Man Tik Li (ml3546)
INFO 371 – Data Mining Applications
Assignment 4

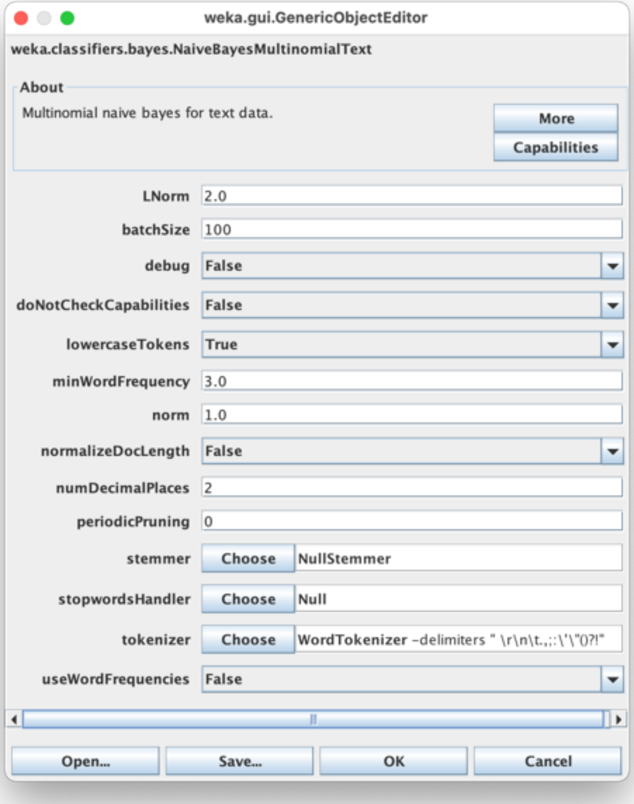
Task 1

The test datasets were selected randomly from the HW2 original dataset (abstract.arff). The test dataset has 2 instances from each domain (AI, STAT, ENVS).

AI: Line 6-7
STAT: Line 24-25
ENVS: Line 53-54

Task 2

Setting for the classifier in Weka



weka.gui.GenericObjectEditor

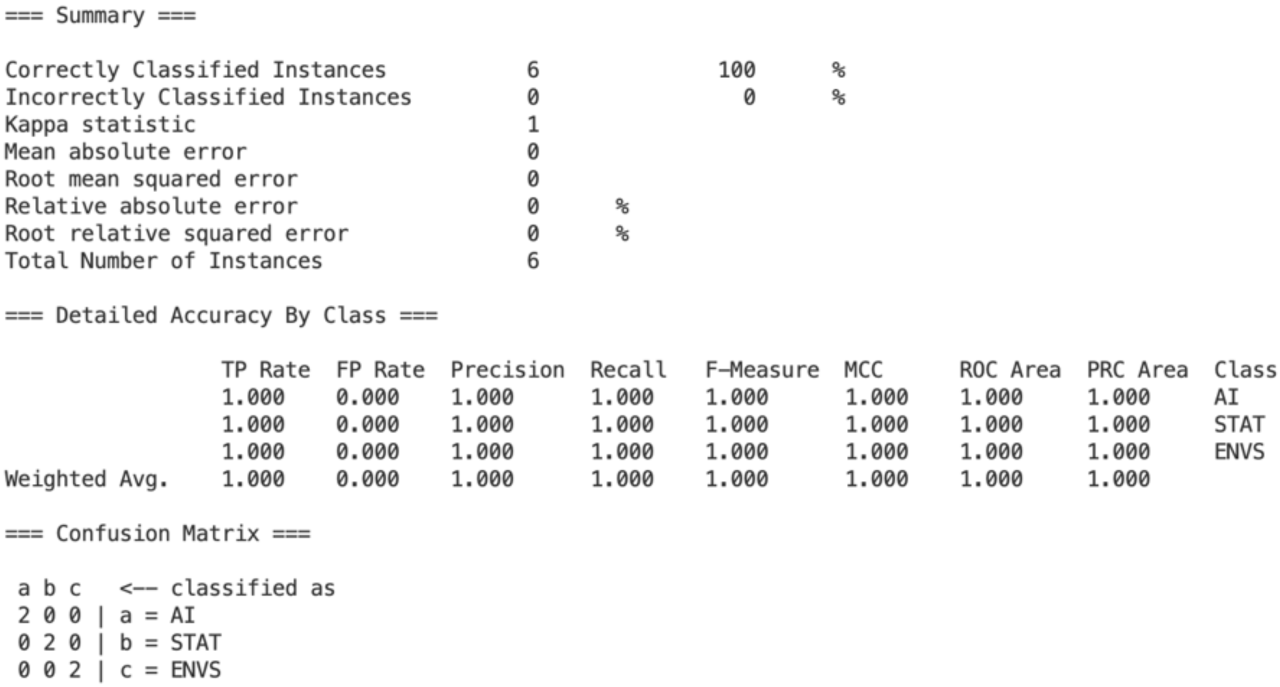
weka.classifiers.bayes.NaiveBayesMultinomialText

