

# Data science midterm examination report

Jirayuwat Boonchan  
6432023321

2110446  
Data science and Data engineering

## Table of content

Table of content.....	2
Introduction.....	3
Data preparation.....	4
Model.....	5
Model architecture.....	5
Training step and hyper-parameter tuning.....	5
Result.....	6
Discussion.....	7
An imbalance dataset.....	7
Scientific name.....	7
Small dataset.....	8
Conclusion.....	8

## Introduction

This is a report which studies classifying topics of research paper using machine learning and data science methodology. The dataset includes 454 data points and each data point has its title and abstract. The goal is to classify the labels of each paper which can be classified into multiple labels (known as multi-label problems). The original title and abstract is transformed into a meaningful vector and fed into a machine learning algorithm to classify its label and the original problem which is multi-label classification is transformed into binary classification on each class instead.

## Data preparation

The preparations are following the below steps.

1. The title and abstract are combined by white space (" ").
2. Remove special characters and digits.
  - accept only a-z (lower case) and A-Z (upper case)
3. Remove the publisher's name
  - Publisher's list: IOP Publishing, IEEE, Elsevier Ltd., Springer International Publishing, John Wiley & Sons, Association for Computer Machinery, Nature Singapore Pte Ltd., KSME, JSME, American Chemical Society, etc.
  - The reason for removing it is that publisher's names are not able to represent that kind of research. (as show in the following figure)

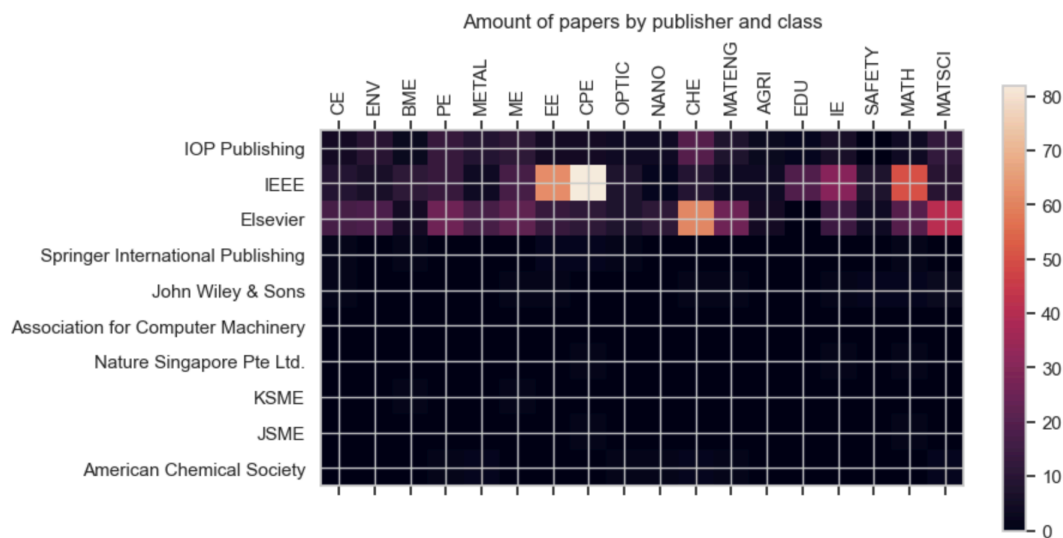


Figure1. Correlation between classes and publishers.

4. Apply lemmatization and remove stop words.
  - Lemmatization is the process of grouping the inflected form of the word into a single word (e.g. dogs will be dog)
  - Stop words are the word list that is used commonly and is meaningless in the context of NLP.
  - Tokenization, lemmatization, and stop word lists are implemented in the [nltk library](#).
5. Transform the text into a vector
  - Text vectorization is the process of transforming text into a vector that is meaningful and representative. This is a very important process so the computer can take this vector in the mathematics process in the next step
  - In this preparation, The text is embedded via the [Universal Sentence Encoder model by Google](#). The model is trained to embed sentences into a 512-dimension vector which is meaningful.

After all the steps above, The title and abstract of each paper will be transformed into 512-dimension vectors and fed into the classification model in the next steps.

