CSE 482: Big Data Analysis (Fall 2022) Homework 2

Due date: Monday, October 31, 2022 (before 23:59pm)

Submit the homework via D2L.

1. For this Python programming assignment, you will need to implement the Misra-Gries algorithm to find the frequent Wikipedia articles edited by users. Your code will take as input the `wiki_edit2.txt` file provided from the homework web page using a dictionary that can store at most 1000 article titles and their respective frequency counts. You need to use the buffer replacement strategy described in class for the Misra-Gries algorithm. Display the final results of the buffer, which includes the article titles and their estimated frequencies. You should implement the code using iPython notebook and submit the solution in a file named `misra_gries.ipynb` (which should display the results of your algorithm).

2. Download the stock market data set, `stock.zip`, from the class web site. The zip file contains the prices of stocks for 9 publicly traded companies in the following sectors: finance and banking (BAC, C, and WFC), automobile industry (F, TYM, and HMC), and computer technology (AAPL, GOOG, and MSFT). Write an ipython notebook named `stocks.ipynb` that performs the following:

   (a) For each data file (e.g., `aapl.csv`), use Pandas DataFrame to load the data.

   (b) For each stock, calculate the change in the closing price of the stock from its previous trading day. For example, if the closing prices of microsoft was 62.14 on Dec 30, 2016 and 62.90 on Dec 29, 2016, then price change is -0.76. You can ignore the change in the stock price for Jan 3, 2007 since you do not have the stock price on its previous closing date. Hint: you can use the examples from `exercise4.ipynb`.

   (c) Create a new DataFrame named stocks that has 9 columns (one for each stock). Each row corresponds to a given trading date.

   (d) Calculate the pairwise correlation between the 9 stocks.

   (e) Calculate the average correlation between stocks within the banking and finance group (BAC, C, WFC). Compare it against the average correlation between the finance and technology stocks (e.g., BAC-AAPL, BAC-MSFT, BAC-GOOG, C-AAPL, etc). Do the stocks within the banking and finance group have higher within-group average correlation compared to their between-group average correlation with respect to the stocks in the technology group?

   (f) Apply principal component analysis to extract the top-3 principal components of the data (where each component corresponds to an eigenvector of the covariance matrix). Plot a bar chart that shows the contributions of each stock to each principal component (see slide 35 in lecture 7).

3. Draw the decision tree that perfectly classifies each of the data sets given below. There could be more than one answer to each question (you only need to draw one). Assume there are no noise and missing attribute values in the data. Note: you should be able to draw the tree without explicitly writing a code to do this.

   (a) Consider a data set with three Boolean attributes, A, B, and C, and a binary class label y whose value is True if the number of attributes with True values is even and False if it is odd (assume 0 is even). For example, if A=True, B=True, C=False, then y=True (because there are two attributes with True values).

   (b) Consider the two-dimensional data set shown in Figure 1. Each data point is classified either as class A or class B depending on its x and y positions in the two-dimensional plot.

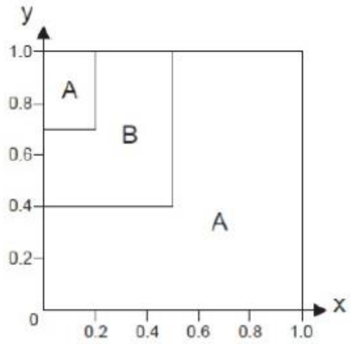

Figure 1: Example of a 2-d dataset.

4. Consider the following training set for classifying whether a patient has a particular disease (denoted as + class in the training data) based on his/her reported symptoms.

Fever	Diarrhea	Stomach Pain	class=+	class=−
Yes	Yes	Yes	8	2
Yes	Yes	No	6	2
Yes	No	Yes	4	0
No	Yes	Yes	5	3
No	No	Yes	1	8
No	No	No	1	10

Each row shows the number of training examples from each class (+ or −) that have the combination of attribute values. For example, among those patients who have fever and diarrhea but no stomach pain, 6 of them belong to the positive class and 2 of them belong to the negative class. Draw a 2-level binary decision tree to classify the patients using gini as the node impurity measure. A 2-level binary decision tree has a root node with at most 4 leaf nodes.

(a) Compute the overall gini for the entire training data.

(b) Calculate the gini after splitting the training data based on the candidate attributes Fever, Diarrhea, and Stomach Pain. Based on their gini values, which attribute should be chosen as root node of the tree? Show your calculations clearly.

(c) Repeat the previous exercise and determine which is the best splitting attributes for the left and right child of the root node.

(d) Calculate the training error of the decision tree.

(e) Use the decision tree to predict whether a patient who has fever but no diarrhea nor stomach pain has the disease.

5. For this question, you will apply classification methods to recognize user activities from their chest-mounted accelerometer data. Download the `activity.zip` file from the homework web page. After unzipping, you will find there are 2 separate CSV files, one file per user. Each CSV file contains a set of rows and four columns. The first three columns represent the x, y, and z values of the accelerometer readings, while the last column represents the class (standing, walking, or sitting down). You need to build 3 models from the data set:

Model 1: Train a single, global model from the combination of training data for both users. Apply the same model to the test data of both users.

Model 2: Train a separate (local) model for each user. For example, the model for user 1 is trained using the training data for user 1, whereas the model for user 2 is trained using the training data for user 2. Apply the models to their respective test sets (e.g., model for user 1 should be applied to the test set for user 1).

Model 3: Train a model using the training set for user 1 only. Apply the same model to the test sets for user 1 and user 2.

To do this, you need to create an ipython notebook named `activity.ipynb` that performs the following tasks:

(a) Partition the data for each user into separate training and test sets. Use 20% of the data for training and the remaining 80% for testing. You may use the `train_test_split` function shown in lecture 9 to do the splitting. Make sure you set the random seed to 1 (to ensure the experiment is repeatable). At the end of this step, you will have 4 data sets: train1, test1, train2, and test2.

(b) Apply the logistic regression classifier to the training data to construct the 3 different models described above.

(c) Calculate the accuracies of the different models on the combined test data for users 1 and 2. Which method has the highest accuracy?

Submit the ipython notebook as your solution for this question.