Drexel University
College of Computing and Informatics

INFO 371 – Data Mining Applications

Assignment 4

Due Date: Sunday, Nov. 17, 2019

A. Requirements

TEAM (up to three members) assignment: Please work with another student or two on this assignment. Please inform the instructor of your team membership before working on the assignment.

Data Collection (10 point)

Use the small data set of paper abstracts (or short news articles/reports) you collected in the Assignment 2. In particular, the data set should contain text instances for **three different domains/areas** of your interest. Each domain has at least **20 papers** (one-paragraph or so abstracts only, with at least 150 words each). The data is described in an ARFF file named **abstracts.arff** with two attributes:

- String attribute: **@attribute abstract string**
- Nominal attribute: **@attribute domain {…}** (replace … with the areas you identified)
- **At least 60 instances** in the data.

If the dataset you used in the Assignment 2 didn't meet the above requirements, you need to complete the dataset in this Assignment. Otherwise, all the points for Data Collection will be deducted.

Task 1: Split the Data set into Training and Test sets (10 points)

--

1. Split the entire dataset into a training set with 90% of the instances and a test set with 10% of the instances as follows:
    a. Choose 10% of the instances in each domain (at least 2 instances/domain) and save the total (at least) 6 instances in a file called **abstracts-test.arff**. Remember the indices of the test instances in the original dataset. You will need the indices to retrieve the instances again in Task 3.
    b. Save the remaining 90% of the instances in a file called **abstracts-train.arff**.
    c. Both files should have the same attribute description as in **abstracts.arff**.

Task 2: Naïve Bayes Classification using Weka (10 points)

--

1. Open **abstracts-train.arff** in Weka Explorer. Click the "Edit" button to view the raw data. Make sure you see 90% of the instances in each domain.
2. Click the classify tab and choose **weka->classifiers->bayes->NaiveBayesMultinominalText**.
3. Click the text box containing the classifier name to open the property editor. On the property editor, check **lowercaseTokens** to be True. Click OK.
4. On Test Option, check **Supplied test set**. Click the **Set…** button. Open the **abstracts-test.arff** file. Make sure the Class is **(Nom) domain**.
5. Click start. Use the results in the Classifier output area to answer the following questions:
    a. What the is the dictionary size of the classifier model?
    b. What is the total number of instances in the training set?

c. How many instances in each domain in the training set?
d. How many instances in the test set?
e. How many test instances were incorrectly classified?
f. What is the confusion matrix of the test results?
g. For each domain, list the recall, precision, and F1-Measure.
h. What is the average recall, precision, and F1-Measure of the Naïve Bayes Classifier on the datasets?

Task 3: Naïve Bayes Classification by Manually Computing Conditional Probabilities (70 points)

In this task, you are asked to manually compute the conditional probabilities to classify the test instances.

1. Open **abstracts.arff (including both the training and the test instances)** in Weka Explorer and choose **unsupervised-> attribute-> StringToWordVector** for Filter.
2. Click the text box containing "StringToWordVector –R …" to bring up the property editor to change options below:
   a. Select **true** for **lowerCaseTokens;**
   b. Select **true** for **outputWordCounts** (this will produce Term Frequency/counts rather than binary 0/1 values);
   c. For **stemmer**, choose **LovinsStemmer**;
   d. For **stopwordsHandler**, choose **Rainbow**;
   e. For **wordsToKeep**, specify **100**;
   f. click **OK.**
2. Click **Apply** to run the vectorization filter and you will see a number of new word attributes after it is done.
3. Click **Save…** to save the results (tf vector representation) as a **CSV** file called **abstracts-tf.csv.**
4. Click **Edit…** to view the processed data and manually select 6 important words to **create a data table as follows**:
   a. Pick 6 words. Consider reusing the 3 rare words (unique in each area/domain) and three other words/attributes in Assignment 2. You will use these 6 words as a small dictionary for Naïve Bayes Classification on the test instances.
   b. Create a data table (**example below**) Containing the frequency of each of the above 6 words in each domain.

	Frequency of the Selected 6 Dictionary Word						
domain	science	nature	machine	info	data	math	**Grand Total**
Stats	1	1	1	2	1	3	9
ML	2	0	3	2	3	2	12
DM	4	2	2	5	2	2	17

   c. In the above table, an inner cell contains the frequency of a dictionary word in a specific domain. For example, the cell in the intersection of DM-science has a value 4. That means the word 'science' appears 4 times in all of the instances in the domain 'DM'. To fill up the above table, open the saved CSV file **abstracts-tf.csv** in MS Excel. Create appropriate pivot table to sum up the frequencies of the chosen 6 dictionary words.
   d. The rightmost column of the above table contains the grand total of word frequency in each domain. These total numbers will be used to compute the conditional probabilities for each word for given a domain.
5. Create a table containing the conditional probabilities of the 6 dictionary words. To estimate the conditional probabilities by taking zero-frequency into consideration, you need to use the following Laplace-estimator formula:

$$P(w|domain) = \frac{frequency\ of\ w\ in\ the\ domain + 1}{grand\ total\ frequency\ of\ all\ words\ in\ the\ domain + size\ of\ the\ dictionary}$$

