 'l2', solver = 'lbfgs'
Naive Bayes	Tf-idf: min_df = 1	alpha = 1, class_prior = None, fit_prior = True
SVM	Tf-idf: min_df = 1	C = 10, gamma = 0.0001, decision_function_shape = 'ovo'

3.4 measurement metrics

We considered four different metrics for this classification task: accuracy, precision, recall and F1 score. The details of these four measures are given in the table below. Classifying each group correctly is equally important in this task and we care about both the proportion of articles in one predicted topic that is correct (Precision) and how likely the actual articles in each topic is correctly classified (Recall). In addition, because irrelevant articles take a large proportion of observation, the precision score will be affected significantly and consequently affect the F1 score. Therefore, we will stick with the most standard accuracy measures as our benchmark for model selection and ensemble.

Accuracy	$1/n \sum I [y\_predict = y\_true]$	Standard measures
Precision	TP/ (TP+FP)	Proportion of articles in predicted topic are correct

Recall	TP/ (TP+FN)	Proportion of articles in one topic are correctly classify
F1 score	2 Precision * Recall / (Precision + Recall)	Balance between precision and recall

3.5 Ensemble method and final model

As we have discussed above, each model has its advantages and limitations. Thus, we consider ensemble all the models above as our final model for prediction and recommendation. Ensemble method is an approach combining different models and votes based on the prediction of those models, which can effectively reduce variance and improve the performance compared to single models.

A 2-level structure is also applied to the ensemble model (Fig 4.1). In the first step, the models trained to distinguish relevant and irrelevant topics are used, and all relevant articles are sent to the second layer to predict the categories of all relevant articles and their related probabilities.

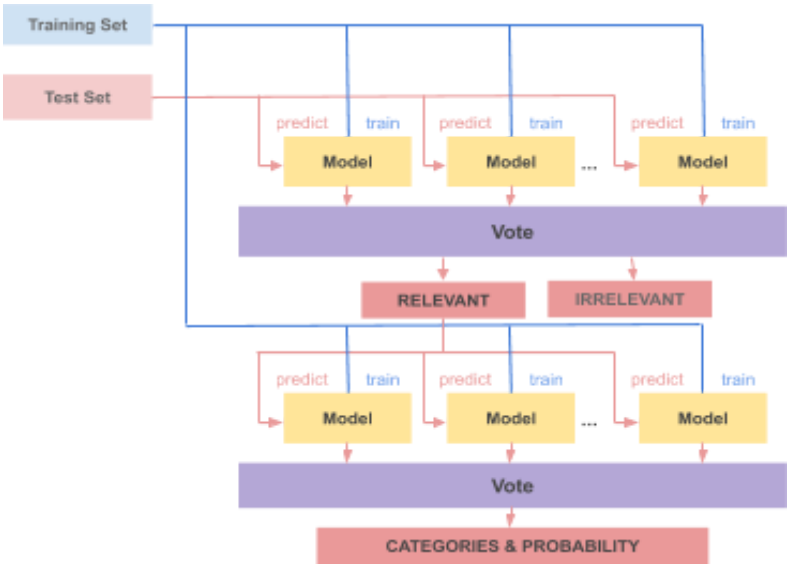


Figure 4.1 The structure of ensemble model

The model is built from scratch and uses a simple ensemble strategy. In the first layer, 3 models are used to vote, (1) a logistic model trained with full article word features, (2) a naive bayes classifier trained with full article word features, and (3) a support vector machine trained with full article word features. In the vote of relevant and irrelevant articles, a simple vote strategy without weight is used so the predicted outcome can be written as

$$\hat{y} = \arg\max_{\hat{y} \in \{0,1\}} \sum_{i=1}^{3} \mathbb{1}(\hat{y}_i = \hat{y})$$

Where {0,1} stands for *{'RELEVANT', 'IRRELEVANT'}*, $\hat{y}$ stands for the predicted outcome of model i and 1(.) is identity function.

