IFN 645 Large Scale Data Mining
Major Assignment

Name	Student ID
Pallav Chamoli	N10635700
Tan Zhi On	N9641556

25/10/2021

Table of Completion

Task	Status of Completion
Task 1	Complete
Task 2	Complete
Task 3	Complete

Task Specification

1. Task 1: Association mining in Java (13 Marks)

A bank has conducted a marketing campaign via phone calls to promote their new products including a term-deposit product. After the campaign, the bank wants to analyse the data collected in the campaign in order to get a better understanding to their customers.

- Dataset

  Download the datasets **`bank.arff`**, **`bank_no.arff`**, and **`bank_yes.arff`** from the Blackboard.

  `bank.arff` is the entire dataset collected in the campaign which consists of 45,211 records. The other two datasets are subsets of the entire dataset containing customers in each class. Each record in the datasets is about one customer described by 11 attributes. To help you understand the data, the following table provides you with the description to each of the attributes.

Attribute	Description
age	Customer age
job	Customer job type
marital	Customer marital status
education	Customer education status
default_credit	Whether customer has credit in default or not
balance	Customer bank account balance
housing	Whether customer has housing/home loan or not
loan	Whether customer has personal loan or not
call_duration	Phone contact duration in seconds, e.g., 500-1k indicates that the phone talk with the customer lasts 500 to 1000 seconds.
past_marketing	Outcome of last marketing to this customer, e.g., failure, indicates that the last marketing was unsuccessful to this customer.
subscribed	After this marketing campaign, whether the customer subscribed the term-deposit product or not. This attribute divides customers into two classes, i.e., yes-class and no-class.

In the context of this task, an item is an attribute with a specific value, e.g., age=40s, and a pattern is a set of items, e.g., {default_credit=no, loan=no, age=21-30s} is a size3 pattern. The patterns generated from the bank dataset describe the characteristics or behaviour of the customers in that dataset.

After the marketing campaign, the bank wants to obtain some information from the dataset. Specifically, they want to know the popular patterns in each customer class and also want to know the associations between patterns and each class, i.e., the association rules that represent the implications from customer attributes to one of the classes. As a data analyst, you are required to develop a Java program that can generate patterns and association rules from the datasets to answer the following questions

Questions

1) Generate frequent patterns from the entire dataset (i.e., **`bank.arff`**) using two frequent pattern mining algorithms and compare their performance in terms of time efficiency. You may use 3 different minimum supports to do the comparison. Show your comparison result and choose the algorithm with better performance.

Ans.

In java program, we use 2 frequent pattern mining algorithms (apriori, fp-growth) with 3 different minimum support (0.2, 0.3, 0.4) and noticed that the algorithm FP-Growth performed better than apriori with different minimum support. The more we increase the minimum support the less time it takes for FP-growth algorithm to process the data

Min sup	**Apriori algorithm Time (ms)**	**Fp-growth algorithm Time (ms)**
0.2	638	123
0.3	276	107
0.4	170	115

2) Using the chosen algorithm in question_1), generate the top 5 most frequent size-3 patterns from the yes-class dataset (i.e., **`bank_yes.arff`**) and no-class dataset (i.e., **`bank_no.arff`**) separately, then compare the generated patterns from the two datasets and identify the same or different characteristics between the customers in the two classes.

Ans.

The top 5 most frequent size-3 pattern from "YES" dataset displays that total of 537 customer have default_credit = No and education = primary they are subscribed to the term-deposit with loan = No whereas in "NO" dataset when a customer have default_credit = No and education = primary they are not subscribed having call_duration = 100-500s. the similarity in both the dataset is whether the customer is subscribed or not they have default_credit = No and education = primary.

3) Generate the top 5 most frequent maximum patterns from yes-class and no class datasets separately, identify any similarity or differences between the two classes in terms of the maximum patterns.

Ans.