For example, we can compute the following conditional probabilities for the above selected 6 dictionary words using the frequencies in the above table:

	Conditional Probability of the Selected 6 Dictionary Word						
domain	science	nature	machine	info	data	math	**Sum of the probabilities**
Stats	$\frac{1+1}{9+6} =$ 0.133	$\frac{1+1}{9+6} =$ 0.133	$\frac{1+1}{9+6} =$ 0.133	$\frac{2+1}{9+6} =$ 0.2	$\frac{1+1}{9+6} =$ 0.133	$\frac{3+1}{9+6} =$ 0.267	1
ML	$\frac{2+1}{12+6} =$ 0.167	$\frac{0+1}{12+6} =$ 0.056	$\frac{3+1}{12+6} =$ 0.222	$\frac{2+1}{12+6} =$ 0.167	$\frac{3+1}{12+6} =$ 0.222	$\frac{2+1}{12+6} =$ 0.167	1
DM	$\frac{4+1}{17+6} =$ 0.217	$\frac{2+1}{17+6} =$ 0.13	$\frac{2+1}{17+6} =$ 0.13	$\frac{5+1}{17+6} =$ 0.261	$\frac{2+1}{17+6} =$ 0.13	$\frac{2+1}{17+6} =$ 0.13	1

6. For each of the test instance, obtain the frequencies of the selected 6 dictionary words from the raw vectorization data in weka by clicking the Edit button, or from the CSV file opened in Excel.

   Given the word frequencies in the test instances and the conditional probabilities of each dictionary word in the above table, use the following formula to classify each instance:

   $$d = argmax_{d \in domains} P(d) \times (P(w_1|d))^{n_1} \times (P(w_2|d))^{n_2} \times \ldots \times (P(w_k|d))^{n_k}$$

   Where $P(d)$ is the probability of the domain d, $P(w_i|d)$ is the conditional probability of the word w_i given the domain d , and n_i is the frequency of word w_i in the instance. Since you should have equal number of instances in your dataset, you can ignore the probability $P(d)$ in the calculation.

   For example, the following table contains the frequencies of each dictionary word in each test instance. The columns under the title "Un-normalized Conditional Probability" contains the results of applying the above formula for each domain. Finally, the last column is the classification result.

	Frequency of the selected 6 Dictionary Word in Test Instance						**Un-normalized Conditional Probability**			**Classification**
Instance# (domain)	science	nature	machine	info	data	math	Given Stats	Given ML	Given DM	**Result**
1 (Stats)	1	0	0	1	0	1	.0071	.0047	.0074	*DM*
2 (Stats)	3	1	3	1	6	3	6.59e-11	2.67e-13	8.08e-15	*Stats*
3 (ML)	0	1	1	0	0	0	0.018	0.012	0.017	*Stats*
4 (ML)	2	0	3	2	3	2	2.79e-10	3e-9	2.62e-10	*ML*
5 (DM)	1	1	0	0	1	1	0.0006	0.0003	0.0005	*Stats*
6 (DM)	3	0	2	2	0	2	2.22e-7	1.79e-7	1.99e-7	*Stats*

7. Create the same table as above and filled up the table using your own data.
8. Create the confusion matrix for the test instances and evaluate the performance in terms of

precision, recall, and F-Measure.

B. What to Hand In

1. The **data files**, including 1) abstracts.arff, 2) abstracts-train.arff, 3) abstracts-test.arff, and 4) abstracts-tf.csv.
2. A well-structured report documenting the tasks, results, discussion, and answers to the questions in MS Word or PDF format. Your report must contain the following content:
   a. For each task, describe the requirements and show the results including necessary screenshots. For each table and figure, you must add sufficient commentary to explain what the table or the figure is about.
   b. Answers to the questions asked in the tasks.
   c. Necessary intermedia steps or raw data to allow any readers to reproduce your results.
   d. Performance evaluation including confusion matrix, accuracy, precision, recall, and F-Measure.

C. How to Hand In

1. Please name your report file as **INFO371-assign4-yourFirstName-yourLastName.docx**.
2. Submit your report file through the course website in the **Blackboard Learn** system.

D. When to Hand In

1. Submit your assignment no later than **11:59pm** in the due date.
2. There will be a 10% (absolute value) deduction for each day of lateness, to a maximum of 3 days; assignments will not be accepted beyond that point. Missing work will earn a zero grade.

E. Written Presentation Requirements

Images must be clear and legible. Assignments will be judged on the basis of visual appearance, grammatical correctness, and quality of writing, as well as their contents. Please make sure that the text of your assignments is well-structured, using paragraphs, full sentences, and other features of well-written presentation. Text font size should be at least 11 point.