For the second layer, 9 models are chosen to vote. The weight of the votes is set to be binary. The weights equal to 1 if the model is having good performance in the predicted topic according to the validation set and equals to 0 otherwise. See Appendix C for the detailed model performance in different topics.

$$\hat{y} = \arg\max_{\hat{y} \in C} \sum_{i=1}^{9} \mathbb{1}(\hat{y}_i = \hat{y}) w(\hat{y}_i, m_i)$$

where C is the set of all topic categories. The weight function $w(\hat{y}, m)$ represents the weight that model m gives prediction $\hat{y}$. If $\hat{y}$ is not in the accurate predicted categories of m, $w = 0$, otherwise, w = 1.

4 Results

4.1 CV results on training set

Topic	Precision	Recall	F1-score
ARTS CULTURE ENTERTAINMENT	0.77	0.73	0.75
BIOGRAPHIES PERSONALITIES PEOPLE	0.70	0.74	0.72
DEFENCE	0.86	0.90	0.88
DOMESTIC MARKETS	0.84	0.74	0.79
FOREX MARKETS	0.49	0.77	0.60
HEALTH	0.82	0.84	0.83
MONEY MARKETS	0.74	0.62	0.68
SCIENCE AND TECHNOLOGY	0.80	0.69	0.74
SHARE LISTINGS	0.90	0.90	0.90
SPORTS	0.95	0.98	0.96

4.2 Final results output table

Topic	Precision	Recall	F1-score
ARTS CULTURE ENTERTAINMENT	0.43	1.00	0.60
BIOGRAPHIES PERSONALITIES PEOPLE	0.64	0.47	0.56
DEFENCE	0.90	0.69	0.78
DOMESTIC MARKETS	0.50	0.50	0.50

FOREX MARKETS	0.59	0.79	0.13
HEALTH	0.67	0.57	0.66
MONEY MARKETS	0.64	0.64	0.17
SCIENCE AND TECHNOLOGY	0.00	0.00	0.00
SHARE LISTINGS	0.56	0.71	0.63
SPORTS	0.95	0.97	0.29

4.3 Final article recommendation and evaluation

Topic	**Recommendations**	**Precision**	**Recall**	**F1-score**
ARTS CULTURE ENTERTAINMENT	9604, 9703, 9789, 9830, 9834, 9933, 9952	0.43	1.00	0.60
BIOGRAPHIES PERSONALITIES PEOPLE	9526, 9575, 9582, 9645, 9758, 9811, 9854, 9878, 9940, 9988	0.70	0.47	0.70
DEFENCE	9559, 9576, 9607, 9616, 9670, 9721, 9770, 9773, 9842, 9987	0.90	0.69	0.95
DOMESTIC MARKETS	9833, 9994	0.50	0.5	0.67
FOREX MARKETS	9529, 9531, 9542, 9554, 9564, 9647, 9751, 9892, 9894, 9943	0.40	0.083	0.50
HEALTH	9621, 9661, 9807, 9873, 9911, 9926, 9929, 9947, 9978, 9982	0.80	0.57	0.80
MONEY MARKETS	9550, 9587, 9599, 9665, 9677, 9753, 9808, 9883, 9916, 9959	0.64	0.10	0.64
SCIENCE AND TECHNOLOGY	9617	0.00	0.00	0.00
SHARE LISTINGS	9518, 9601, 9655, 9666, 9667, 9668, 9715, 9972, 9999	0.56	0.71	0.71
SPORTS	9608, 9695, 9701, 9736, 9757, 9801, 9809, 9818, 9862, 9931	1.00	0.17	0.91

5 Discussion

In this project, we trained 5 machine learning classifiers: decision tree, logistic regression, naive bayes, support vector machine and an ensemble model of them as the final classifier. Each model performs slightly differently in a way that they achieve high accuracy on different topics, which is the motivation of the final ensemble model. A summary of the

accuracy of different models and is listed below, the features selected for each model is listed in Appendix:

1. Decision tree: 0.65, high accurate topics: SPORTS
2. Logistic regression: 0.79, high accurate topics: ARTS CULTURE ENTERTAINMENT,DOMESTIC MARKETS, SPORTS, SHARE LISTINGS, FOREX MARKETS, DEFENCE
3. Naive bayes: 0.72, high accurate topics:SHARE LISTINGS, DOMESTIC, MARKETS, ARTS CULTURE ENTERTAINMENT, SPORTS, DEFENCE,
4. SVM: 0.78, high accurate topics: DOMESTIC MARKETS, SPORTS, FOREX MARKETS, DEFENCE, SHARE LISTINGS
5. Final ensemble model: 0.81

There are three metrics we have considered to evaluate the performance in each article topic: Precision, Recall and F1 score. We also evaluate the model's performance by accuracy. Accuracy is preferred to compare models' overall performance as it is a more conventional measure and we are assigning equal weights penalty when classifying any articles into incorrect topics. In terms of the evaluation on different topics, precision measure is prefered over Recall and F1. This is due to the recommendation purpose for this classification task. All the readers can only read at most 10 articles, so maintaining the high accuracy in the articles recommended is much more important than predicting other articles correctly. Therefore, Precision is the preferred measure over recall and since F1 will depend on recall, it will also be less preferred compared to Precision.

We also tried to implement deep learning methods including linear neural networks and long-short term memory (LSTM). However, linear neural networks cannot accurately classify articles because of its simple structure, and LSTM fails to predict article categories since the raw data does not preserve enough context information.

6 Conclusion

An ensemble of four different classifiers is developed in this project to combine their advantages. This method is supported by the fact that each model is better at predicting specific topics. While each model made mistakes on some topics, the final model is able to achieve an ideal accuracy on unseen articles.

References

[1] Chawla, N, Bowyer, K, Hall, L & Kegelmeyer, W 2002, *'SMOTE: Synthetic minority over-sampling technique'*, Journal Of Artificial Intelligence Research, vol. 16, pp. 321–357.

[2] L. Wu, S. C. H. Hoi and N. Yu, *'Semantics-Preserving Bag-of-Words Models and Applications',* in IEEE Transactions on Image Processing, vol. 19, no. 7, pp. 1908-1920, July 2010.

[3] Rajaraman, A., & Ullman, J 2011. '*Data Mining. In Mining of Massive Datasets. Cambridge*', Cambridge University Press, pp. 1-17.

[4] Joachims T. 1998, '*Text categorization with Support Vector Machines: Learning with many relevant features'* In: Nédellec C., Rouveirol C. (eds) Machine Learning: ECML-98. ECML 1998.