The top 5 frequent maximum patterns from yes-class and no-class displays that in both yes and no dataset the past_marketing = failure is very high for yes-class the occurrence is 618 and for no-class the occurrence is 4283.

4) Use three algorithms to generate frequent closed patterns from the entire dataset and compare their time efficiency.

Ans.

Minsup	Frequent close algorithm	Total time (ms)	Frequent closed patterns
0.3	AprioriClose	265	154
0.3	FPClose	131	154
0.3	Charm_bitset	50	154

5) Using the entire dataset, generate the top 10 most frequent association rules with subscribed=yes as the consequent and also the top 10 most frequent association rules with subscribed=no as the consequent. You can specify some appropriate minimum support and minimum confidence.

a. List the rules generated for each class.

Ans.

For subscribed = yes the top 10 rules that are generated are:

1. past_marketing=success ==> subscribed=yes #SUP: 978 #CONF: 0.6472534745201853
2. default_credit=no past_marketing=success ==> subscribed=yes #SUP: 978 #CONF: 0.6481113320079522
3. default_credit=no loan=no past_marketing=unknown call_duration=500-1k ==> subscribed=yes #SUP: 1061 #CONF: 0.3585670834741467
4. loan=no past_marketing=unknown call_duration=500-1k ==> subscribed=yes #SUP: 1079 #CONF: 0.35978659553184394
5. default_credit=no past_marketing=unknown call_duration=500-1k ==> subscribed=yes #SUP: 1214 #CONF: 0.34915156744319814
6. past_marketing=unknown call_duration=500-1k ==> subscribed=yes #SUP: 1242 #CONF: 0.350253807106599
7. default_credit=no loan=no call_duration=500-1k ==> subscribed=yes #SUP: 1428 #CONF: 0.39284731774415405
8. loan=no call_duration=500-1k ==> subscribed=yes #SUP: 1448 #CONF: 0.39358521337319924
9. default_credit=no call_duration=500-1k ==> subscribed=yes #SUP: 1614 #CONF: 0.3793184488836663
10. call_duration=500-1k ==> subscribed=yes #SUP: 1646 #CONF: 0.38005079658277535

For subscribed = no top 10 rules generated are:

1. marital=married ==> subscribed=no #SUP: 24459 #CONF: 0.8987653413684134
2. default_credit=no call_duration=100-500s ==> subscribed=no #SUP: 25538 #CONF: 0.8988139232041671
3. call_duration=100-500s ==> subscribed=no #SUP: 26044 #CONF: 0.900117508813161
4. default_credit=no loan=no past_marketing=unknown ==> subscribed=no #SUP: 27371 #CONF: 0.9015777858295728
5. loan=no past_marketing=unknown ==> subscribed=no #SUP: 27814 #CONF: 0.901997665066805
6. default_credit=no loan=no ==> subscribed=no #SUP: 32685 #CONF: 0.8726937761995034
7. default_credit=no past_marketing=unknown ==> subscribed=no #SUP: 32862 #CONF: 0.9077649788679871
8. loan=no ==> subscribed=no #SUP: 33162 #CONF: 0.8734427265783443
9. past_marketing=unknown ==> subscribed=no #SUP: 33573 #CONF: 0.9083849671257339
10. default_credit=no ==> subscribed=no #SUP: 39159 #CONF: 0.8820389224254437

b. Observe the rules and identify any redundant rules in each set of the rules. If there exist redundant rules, list them and state why you think they are redundant.

in subscriber = yes :

1. past_marketing=success ==> subscribed=yes #SUP: 978 #CONF: 0.6472534745201853
2. loan=no call_duration=500-1k ==> subscribed=yes #SUP: 1448 #CONF: 0.39358521337319924
3. default_credit=no call_duration=500-1k ==> subscribed=yes #SUP: 1614 #CONF: 0.3793184488836663
4. call_duration=500-1k ==> subscribed=yes #SUP: 1646 #CONF: 0.38005079658277535

in subscriber = No :

1. call_duration=100-500s ==> subscribed=no #SUP: 26044 #CONF: 0.900117508813161
2. loan=no ==> subscribed=no #SUP: 33162 #CONF: 0.8734427265783443
3. past_marketing=unknown ==> subscribed=no #SUP: 33573 #CONF: 0.9083849671257339
4. default_credit=no ==> subscribed=no #SUP: 39159 #CONF: 0.8820389224254437

These are redundant rules, because the top rules already consist of the combination of these redundant rule and single rules doesn't really display proper knowledge.

