CSE 5243: Homework #2

Deadline: **11:59PM on 02/19/2020**.
No late submissions will be accepted.

Instructions.

We currently use a 100-point scale for this homework, but it will take 10% of your final grade.

What you should turn in:

1. For Problem 1-3, please prepare your answers in a single PDF file named as HW2-p1-3.pdf.
2. For Problem 4, please put your code (a `.ipynb` file for Jupyter Notebook) and the report in a folder named as HW2-p4.
3. Put all your files (Problem 1 to 4) in a zip file named as HW2.zip, and submit it on Carmen.

Have Questions?

Please create a post on Carmen Discussion to get timely help from other students, the TA, and me. Everyone can benefit from first checking what have been asked previously. Please try to avoid directly sending me/TA emails.

Problem 1 (10 points)

For the following group of data:

100, 200, 400, 500, 700, 1000, 3000

a) Calculate its mean and standard deviation.

b) Normalize the above group of data by min-max normalization with min = 0 and max = 1.

c) In z-score normalization, what value should the first number 100 be transformed to? What about the last number 3000?

Problem 2 (10 points)

Given the following table,

X_1	X_2
-3	a
3	b
-4.4	a
6.0	a
-4.0	a
-12.0	b
1.2	a
16.0	b
-16.0	b
13.2	a

assuming that X_1 is discretized into three bins as follows:

$$c_1 = (-20, -5]; \; c_2 = (-5, 5]; \; c_3 = (5, 20]$$

Answer the following questions:

a) Construct the contingency table between the discretized X_1 and X_2 attributes. Include the marginal counts.

b) Compute the χ^2 statistic between them.

Problem 3 (50 points)

Assume we get some data from a car insurance company in Table 1, where there are 6 data instances representing 6 people, with 2 attributes (Age and Car) and 1 class label (Risk). Here Age is a continuous attribute. Now we will build decision trees for this data set.

Data Point	Age	Car	Risk
x_2	40	Vintage	H
x_6	25	SUV	L
x_4	45	SUV	L
x_3	20	Sports	H
x_5	40	Sports	L
x_1	45	Sports	L

Table 1: Data for Problem 3. *Age* is numeric and *Car* is categorical. *Risk* gives the class label for each point: high (H) or low (L).

a) Let us consider a multi-way split for the Car attribute (using its unique values for partition). What is the information gain if we choose the Car attribute to split the root node? (5 points)

b) Let us consider the binary splits for the Car attribute. Using information gain as the measure, which binary split of the Car attribute is the best at the root node? (5 points)

c) Between (**a**) and (**b**), which one do you prefer for splitting the root node using the Car attribute? Hint: Consider the GainRatio measure. (5 points)

d) Now, construct an entire decision tree for the given data set, using information gain as the split point evaluation measure. You can use your calculations or conclusions in (**a-c**). (30 points)

e) Classify the point (Age=27, Car=SUV) based on the constructed decision tree in **(d)**. (5 points)

Problem 5 (30 points). Programming Task on Real Text Data Preprocessing.

This assignment is the first part of a longer-term project. The objective of this assignment is to give you the experience of preprocessing real data. In subsequent assignments you will continue to use your preprocessed data for data mining tasks such as classification and clustering.

- **Programming Requirement**: ***You are required to use Python and Jupyter Notebook. If you really have to use other programming languages, please talk to the TA and make sure she can run your code successfully***.
- **Data**: 20 Newsgroups dataset, which contains text documents regarding different topics. It can be downloaded here `https://ysu1989.github.io/courses/sp20/cse5243/20news-train.zip`. Unzip the file, and there will be 20 subdirectories, each regarding a certain topic (indicated by the subdirectory name) and containing a set of text documents about that topic (each file is a document).
- **Data Format**: Each subdirectory contains posts to a newsgroup regarding a certain topic (i.e., the name of the subdirectory). Each post is a text document. Note that although they don't have the `.txt` suffix, they can be viewed using text editors or read in your code as ordinary text files. Each document has two main sections: meta-data and main body. Meta-data are fields like "From" (who posted it), "Organization", and "Subject" (it could be very indicative of the topic of the post). The meda-data usually has "Lines" as the last line, but not always. There is a new line between the main body and the meta-data.
- **Task**: In this assignment, your task is to construct a feature vector and class label for each document in the dataset. For now, please use the frequency of words in the document to construct a feature vector, and convert

the subdirectory name into an integer as class label (e.g., `alt.atheism` becomes 1, `comp.graphics` becomes 2, and so on). For example, if there are totally M documents and N distinct words in the entire dataset, you will construct a $M \times N$ matrix D and a $M \times 1$ vector Y, where $D_{i,j}$ is the number of occurrences of word j in document i, and Y_i is the class label of document D_i.