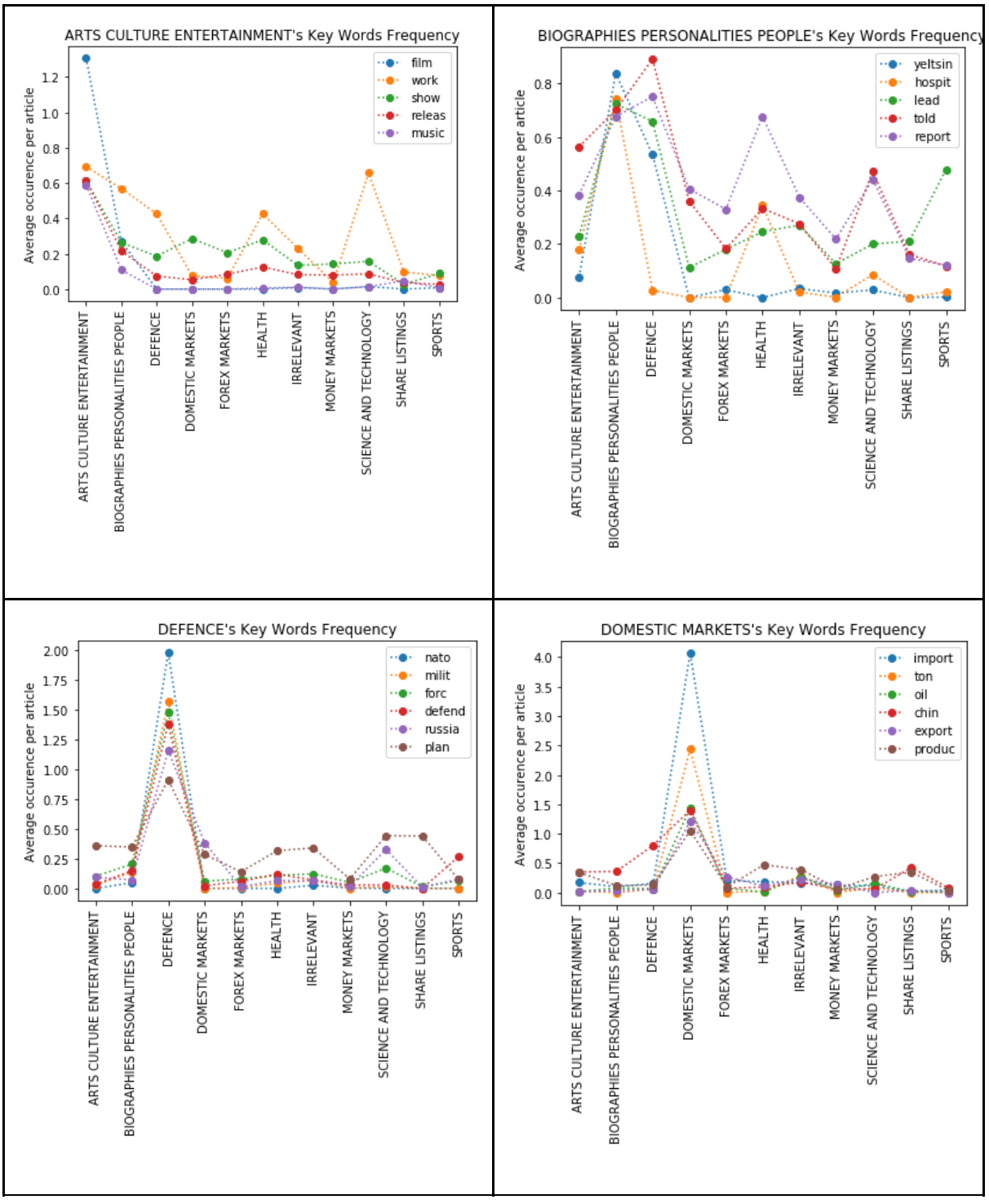
[5] Kivinen, J, Warmuth, M & Auer, P 1997, '*The Perceptron algorithm versus Winnow: linear versus logarithmic mistake bounds when few input variables are relevant*', Artificial Intelligence, vol. 97, no. 1-2, pp. 325–343.