STATEMENT : ALL TASK HAS BEEN COMPLETED WITH JAVA CODE IN TASK 1 FOLDER

2. Task 2: Classification in Weka and Java (13 marks)

In Task 1, you have developed a Java program to generate patterns and association rules from the bank datasets to describe the behaviour of the customers in the dataset. In this task, you are required to write a Java program to classify the customers in the entire bank dataset **`bank.arff`**. Before writing your Java program, you are required to analyse the dataset in Weka to select 3 classification algorithms based on their classification performance in Weka. For evaluation, you can use the default 10-fold cross-validation.

2.1. Data analysis in Weka

For the following questions, you need to use the **`AttributeSelectedClassifier`** in Weka to analyse the data. Describe your working process. Provide evidence to justify your decision. The evidence can be tables to show performance comparison, screenshots, or some outputs from Weka.

1) Select 4 classification algorithms based on attribute analysis. You can choose the 4 algorithms from **`NaiveBayes`**, **`NaiveBayesMultinomial`**, **`IBk`**, **`PART`**, **`OneR`**, **`ZerR`**, **`J48`**, **`Randomforest`**. This question is not to choose attributes. This question is to choose algorithms based on their classification accuracy performance. You can use any evaluator, search method, or ranker to do it. Justify your decision.

Ans.

Algorithms	Classification Accuracy Performance
NaiveBayesMultinominal	88.30%
IBk	88.83%
PART	89.19%
OneR	89.28%
ZeroR	88.30%
J48	89.77%
Random Fores	88.82%
Naïve Bayes	89.24%

As we can see that using attributeSelectedClassifier with the algorithm there are not major difference between them. The 4 algorithm that we would be working with after the analysis are NaiveBayes, PART, J48, OneR. The reason of choosing these algorithms is that they produced high accuracy in less time.

2) Select 3 algorithms from the 4 algorithms chosen in question 1) of Section 2.1 based on cost analysis. You can assume that class "subscribed = y" is more important to you and you want to minimize the classification error to this class. Justify your decision.

Ans.

Algorithm	Cost-effective Analysis	Correctly Classified Instance	Remark
Naïve Bayes	5289	88.30%	Less accuracy
PART	5289	89.81%	More accurate
J48	5289	89.84%	More accurate
OneR	5289	89.28%	More accurate

The 3 algorithms that were chosen were PART, J48, and OneR. based on Cost analysis these 3 algorithms performed better in accuracy. Although the difference is not major but these 3 have higher accuracy than NaiveBayes.

3) For each of the 3 algorithms chosen in question 2) of Section 2.1, determine the number of attributes to select in order to achieve a relatively better classification performance by using that algorithm. Justify your decision.
There are 11 attributes in this dataset. For saving your time, you may choose a number between 4 to 8. This question is to determine the **number**, not to choose the attributes.

Ans.

Algorithm	**Number of attributes**
PART	3,7,8,9,10
J48	3,7,8,9,10
OneR	3,7,8,9,10

2.2. Java program

For this part, you are required to develop a Java program to classify customers in the entire bank dataset. Your program should satisfy the following requirements:

1) Perform the classification task using the 4 algorithms chosen in question_1) of Section 2.1 by taking cost into consideration. Your program should display classification accuracy (i.e., correctly classified instances) and total cost for each classifier. Basically, your program should produce the same results as the results obtained in question_2) of Section 2.1 using Weka.