About

Multinomial naive bayes for text data.

LNorm	2.0
batchSize	100
debug	False
doNotCheckCapabilities	False
lowercaseTokens	True
minWordFrequency	3.0
norm	1.0
normalizeDocLength	False
numDecimalPlaces	2
periodicPruning	0
stemmer	NullStemmer
stopwordsHandler	Null
tokenizer	WordTokenizer –delimiters " \r\n\t.,;:\'\"()?!"
useWordFrequencies	False

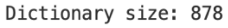
Classifier Output:



```
=== Summary ===

Correctly Classified Instances           6              100      %
Incorrectly Classified Instances         0                0      %
Kappa statistic                          1
Mean absolute error                      0
Root mean squared error                  0
Relative absolute error                  0      %
Root relative squared error              0      %
Total Number of Instances                6

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     AI
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     STAT
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     ENVS
Weighted Avg.    1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000

=== Confusion Matrix ===

 a b c   <-- classified as
 2 0 0 | a = AI
 0 2 0 | b = STAT
 0 0 2 | c = ENVS
```

Questions:

1. What is the dictionary size of the classifier model?



```
Dictionary size: 878
```

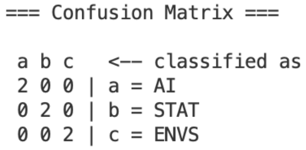
2. What is the total number of instances in the training set?
54

3. How many instances in each domain in the training set?
18

4. How many instances in the test set?
6

5. How many test instances were incorrectly classified?
0

6. What is the confusion matrix of the test result? What do the values mean in the confusion matrix?



```
=== Confusion Matrix ===

 a b c   <-- classified as
 2 0 0 | a = AI
 0 2 0 | b = STAT
 0 0 2 | c = ENVS
```

7. For each domain, list the recall, precision, and F1-Measure. Describe the underlying methods of computing the metrics.

