IS 577: Data Mining
HW#2

Instructor: Dr. Jingrui He

Out date: Mar 4, 2021; Due date: Apr 8, 2021

Submit electronically, for Assignment #2, a file named `yourFirstName-yourLastName-HW2.zip` *containing your solutions to this assignment. Written questions should be in .pdf format. We suggest that you follow the instructions in* `https://wiki.illinois.edu/wiki/display/ischool/How+to+Compress+a+PDF` *to compress your pdf file. The code implementations should be bug-free and well-commented.*

1 [10 points] Basic Concepts

Please keep your answers to less than 50 words.

- What does it mean for a measure to be null-invariant?
- What does conditional independence mean in the context of Naive Bayes?
- Why should the test set be independent of the training set?
- Is accuracy a good metric for evaluating results on an imbalanced dataset? Why or why not?
- Suppose we have a dataset with 1 million examples and a classifier that takes approximately 1 second/example to train. What evaluation method (i.e., holdout, cross-validation, or bootstrap) would you recommend to use and why?

2 [10 points] Advanced Frequent Pattern Mining

Given a transactional database TDB as shown in Table 2, find **all** the pairs of items whose Kulczynski measure is below a threshold of $\frac{1}{4}$. **To get full credits, you will have to show the Kulczynski measure values of these pairs**.

Transaction id	Items
1	beer, coffee, diaper
2	coffee, milk, nuts
3	diaper, milk
4	beer, diaper, nuts
5	milk, nuts
6	beer, diaper, milk, nuts
7	beer, nuts
8	beer, coffee, diaper

Table 1: Transactions

3 [20 points] Classification

Use Table 2 to answer the questions in this section. We would like to predict whether a team plays ultimate frisbee based on the weather. **You are required to show all calculations.**

Outlook	Temp	Windy	Play Frisbee
Rainy	Hot	Yes	No
Sunny	Hot	No	Yes
Sunny	Cool	Yes	Yes
Rainy	Cool	Yes	No
Overcast	Cool	No	Yes
Sunny	Hot	Yes	No
Rainy	Cool	No	Yes
Overcast	Cool	No	No
Overcast	Cool	Yes	Yes
Sunny	Hot	No	Yes

Table 2: Data for 10 Ultimate Frisbee Outings

- **[8 points]** Create a decision tree using information gain for attribute selection.
- **[2 points]** Using this decision tree, predict whether the team will play frisbee or not in the following days.
    - It is sunny, hot and not windy.
    - It is rainy, cool, and windy.
- **[2 points]** What are the prior probabilities for each outcome (play frisbee or not) in Table 2?
- **[8 points]** Using Naive Bayes, predict whether the team will play frisbee or not in the following days.
    - It is sunny, hot and not windy.
    - It is rainy, cool, and windy.

4 [60 points] Coding

You can use any programming language for this coding exercise, but you are suggested to use Python or R. For Python, you can use Jupyter. **You are required to submit your executable coding file (e.g., .py, .ipynb, .r or .m)**. Please contact the TA if you need help.

Download the three files provided with this homework (*smsspam_train.csv*, *smsspam_test.csv*, and *dt_predictions.txt*), and save these files in the same directory as your source file.

You will use the data from the SMSSpamCollection dataset, which we have been processed and separated into training (*smsspam_train.csv*) and test (*smsspam_test.csv*) sets. Each line in these files represents an SMS message. The first 500 columns indicate whether a particular word appears in the SMS message (1 if present, 0 otherwise). These columns will be our features. The last column (i.e., *_label_*) contains our class label (1 for spam, 0 for normal message).

- **[15 points]** Train a Naive Bayes classifier on the training set and predict whether each message in the test set is spam or not. Store your predictions in a variable named *nb_predictions*, and output the predictions in a file named *nb_predictions.txt*, with one line per prediction, i.e., the 1st line corresponds to the prediction of the 1st SMS in the test set, 2nd line to the 2nd SMS, etc.

  Hint: If you are using Python, you can use the *BernoulliNB* class from the *scikit-learn* package. For R, you can use the *bernoulli_naive_bayes* from the *naivebayes* library.

- **[15 points]** Train a logistic regression classifier on the training set and predict whether each message in the test set is spam or not. Store your predictions in a variable named *lg_predictions*, and output the predictions in a file named *lg_predictions.txt*, with one line per prediction, i.e., the 1st line corresponds to the prediction of the 1st SMS in the test set, 2nd line to the 2nd SMS, etc.

  Hint: If you are using Python, You can use the *LogisticRegresssion* class in the *scikit-learn* package. For R, you can use the *glm* function.

- **[10 points]** Write a function *compute_score* that takes in the true labels and predicted labels and returns the F1-score. Compute the F1-scores of the Naive Bayes and logistic regression predictions. Which classifier performs better on the test set?

- **[10 points]** We trained a decision tree classifier and stored predictions in the *dt_predictions.txt* file. Read and store the predictions in a variable named *dt_predictions*. Compute the F1-score. How does our decision tree classifier compare with the two previous classifiers on the test set?

- **[10 points]** One way to improve the classification results is to combine the predictions of multiple classifiers and perform majority voting. This is an example of a technique called ensemble learning. Assign the final prediction of *spam* (1) if at least two classifiers predict *spam*. Otherwise, assign the final prediction of *normal message* (0). Compute the F1-score of your predictions. Is the ensemble method better than using a single classifier on the test set?