   Hint: You can use Weka class `CostSensitiveClassifier` (https://weka.sourceforge.io/doc.dev/weka/classifie2rs/meta/CostSensitiveClassifie r.html) to do the classification.

2) Perform the classification task using the 3 algorithms chosen in question_2) of Section 2.1 with the number of attributes determined in question-3) of Section 2.1 for each of the three algorithms. Your program should display correctly classified instances values and accuracy. The accuracy results should be the same as the results obtained in question_3) of Section 2.1 using Weka.

STATEMENT : ALL TASK HAS BEEN COMPLETED WITH JAVA CODE IN TASK 2 FOLDER

3. Task 3: Text classification in Weka and Java (12 marks)

Download dataset **`News.arff`**. This is a text dataset consisting of news documents. These news documents are categorised into four classes: computer, politics, science, and sports. In this task, you are required first to classify the news documents using Weka to determine some parameters in the filter, then develop a Java program to classify the news in this dataset.

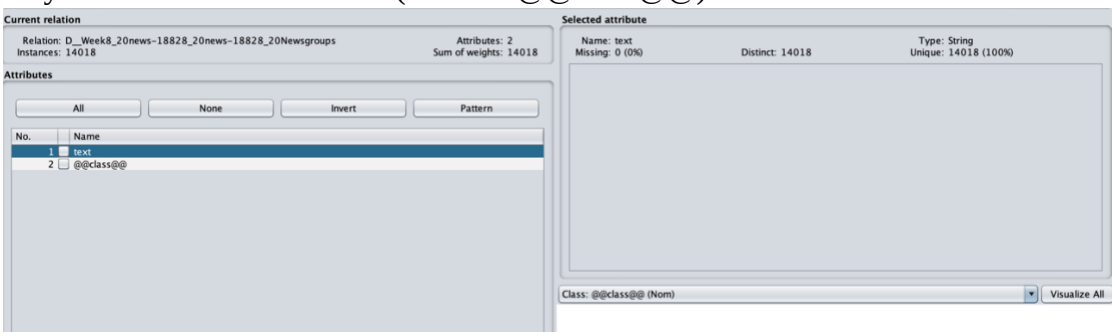
3.1. Attribute selection in Weka

In this part, you need to use a filter in Weka to extract attributes from the documents. You can choose 100 attributes and use J48 classifier to do the classification. For the parameters in the filter, you can use their default values or you can set up some values of your choice without tuning them (i.e., you can randomly choose some values). But you are required to tune 3 or 4 parameters that you think are important for determining attributes to represent the documents.

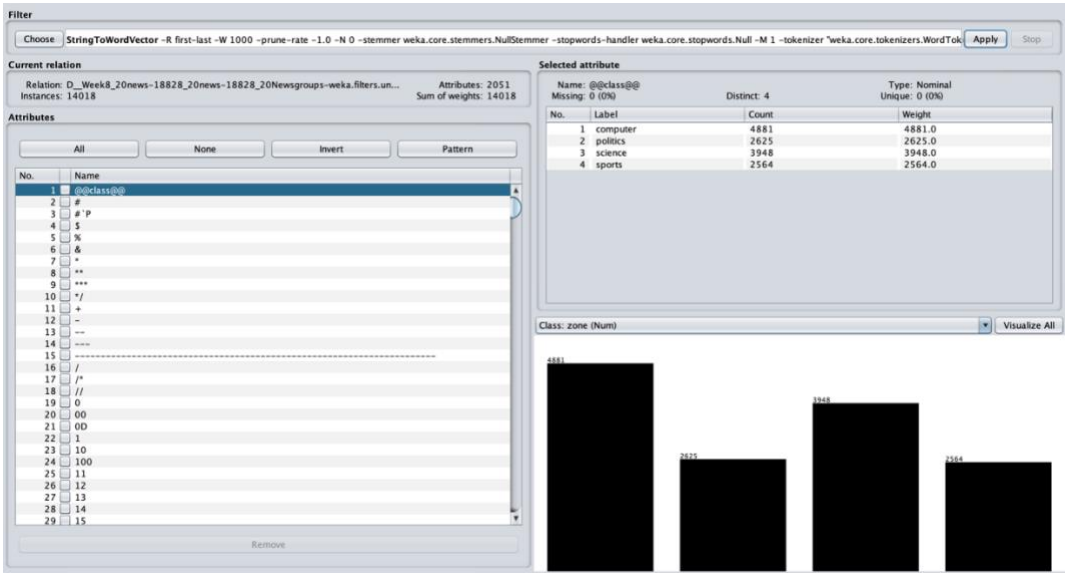
1) Briefly describe your working process in Weka to determine the values for the parameters in the filter. Provide evidence to show your working.