- **Libraries**: You first need to tokenize every document to get the collection of words in it. You can use off-the-shelf tokenizers from libraries like NLTK[1]. After that, it is up to you whether to do more advanced preprocessing like stemming[2] (e.g., both "cats" and "catty" are stemmed to "cat") or removing unnecessary characters from real words[3]. **You should NOT use off-the-shelf libraries beyond basic preprocessing like tokenization in this assignment.** For example, *scikit-learn* provides a "vectorizer" function that directly converts text documents into feature vectors, which should not be used in this assignment (you are encouraged to use it for purposes other than our assignments, of course).

- **Competition**: We will be hosting a competition similar to Kaggle[4] in **subsequent assignments on classification and clustering**. What is released to you is only the public training set, and there will be a hidden test set. Students whose model achieves the **top K scores** on the hidden test set will get **bonus points**. Details about K and bonus points will be revealed later. Note that **as long as you successfully achieve the requirement of an assignment, you will get the full points**. The competition is only for bonus points. There is no competition for this assignment, but good preprocessing is key to good performance in subsequent assignments.

What You Should Turn in:

You are expected to turn in the following files:

- Source code in a `.ipynb` file for Jupyter notebook. You do not have to include the raw dataset. Your Jupyter notebook should be made such that one can successfully run all the cells to replicate your results as long as the data directory is in the same directory as the notebook. In the Jupyter notebook, you should explain, in Markdown cells[5], what the following code cell is doing and how to interpret the output, plus any comment you'd like to add that may help others like the TA understand. **If you use off-the-shelf libraries like NLTK, please document which version of**

[1] `https://www.nltk.org/api/nltk.tokenize.html`

[2] `https://en.wikipedia.org/wiki/Stemming`

[3] Due to the format of posts, you may find words like ">Last" where ">" indicates the start of a line. Not removing them may compromise the quality of your features.

[4] `https://www.kaggle.com/competitions`

[5] `https://jupyter-notebook.readthedocs.io/en/stable/examples/Notebook/Working%20With%20Markdown%20Cells.html`

what library is used and how you acquired and installed it so that the TA can properly run the code. The lastest version of NLTK is recommended.

- A short report (no more than 2 pages in 12 pt. font), describing the approach/procedure you took to construct a feature vector, and where applicable the rationale for doing it (name this file **report1.pdf** or something like that). **The report is optional if you think you have sufficiently documented what you have done in the Jupyter notebook**.

Detailing what you did is very important even if it did not work, in which case it's even more important since the TA can then give partial points accordingly. Describe any difficulties you may have encountered and any assumptions you are making. Important: **You need to clearly state what you filtered and what you did not filter from your preprocessing**.

How to submit your files to Carmen:

After you finish all the problems, you just need to upload a single .zip file to Carmen, as we mentioned in Instructions.

The easiest way to upload files specific to the stdlinux environment is by opening carmen from the stdlinux and uploading the corresponding files from there. To open carmen from stdlinux,

- Login to stdlinux using CSE remote access. Check here for *remote access*, if needed: `https://cse.osu.edu/computing-services/resources/remote-access`.
- Open terminal
- Type "firefox". Click enter
- Navigate to carmen.osu.edu

If it is necessary to transfer files from Windows/Mac environment to stdlinux, one can make use of SCP and SFTP protocols to achieve so. This site[6] can provide additional details. Contact CSE Help Desk if further help is needed setting up File transfer clients.

Necessary Details:
a. Host Name: stdlinux.cse.ohio-state.edu
b. Port Number: 22
c. User Name: osu user-id
d. Password: OSU password

[6] `https://u.osu.edu/floss/documentation/using-scp-sftp-for-ohio-state-cse-students-to-sync-projects-and-remotely-access-your-files/`