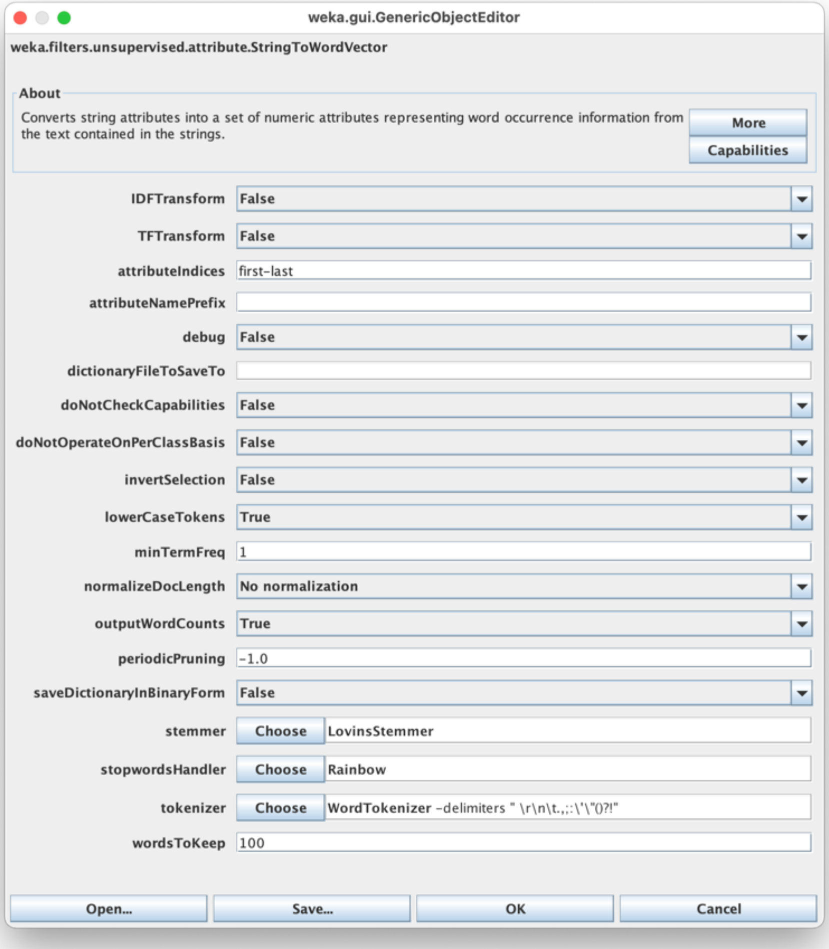
Domain	Recall	Precision	F-Measure
AI	1.000	1.000	1.000
STAT	1.000	1.000	1.000
ENVS	1.000	1.000	1.000

8. What is the average recall, precision, and F1-Measure of the Naïve Bayes Classifier on the datasets? Describe the underlying methods of computing the averages.

	Recall	Precision	F-Measure
Weighted Avg.	1.000	1.000	1.000

Task 3

Setting used for filter in Weka



weka.gui.GenericObjectEditor

weka.filters.unsupervised.attribute.StringToWordVector

About

Converts string attributes into a set of numeric attributes representing word occurrence information from the text contained in the strings.

IDFTransform	False
TFTransform	False
attributeIndices	first-last
attributeNamePrefix	
debug	False
dictionaryFileToSaveTo	
doNotCheckCapabilities	False
doNotOperateOnPerClassBasis	False
invertSelection	False
lowerCaseTokens	True
minTermFreq	1
normalizeDocLength	No normalization
outputWordCounts	True
periodicPruning	-1.0
saveDictionaryInBinaryForm	False
stemmer	LovinsStemmer
stopwordsHandler	Rainbow
tokenizer	WordTokenizer -delimiters " \r\n\t.,;:\'\"()?!"
wordsToKeep	100

Term Frequency of selected words:

	Frequency of the Selected 6 Dictionary Word						
Domain	ai	computer	stat	sampl	plant	water	**Grand Total**
AI	50	14	5	0	1	0	70
STAT	0	8	51	21	2	0	82
ENVS	0	0	2	3	8	29	42

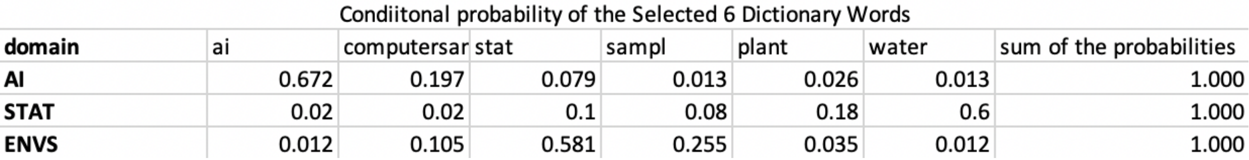
Pivot Table:

Row Labels	**Sum of ai**	**Sum of computer**	**Sum of stat**	**Sum of sampl**	**Sum of plant**	**Sum of water**	Grand Total
AI	50	14	5	0	1	0	70
ENVS	0	0	4	3	8	29	44
STAT	0	8	49	21	2	0	80
Grand Total	**50**	**22**	**58**	**24**	**11**	**29**	194

Conditional Probability Chart:

Conditional Probability of the Selected 6 Dictionary Word							
Domain	ai	computer	stat	sampl	plant	water	Sum of the probabilities
AI	$\frac{50+1}{70+6} = 0.672$	$\frac{14+1}{70+6} = 0.197$	$\frac{5+1}{70+6} = 0.079$	$\frac{0+1}{70+6} = 0.013$	$\frac{1+1}{70+6} = 0.026$	$\frac{0+1}{70+6} = 0.013$	1.000
STAT	$\frac{0+1}{44+6} = 0.02$	$\frac{0+1}{44+6} = 0.02$	$\frac{4+1}{44+6} = 0.1$	$\frac{3+1}{44+6} = 0.08$	$\frac{8+1}{44+6} = 0.18$	$\frac{29+1}{44+6} = 0.6$	1.000
ENVS	$\frac{0+1}{80+6} = 0.012$	$\frac{8+1}{80+6} = 0.105$	$\frac{49+1}{80+6} = 0.581$	$\frac{21+1}{80+6} = 0.255$	$\frac{2+1}{80+6} = 0.035$	$\frac{0+1}{80+6} = 0.012$	1.000

In Excel

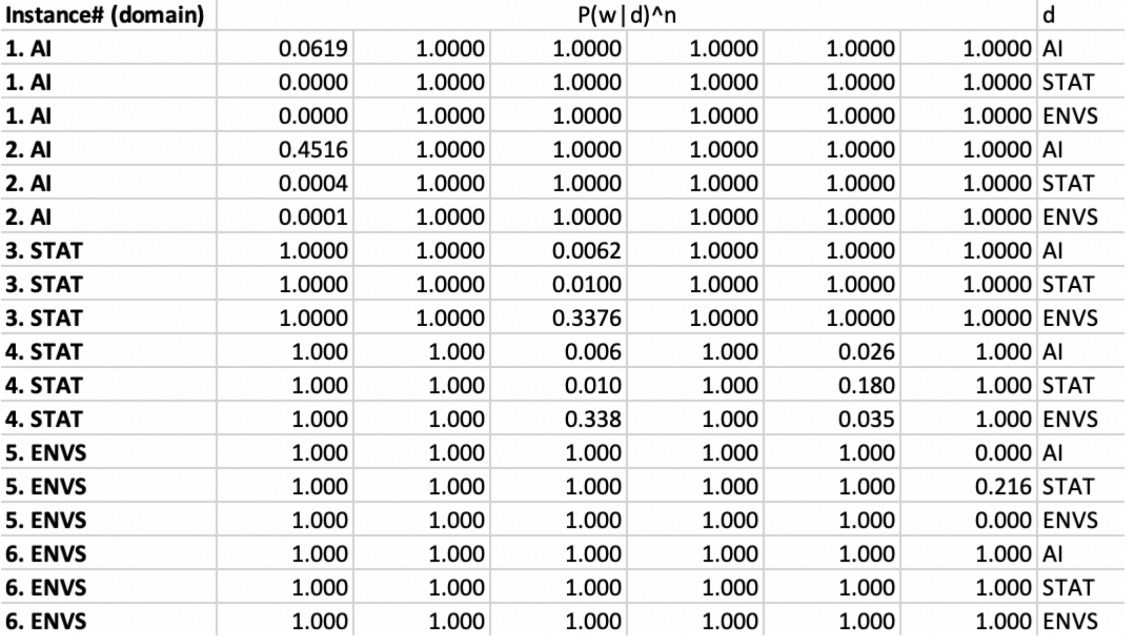


Condiitonal probability of the Selected 6 Dictionary Words

domain	ai	computersar	stat	sampl	plant	water	sum of the probabilities
AI	0.672	0.197	0.079	0.013	0.026	0.013	1.000
STAT	0.02	0.02	0.1	0.08	0.18	0.6	1.000
ENVS	0.012	0.105	0.581	0.255	0.035	0.012	1.000