Ans.

1. First loading the dataset "News.arff" into the weka.. We observe that intialaly there are only 2 attributes in the dataset (text and @@class@@).

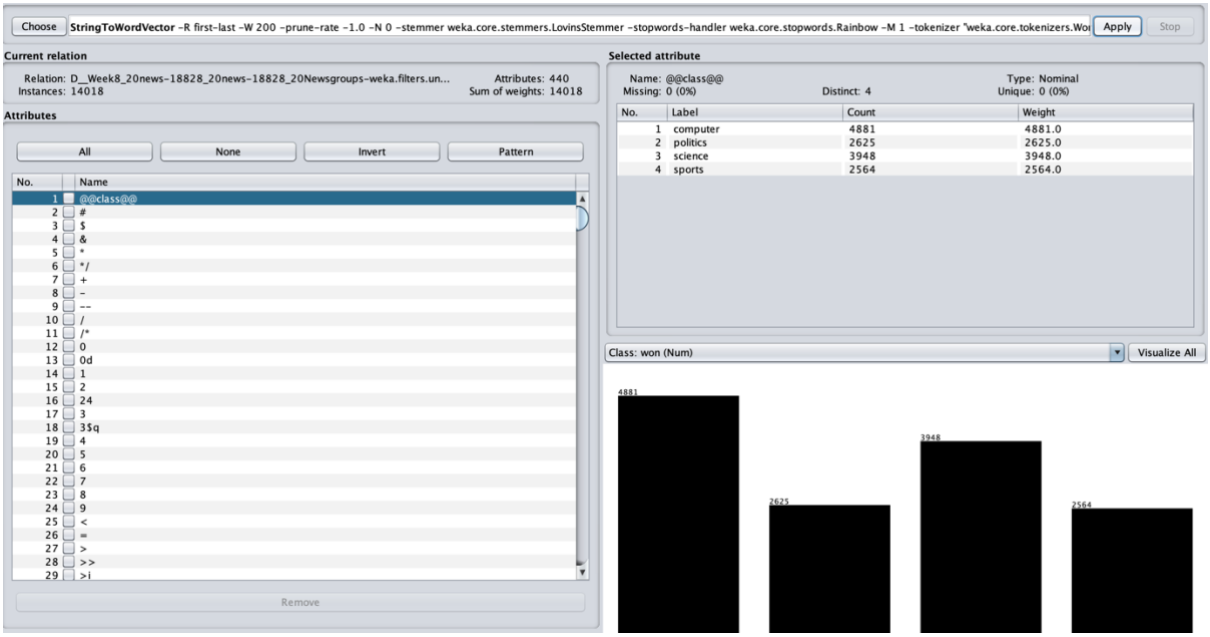


2. Applying StringToWordVector filter to the dataset. We observe the total attrributes changed to 2051 attributes.



3. We need to perform preprocessing by removing words, stemming etc. To optimize the dataset.

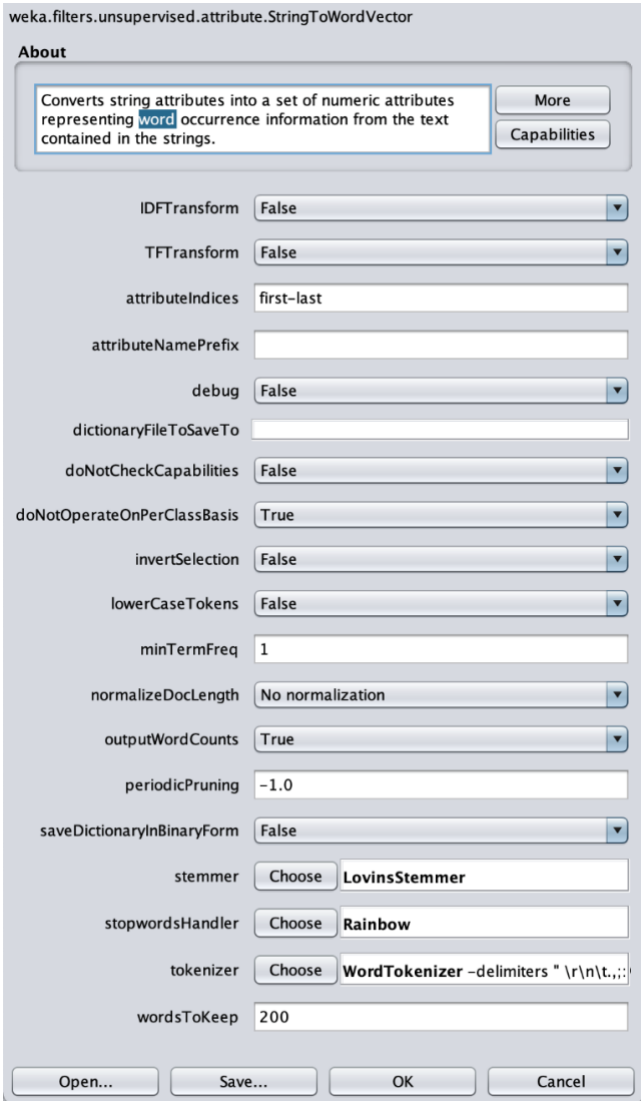