Note The processed data is stored in github and accessible via this [link](#)

## Model

Due to a lack of data, the model that was trained to classify 18 classes (multilabel problem) gives the worst performance. To solve this problem, The model is trained class by class (binary classification problem) and trained by all the data points.

## Model architecture

The model in this report is a voting ensemble classifier that includes a Support Vector Classifier, Logistic Regression, and Gaussian Naive Bayes. The voting algorithm is a hard vote in which the result's prediction is the maximum counting of votes among all the models.

The final model configuration

1. Support Vector Classification
  - *Class weight*: balanced
  - *Kernel*: linear
2. Logistic Regression
  - *Class weight*: balanced

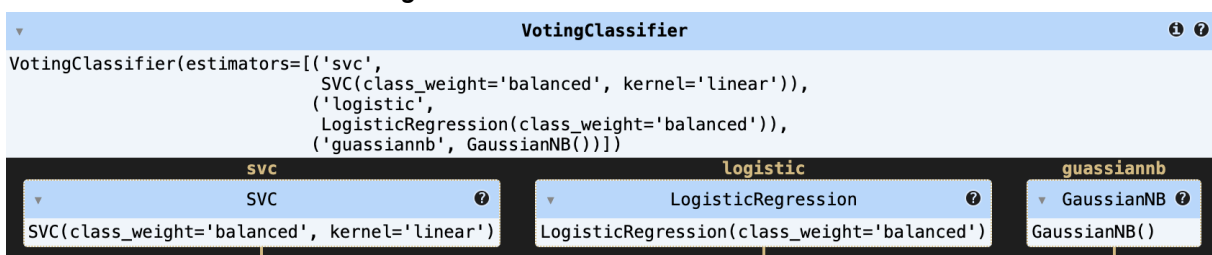


Figure2. Model architecture

## Training step and hyper-parameter tuning

In the hyper-parameter tuning phase, each class has its own instance of model with the same structure and the model is trained by all the data points without oversampling (uses `class_weight='balanced'` in order to fix unbalanced dataset) and evaluated by the trim-mean of 5-fold validation F1 score which is more reasonable than splitting dataset in this case. Furthermore, the model is tuned to its parameter within the following list by random search algorithm.

1. Support Vector Classifier
  - *Class weight*: balanced, and none
  - *Kernel*: linear, poly, rbf, and sigmoid
2. Logistic Regression
  - *Class weight*: balanced, and none
  - *Penalty*: L1, L2, elastic net, and none

After the hyper-parameter tuning phase, the best model's configuration is the same as the Model architecture topic above, and set this configuration to the model.

## Result

In the report, the model for each class has the same architecture and is independently evaluated as shown in the following table.

	▼	Ratio of true	F1-score ▼	Precision ▼	Recall ▼	Accuracy ▼
	SAFETY	4.85%	30.26%	31.11%	30.0%	93.41%
	PE	18.94%	57.17%	48.35%	68.63%	79.85%
	OPTIC	6.83%	37.14%	35.83%	44.44%	89.34%
	NANO	7.05%	45.55%	33.11%	73.02%	88.28%
	METAL	15.86%	61.62%	47.03%	83.97%	82.72%
	ME	19.82%	43.1%	34.11%	61.11%	68.75%
	MATSCI	26.21%	66.31%	56.81%	78.86%	78.75%
	MATH	24.67%	62.95%	54.28%	76.15%	78.02%
	MATENG	14.1%	48.76%	36.92%	70.94%	78.39%
	IE	16.3%	52.24%	42.66%	68.1%	79.85%
	ENV	13.0%	51.7%	45.7%	68.43%	84.62%
	EE	25.99%	69.6%	65.29%	74.15%	83.45%
	EDU	7.05%	43.1%	32.03%	62.7%	87.55%
	CPE	31.28%	71.28%	65.98%	77.75%	80.88%
	CHE	38.99%	89.13%	89.65%	87.86%	91.55%
	CE	11.45%	48.43%	43.38%	54.55%	86.77%
	BME	7.93%	50.5%	49.47%	52.38%	92.31%
	AGRI	4.41%	30.51%	38.1%	25.0%	94.51%

Figure3. Evaluation matrix

From the above table, Accuracy does not reflect the real performance, because most of the class is an imbalance class, so the model is evaluated by F1-score. The most F1-score class is Chemical engineering paper (CHE class). On the other hand, Agriculture paper (AGRI) and Safety paper (SAFETY) are the worst F1-score.