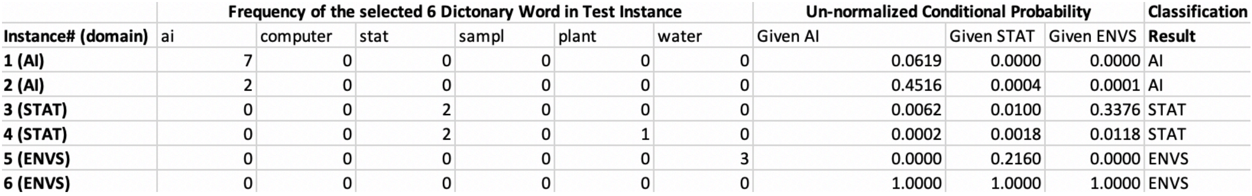
When doing the test data, I put all the data into excel sheet to make it filter easier. I identify the rows in the term frequency output and corresponded with the testing data, All the calculation completed in abstract-tf.csv. I created an auxiliary table to computed P(w|d)^n for each instance words. Then, I used the data to calculate the un-normalized conditional probability for each instance and given a different domain. After the calculation, I used the higher of the 3 values to predict which instance belongs to the domain.

Auxiliary table calculating P(w|d)^n in Excel



Instance# (domain)	P(w|d)^n						d
1. AI	0.0619	1.0000	1.0000	1.0000	1.0000	1.0000	AI
1. AI	0.0000	1.0000	1.0000	1.0000	1.0000	1.0000	STAT
1. AI	0.0000	1.0000	1.0000	1.0000	1.0000	1.0000	ENVS
2. AI	0.4516	1.0000	1.0000	1.0000	1.0000	1.0000	AI
2. AI	0.0004	1.0000	1.0000	1.0000	1.0000	1.0000	STAT
2. AI	0.0001	1.0000	1.0000	1.0000	1.0000	1.0000	ENVS
3. STAT	1.0000	1.0000	0.0062	1.0000	1.0000	1.0000	AI
3. STAT	1.0000	1.0000	0.0100	1.0000	1.0000	1.0000	STAT
3. STAT	1.0000	1.0000	0.3376	1.0000	1.0000	1.0000	ENVS
4. STAT	1.000	1.000	0.006	1.000	0.026	1.000	AI
4. STAT	1.000	1.000	0.010	1.000	0.180	1.000	STAT
4. STAT	1.000	1.000	0.338	1.000	0.035	1.000	ENVS
5. ENVS	1.000	1.000	1.000	1.000	1.000	0.000	AI
5. ENVS	1.000	1.000	1.000	1.000	1.000	0.216	STAT
5. ENVS	1.000	1.000	1.000	1.000	1.000	0.000	ENVS
6. ENVS	1.000	1.000	1.000	1.000	1.000	1.000	AI
6. ENVS	1.000	1.000	1.000	1.000	1.000	1.000	STAT
6. ENVS	1.000	1.000	1.000	1.000	1.000	1.000	ENVS

Result Table:



	Frequency of the selected 6 Dictonary Word in Test Instance						Un-normalized Conditional Probability			Classification
Instance# (domain)	ai	computer	stat	sampl	plant	water	Given AI	Given STAT	Given ENVS	Result
1 (AI)	7	0	0	0	0	0	0.0619	0.0000	0.0000	AI
2 (AI)	2	0	0	0	0	0	0.4516	0.0004	0.0001	AI
3 (STAT)	0	0	2	0	0	0	0.0062	0.0100	0.3376	STAT
4 (STAT)	0	0	2	0	1	0	0.0002	0.0018	0.0118	STAT
5 (ENVS)	0	0	0	0	0	3	0.0000	0.2160	0.0000	ENVS
6 (ENVS)	0	0	0	0	0	0	1.0000	1.0000	1.0000	ENVS

Create the Confusion Matrix:

a b c ← classified as
2 0 0 | a = AI
0 2 0 | b = STAT
0 0 2 | c = ENVS

Create a Performance Evaluation:

Domain	Precision	Recall	F1 Score
AI	1.000	1.000	1.000
STAT	1.000	1.000	1.000
ENVS	1.000	1.000	1.000

Overall

After everything is completed, I see that running the test manually can produced precise and accurate result. I had the same result as the Weka output.