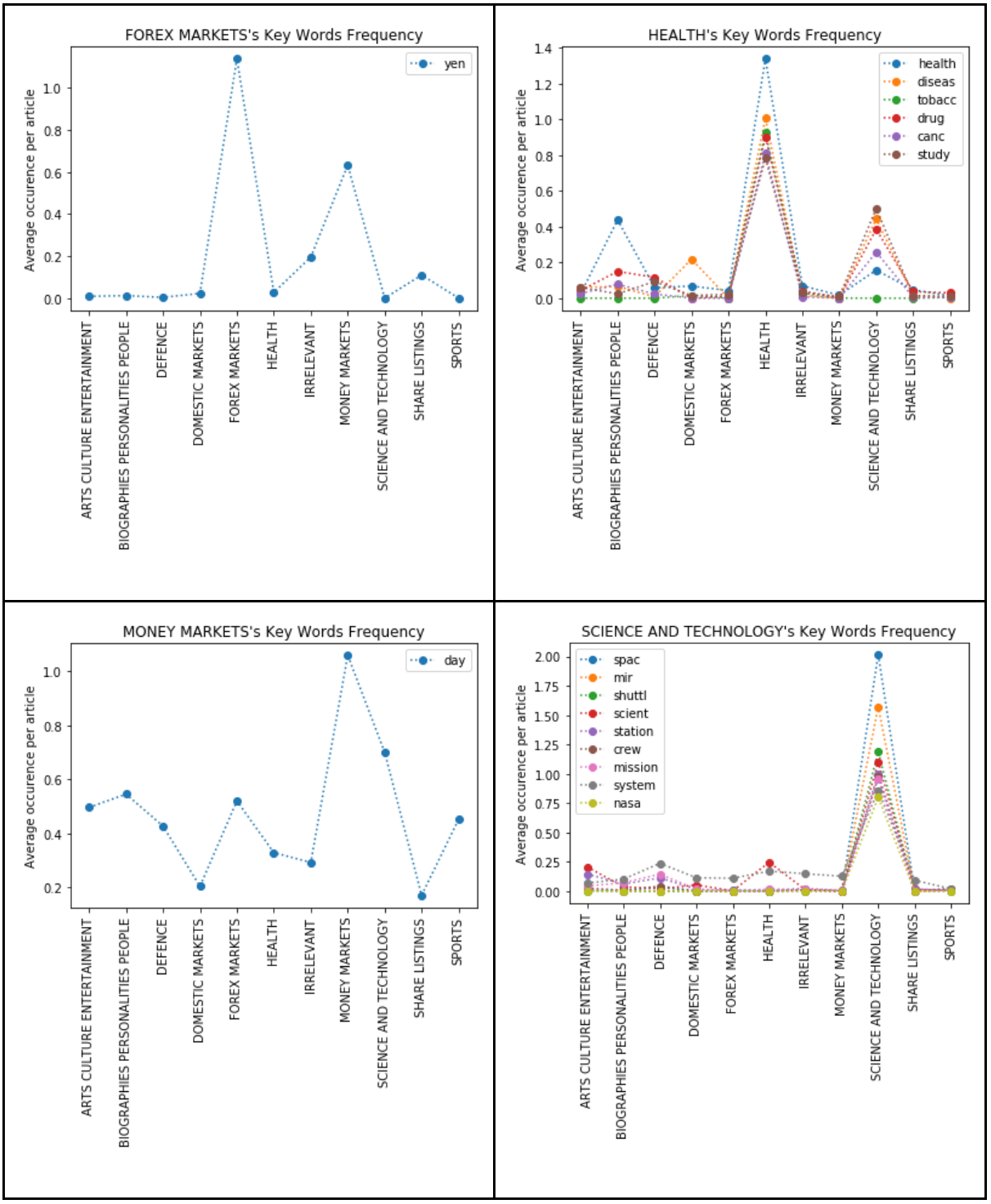
[6]Platt, C.J 1998, *'Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods',* MIT Press, pp. 61-74.