The F1-macro score in 5-folds validation, public score, and private score in kaggle is 53.30%, 57.63%, and 52.90% respectively.


Submission and Description		Private Score ⓘ	Public Score ⓘ
 <b>submission.csv</b> Complete (after deadline) · 16s ago · test submission			
		<b>0.52906</b>	<b>0.57626</b>

Figure4. Submission score on Kaggle.com

## Discussion

From the above result and some analysis, The problem can be classified into 3 groups which are an imbalance dataset, scientific name, and small dataset.

### An imbalance dataset

From the result topics, F1-score has a strong relation with a ratio of true class(indicating how imbalance labels are) as shown in Figure5. Then, it's potentially increase the F1-score of these classes if I could correct some data for the class that has a low ratio of true class, e.g. AGRI, SAFETY, OPTIC, NANO, EDU, and BME.

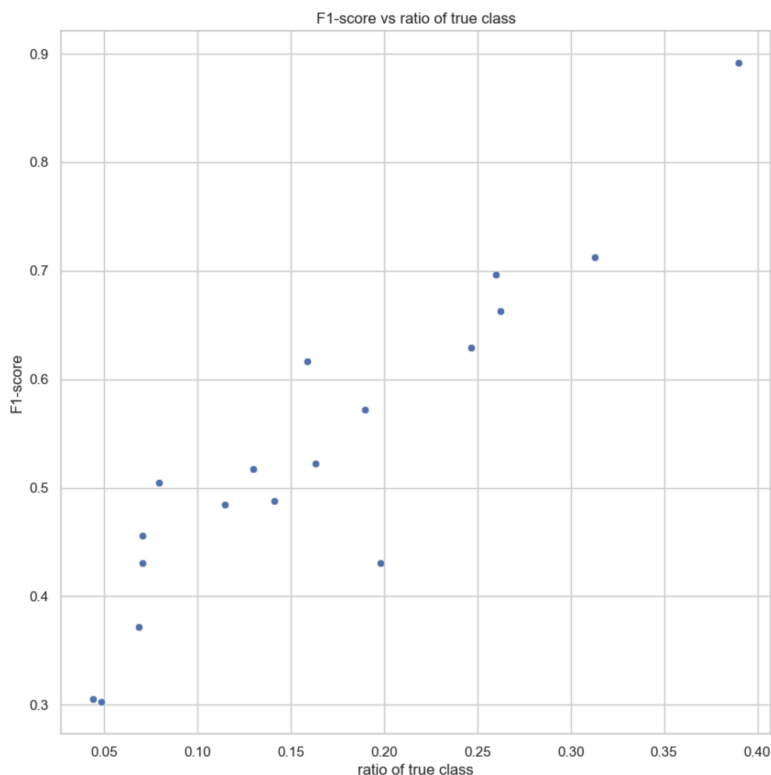


Figure5. F1-score vs Ratio of true class

### Scientific name

The text includes scientific names such as chemical formula, scientist name, specific formula name, etc. To solve this problem, the scientific names dictionary must be created to prevent wrong tokenization and the text embedding must use the model which is trained for this task.

```
Operando DRIFTS B V type acidity W0x SiO catalyst prepared different tungsten loading wt W pretreatment atmosphere N H
```

Figure6. Example text that is tokenized incorrect.

## Small dataset

The main task of this problem is multi-label which includes totally 18 classes but the dataset contains only 454 data points. From the experiments and research, the current state-of-the-art for text classification is Transformer, by the way, there is no model that is trained for this task yet and I have not enough data to fine-tune the model to achieve preferred performance. So, I need to focus on classical machine learning algorithms and ensemble methods.

## Conclusion

In this report, The multi-label classification is transferred into binary classification problems of each class and the hard voting ensemble model which includes Support Vector Classifier, Logistic Regression, and Gaussian Naive Bayes is used to classify the category of the paper. According to this approach, The models achieve 52.90% of F1-macro score in private score.