4. We change stemmer to LovinStemmer , stopwords handler to Rainbow and setwordsToKeep as 200. And click apply these parameters. We observer that the total attributes are now 440 which also shows overlapping of two classes.



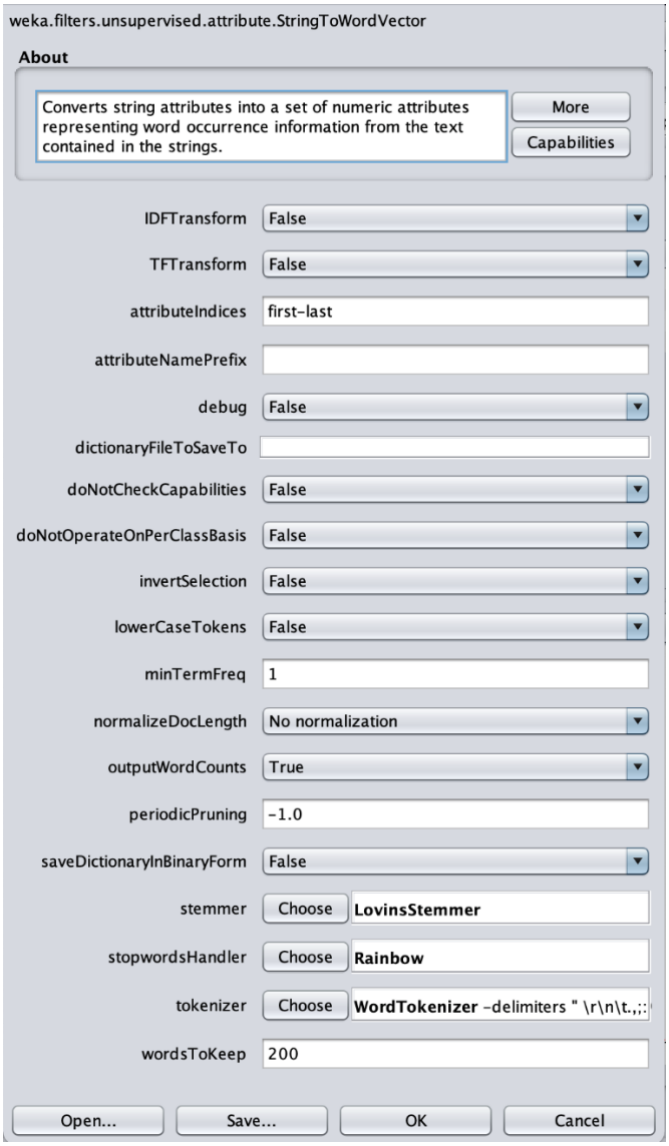
5. To overcome overlapping and to represent data in vector format we change outputWordCount to True and doNotOperateOnPerClassBasis to True. Observing the attributes in total of 201. Due to doNotOperateOnPerClassBasis is true. The max number of words is not enforced on per-class basis it is now based on all the classes.