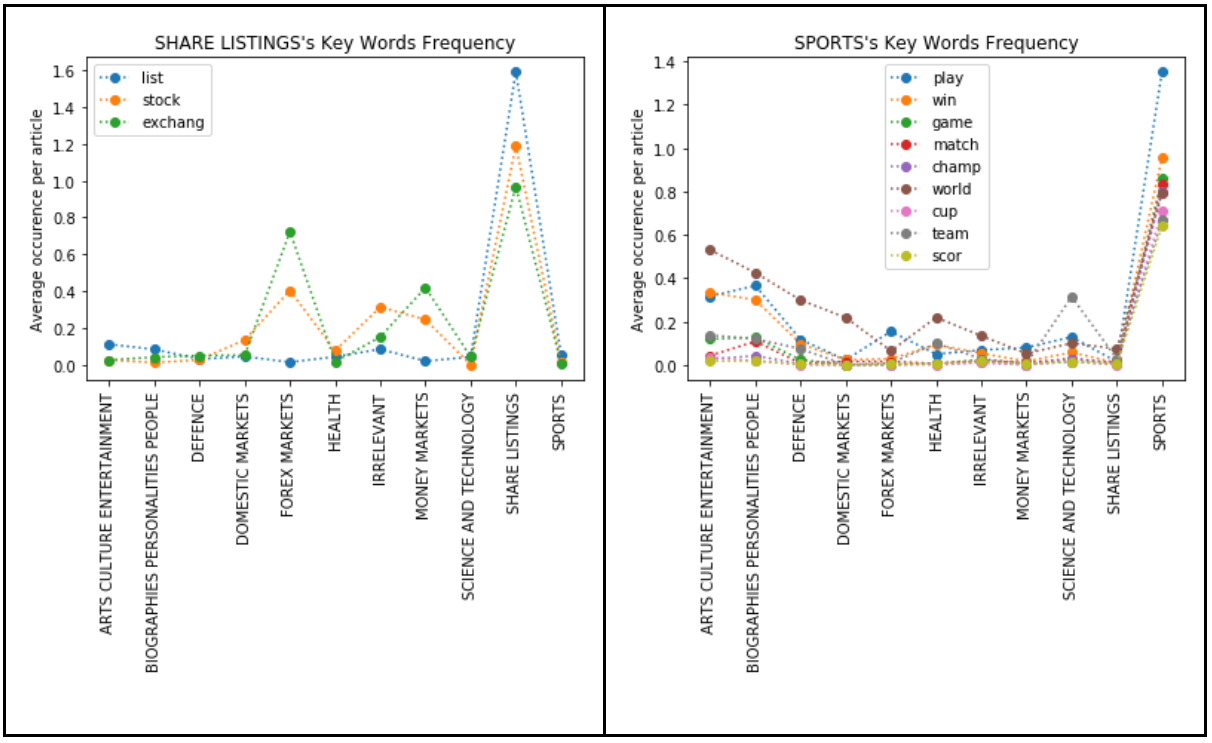
[7] Marques, YB, de Paiva Oliveira, A, Ribeiro Vasconcelos, AT & Cerqueira, FR 2016, *'Mirnacle: machine learning with SMOTE and random forest for improving selectivity in pre-miRNA ab initio prediction'*, BMC bioinformatics, vol. 17, no. Suppl 18, p. 474.

[8] Juanjuan Wang, Mantao Xu, Hui Wang & Jiwu Zhang 2006, *'Classification of Imbalanced Data by Using the SMOTE Algorithm and Locally Linear Embedding'*, in 2006 8th international Conference on Signal Processing.

Appendix A


ARTS CULTURE ENTERTAINMENT's Key Words Frequency
Average occurence per article
film
work
show
releas
music
ARTS CULTURE ENTERTAINMENT
BIOGRAPHIES PERSONALITIES PEOPLE
DEFENCE
DOMESTIC MARKETS
FOREX MARKETS
HEALTH
IRRELEVANT
MONEY MARKETS
SCIENCE AND TECHNOLOGY
SHARE LISTINGS
SPORTS
BIOGRAPHIES PERSONALITIES PEOPLE's Key Words Frequency
Average occurence per article
yeltsin
hospit
lead
told
report
DEFENCE's Key Words Frequency
Average occurence per article
nato
milit
forc
defend
russia
plan
DOMESTIC MARKETS's Key Words Frequency
Average occurence per article
import
ton
oil
chin
export
produc


FOREX MARKETS's Key Words Frequency
yen
Average occurence per article
ARTS CULTURE ENTERTAINMENT
BIOGRAPHIES PERSONALITIES PEOPLE
DEFENCE
DOMESTIC MARKETS
FOREX MARKETS
HEALTH
IRRELEVANT
MONEY MARKETS
SCIENCE AND TECHNOLOGY
SHARE LISTINGS
SPORTS
HEALTH's Key Words Frequency
health
diseas
tobacc
drug
canc
study
MONEY MARKETS's Key Words Frequency
day
SCIENCE AND TECHNOLOGY's Key Words Frequency
spac
mir
shuttl
scient
station
crew
mission
system
nasa


