CSC520 Fall 2019 Assignment 5

Due December 5^{th} 11:59pm

This assignment includes both conceptual and code questions. It must be completed individually. You may not collaborate with other students, share code, or exchange partial answers. Questions involving answers or code must be emailed to the instructor or TAs directly or discussed during office hours. Your answers to the conceptual questions must be uploaded to Moodle as a pdf file titled `Assign5_UnityID.pdf`. Your source code for question 2 must be submitted as a self-contained zip called `Assign5_q2_UnityID.zip` and your sourcecode for question 3 must be submitted as a *separate* self-contained zip called `Assign5_q3_UnityID.zip`. All code must be clear, readable, well-commented and entirely your own. The code will be tested on the NCSU VCL system("CSC520_VCL"). You are advised to test your code there before submission.

Question 1 (50 pts)

Here's some domain knowledge you can use to construct an intelligent agent for simple symptom classification.
Lexicon:

Nausea - The patient has Nausea.
Coughing - The patient is Coughing.
Vomiting - The patient is Vomiting.
Pneumonia - The patient has Pneumonia.
Pulmonary Oedema - The patient has Pulmonary Oedema.
Pulmonary Fibrosis - The patient has Pulmonary Fibrosis.
Total Organ Failure - The patient has Total Organ Failure.

Rules:

If it evolves towards Vomiting then it's Nausea.
If it evolves towards Pulmonary Oedema then it's Vomiting.
If it evolves towards Pneumonia then it's Coughing.
If it evolves towards Pulmonary Oedema and Pulmonary Fibrosis then it's Pneumonia.
If it evolves towards Pulmonary Fibrosis then it's Pulmonary Oedema.
If it evolves towards Total Organ Failure then it's Pulmonary Fibrosis.

a. Using the knowledge base below show the corresponding Bayes Net. (Caution: the complexity of the rest of the assignment depends on the orientation of the edges in the net, so think carefully).

b. Using the ideas of conditional independence discussed in Ch. 14 of R&N, identify which variables in this network are conditionally independent of the others and under what conditions. Where appropriate, it is sufficient to make shorthand statements such as X is conditionally independent of the rest of the network given Y, to save enumerating all the nodes in the network.

c. Use the prior and conditional probabilities provided to construct probability tables for each node of this network. Then use exact inference to answer the following questions. What is the probability that:

1. The symptoms include Nausea and Vomiting.
2. The symptom is Coughing given that it is Total Organ Failure.
3. The symptom is Pulmonary Oedema.

Prior and conditional probabilities known beforehand:

Nausea	P(Nausea)	Coughing	P(Coughing)
T	0.07	T	0.05

nausea	vomiting	P(vomiting $\mid nausea$)
T	T	0.8
F	T	0.05

coughing	pneumonia	P(pneumonia $\mid coughing$)
T	T	0.65
F	T	0.05

vomiting	pneumonia	pulmonary oedema	P(pulmonary oedema $\mid vomiting, pneumonia$)
T	T	T	0.95
T	F	T	0.9
F	T	T	0.75
F	F	T	0.1

pulmonary oedema	pneumonia	pulmonary fibrosis	P(pulmonary fibrosis $\mid pulmonaryoedema, pneumonia$)
T	T	T	0.8
T	F	T	0.75
F	T	T	0.6
F	F	T	0.05

pulmonary fibrosis	total organ failure	P(total organ failure $\mid pulmonaryfibrosis$)
T	T	0.7
F	T	0.02

Question 2 (50 pts)

For this question you have been given two sets of datafiles representing binary decision problems. The first dataset is heart disease classification based upon single proton emission computed tomorgraphy (SPECT). The second represents classification of mushrooms as edible or poisonous based upon observable features. Descriptions of the datasets are available on the UCI Machine Learning Repository: `http://archive.ics.uci.edu/ml/datasets/SPECT+Heart` `http://archive.ics.uci.edu/ml/datasets/Mushroom`. Each row represents a single problem instance with a binary outcome variable "class." Each dataset has been split into two in the datasets defines an instance of the problem in terms of a set of binary or categorical features. The datasets have been split into static TEST and TRAIN files.

Your task in this assignment is to Implement a Naïve Bayes classifier and to evaluate it on the available data. Your code should take as input a *training* and *testing* file. It will use the former to construct a Naïve Bayes classifier that predicts the class variable from the features with a cutoff of ≥ 0.5. It will then evaluate this model using the training dataset and calculate a confusion matrix. Your code should produce two output files, a *model file* that lists the structure of your model along with a probability distribution for each node, and a *result file* which lists the prediction and the real value for each row in the test set along with a final confusion matrix.

Your code file should be named NaiveBayes.java or NaiveBayes.py and should be called as follows:

```
java -jar NaiveBayes.jar TRAIN TEST MODEL-FILE RESULT-FILE
NaiveBayes.py TRAIN TEST MODEL-FILE RESULT-FILE
```

Where TRAIN is a training dataset; TEST is a test dataset; MFILE is an output file where you will save the model nodes and distributions; and RFILE is the result file. As with other assignments the code must be your own but you may use third party libraries to process the CSV files.

Question 3 (50 pts)

For this question you are tasked with implementing and evaluating a simple probabilistic language model. You should begin by downloading a set of books by a single author from public domain host Project Gutenberg (`https://www.gutenberg.org/`). NOTE: you must clean your downloaded text files by removing the project header and the license text (located at the end) at the files. You will then implement code to generate a probabilistic markov chain for the author.

When called your code will take a directory containing your downloaded texts. It will process the supplied texts to collect a set of the unique unigrams, bigrams, and trigrams present in the texts after removing stopwords, and then use this set to calculate an independent probability of each unigram as well as conditional probabilities given preceeding unigrams and bigrams. The resulting distributions should be saved to a probability file. Your code will then generate a random sample of 10 sequences of at most 20 tokens. These sequences and their probabilities will then be written to a result file. Include these sentences in your report along with a qualitative analysis of them.

Your code file should be named MarkovChain.java or MarkovChain.py and should be called as follows:

```
java -jar MarkovChain.jar AUTH-DIR/ PROB-FILE RESULT-FILE
MarkovChain.py AUTH-DIR/ PROB-FILE RESULT-FILE
```

where AUTH-DIR is the directory containing texts from the author; PROB-FILE is the output file for the probability distributions; and RESULT-FILE is where the sentences are saved. As always all code must be your own but for the purposes of this assignment you may make use of third party libraries to load and tokenize the text files. You have also been provided with a standard list of stopwords to be removed.

Extra Credit: For 15 points of extra credit you may adapt your code to take two directories belonging to different authors and silmultaneously generate models and sentences for each author and then evaluate the sentences *using models from both authors*. In this second case your code should be called as shown below and your result file should contain two sets of sentences, one for each author, and each sentence should show two probability scores, one for each model.

```
java -jar MarkovChain.jar AUTH-DIR-1/ AUTH-DIR-2/ PROB-FILE-1 PROB-FILE-2 RESULT-FILE
MarkovChain.py AUTH-DIR-1/ AUTH-DIR-2/ PROB-FILE-1 PROB-FILE-2 RESULT-FILE
```

where AUTH-DIR-1 and 2 are the directories containing texts from the authors; PROB-FILE-1 and 2 are the output files for the probability distributions; and RESULT-FILE is where the sentences are saved.