6. Now the data is preprocessed. Performiong classification using j48 classfiers and compaing the results with different parameters for measuring the best performace.

7. Performing j48 classfication with these parameters: stemmer = LovinStemmer , stopword handler = Rainbow, setwordstoKeep = 200 ,outputWordCount = True, doNotOperateOnPerClassBasis = True. The Correctly Classified Instance are: 68.61%

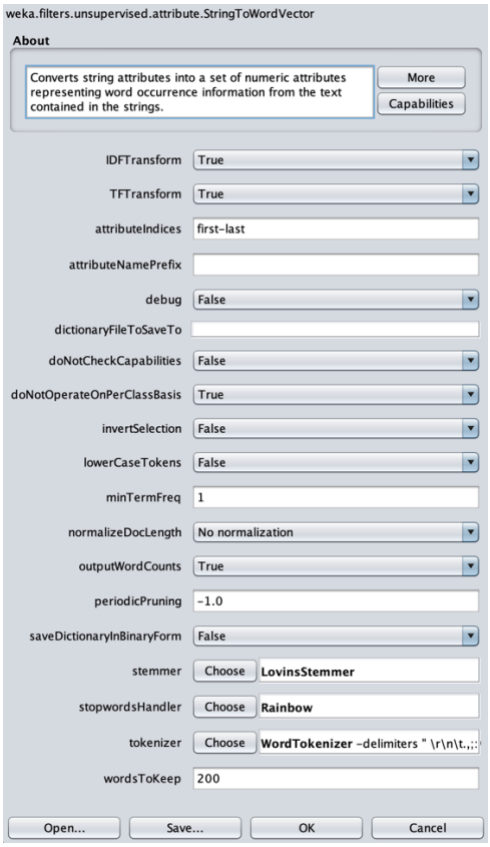


8. Performing j48 classfication with these parameters: stemmer = LovinStemmer , stopword handler = Rainbow, setwordstoKeep = 200 ,outputWordCount = True, doNotOperateOnPerClassBasis = False. The Correctly Classified Instance are: 68.61%