SHARE LISTINGS's Key Words Frequency
Average occurence per article
list
stock
exchang
ARTS CULTURE ENTERTAINMENT
BIOGRAPHIES PERSONALITIES PEOPLE
DEFENCE
DOMESTIC MARKETS
FOREX MARKETS
HEALTH
IRRELEVANT
MONEY MARKETS
SCIENCE AND TECHNOLOGY
SHARE LISTINGS
SPORTS
SPORTS's Key Words Frequency
Average occurence per article
play
win
game
match
champ
world
cup
team
scor
ARTS CULTURE ENTERTAINMENT
BIOGRAPHIES PERSONALITIES PEOPLE
DEFENCE
DOMESTIC MARKETS
FOREX MARKETS
HEALTH
IRRELEVANT
MONEY MARKETS
SCIENCE AND TECHNOLOGY
SHARE LISTINGS
SPORTS

Appendix B

TFIDF

The value of a term in a document is calculated as follow:

$$TFIDF(w,d) = tf \cdot log(\frac{N_d + 1}{N_{d|w} + 1}) + 1$$

N_d: the number of documents in the training set

$N_{d|w}$: the number of documents in the training set that contains the word w

SMOTE

By finding out the line segments that join the minority class nearest neighbors, and then choosing samples in each minority topic, SMOTE algorithm creates synthetic examples along the line segments to get the minority topics over-sampled [4]. By doing this, SMOTE decreases the possibility of information loss [5]. However, J.Wang, Xu, H.Wang & Zhang [6] have a strict assumption on SMOTE that a local space would be positive or under the minority class if it is between two positive instances, which might conflict with the training text data when some words are not linear separable.

After applying the SMOTE algorithm, the size of training data set comes to $4734 \times 11 = 52,074$ lines. Though the imbalance is solved, a new issue comes up that the constructed model could be over-fitting since a lot of training lines are duplicated, which means there exists some vectors, for each of them, the model learns it for multiple times.

Naive Bayes

In this project, assuming that t_r is the r^{th} topic, X is a transformed vector from a line in the training data with length d, we have the probability:

$$P(X|t_r) = \prod_{k=1}^{d} P(x_k|t_r)$$

where $P(x_i|t_r)$ is given the topic t_r, what is the probability of x_i be seen.

After training the model, the algorithm summarises some patterns from training vectors. And now assuming after transformation, a line in the test set has n patterns ϕ_i in its vector v, we have:

$$P(t_r|v) = \prod_{i=1}^{n} \frac{P(\phi_i|t_r)P(t_r)}{P(\phi_i)}$$

Appendix C Chosen Model

Model	Training Feature	Test Feature	Data Representation	Accurate topics
Logistic Regression	Top 50 keywords	Top 50 keywords	TF-IDF	ARTS CULTURE ENTERTAINMENT, DOMESTIC MARKETS, SPORTS, SHARE LISTINGS, FOREX MARKETS, DEFENCE
Naive Bayes	Full article words	Top 50 keywords	TF-IDF	SHARE LISTINGS, DOMESTIC MARKETS, ARTS CULTURE ENTERTAINMENT, SPORTS, DEFENCE, FOREX MARKETS, SCIENCE AND TECHNOLOGY
Naive Bayes	Full article words	Top 100 keywords	CountVectorizer	SHARE LISTINGS, DOMESTIC MARKETS, SPORTS, HEALTH, MONEY MARKETS, DEFENCE
Naive Bayes	Top 100 keywords	Top 50 keyword	TF-IDF	BIOGRAPHIES PERSONALITIES PEOPLE
SVM	Top 100 keywords	Top 100 keywords	TF-IDF	DOMESTIC MARKETS, SHARE LISTINGS, ARTS CULTURE ENTERTAINMENT, SPORTS, DEFENCE, FOREX MARKETS, BIOGRAPHIES PERSONALITIES PEOPLE
SVM	Top 100 keywords	Full article words	TF-IDF	DOMESTIC MARKETS, SPORTS, FOREX MARKETS, HEALTH
Decision Tree	Full article words	Full article words	TF-IDF	SPORTS
Decision Tree	Top 100 keywords	Top 100 keywords	TF-IDF	SPORTS
Decision Tree	Full article words	Full article words	CountVectorizer	SPORTS

Appendix D Project link:

https://github.com/HenryLiangzy/COMP9417_Project