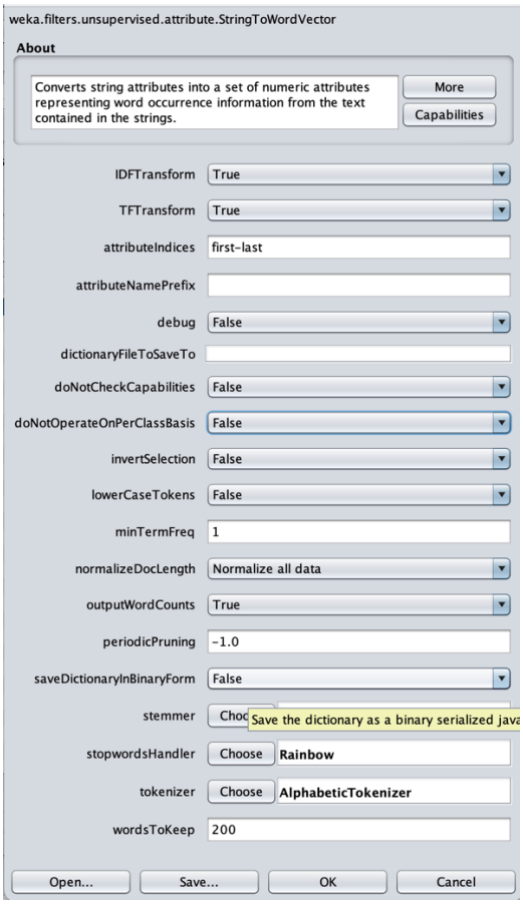
9. Performing j48 classfication with these parameters: stemmer = LovinStemmer , stopword handler = Rainbow, setwordstoKeep = 200 ,outputWordCount = True, doNotOperateOnPerClassBasis = True, IDFTransform = True. The Correctly Classified Instance are: 68.53%

10. Performing j48 classfication with these parameters: stemmer = LovinStemmer , stopword handler = Rainbow, setwordstoKeep = 200 ,outputWordCount = True, doNotOperateOnPerClassBasis = False, IDFTransform = True, TFTransform = True. The Correctly Classified Instance are: 68.50%



11. By looking at the correctly classified instances with different parameters the difference is minors.

12. Performing j48 classfication with these parameters: stemmer = LovinStemmer , stopword handler = Rainbow, setwordstoKeep = 200 ,outputWordCount = True, doNotOperateOnPerClassBasis = False, IDFTransform = True, TFTransform = True and normalizedDocLength = normalize all data. The Correctly Classified Instance are: 68.74%



13. Performing j48 classfication with these parameters: stemmer = LovinStemmer , stopword handler = Rainbow, setwordstoKeep = 200 ,outputWordCount = True, doNotOperateOnPerClassBasis = False, IDFTransform = True, TFTransform = True, normalizedDocLength = normalize all data. And tokenizer = alphabetic tokenizer. The Correctly Classified Instance are: 80.88%

14. So, after applying and trying different parameters settings we can see that the best parameters setting resulted in 80.88% of correctly classified instances

2) Which parameters in the filter that you want to tune? What are the chosen values for these parameters? Justify your decision with evidence.

The parameters in the filter which are required to tune are stemmer = LovinStemmer , normalizedDocLength = normalize all data, stopwords handler = Rainbow, setwordstoKeep = 200, tokenizer = Alphabetic tokenizer. Resulted in greater difference in results.

3.2. Java program

For this part, you are required to develop a Java program to classify the documents in the news dataset.

1) Perform the classification task using 4 classification algorithms, **`IBk`**, **`SMO`**, **`J48,`** and the method **`HoeffdingTree`** in Weka, and use the filter with the parameter settings determined in section 3.1.
   Weka Result:

Algorithm	**Correctly_Classified Instance**	**Time accuracy (s)**
IBk	74.34%	0.05
SMO	88.78%	38.51
J48	80.88%	131.76
HoeffdingTree	77.09%	4.15

2) Your program should display the correctly classified instances results, accuracy, and the time taken by each algorithm.

3) Which classifier performs the best in terms of time efficiency? Describe why this algorithm is faster than others.
   In weka, the SMO performed better than the rest of the algorithms and also in java program the SMO performed slightly better than other algorithms.

STATEMENT: ALL TASK HAS BEEN COMPLETED WITH JAVA CODE IN TASK 3 FOLDER