Homework 5

Overview

This assignment may be completed with your group. Each group will submit one copy of their homework responses to blackboard.

You will turn in both the .Rmd file and the knitted .pdf file to blackboard. I have provided a template R markdown file for you to use (on blackboard).

Objectives

The goal of this homework is as follows:

- Identify good data visualizations published in the media
- Apply K-nearest neighbors classification
- Apply decision tree classification

Grading

- Uploaded *all* requested files, 5%
- Files are properly/clearly formatted, 5%
  - Proper section headers for each part of your homework.
  - You clearly indicate which question each of your responses are associated with.
- Part A: 30%
- Part B: 30%
- Part C: 30%

Deliverables

- .Rmd file used to knit your .pdf file
- .pdf file knitted from your .Rmd file

I strongly recommend that you read over your knitted .pdf file before submitting your homework to make sure that it is formatted as you expect.

Setup

Download the R markdown template provided to you (on blackboard), and answer all of the questions below. The template is meant to help you organize/format your responses. You will need to add explanatory text and code chunks as necessary (be sure to update your names!).

For this assignment, you will need to use the following R packages:

- `rpart`
- `rpart.plot`

You may need to install them if you have not already.

Part A - Effective data visualization

Find *two* examples of effective data visualizations that communicate a point. For each, embed an image of the visualization in your homework and state where the image came from (be sure that I will be able to find the original source). If you don't know how to embed images in an R markdown document, see this: https://www.markdownguide.org/basic-syntax/#images; I am also happy to help you if you have trouble.

For each of the two examples that you find, answer the following questions:

- what data are being visualized?
- what point is the visualization trying to communicate?

Part B - K-nearest neighbors classification

The table below shows information for 14 patients that have been diagnosed with either having a cold or not having a cold. For this part, we will use K-nearest neighbors to predict whether a new patient has a cold or not given the data we have on 14 other patients (given below).

The predictor attributes include:

- `sore_throat` (burning, scratchy, tender)
- `caugh` (yes, no)
- `fever` (yes, no)
- `congestion` (yes, no)

`patient_id` is used to uniquely identify individual patients and **should not** be used for classification. `cold` (yes, no) is the target attribute (i.e., the attribute we'd like to predict given the predictor attributes).

Table 1: Cold Symptoms

patient_id	sore_throat	caugh	fever	congestion	cold
1	burning	annoying	yes	no	yes
2	burning	annoying	yes	yes	yes
3	scratchy	annoying	yes	no	no
4	tender	persistent	yes	no	no
5	tender	debilitating	no	no	no
6	tender	debilitating	no	yes	yes
7	scratchy	debilitating	no	yes	no
8	burning	persistent	yes	no	yes
9	burning	debilitating	no	no	no
10	tender	persistent	no	no	no
11	burning	persistent	no	yes	no
12	scratchy	persistent	yes	yes	no
13	scratchy	annoying	no	no	no
14	tender	persistent	yes	yes	yes

Given a new patient with the following attributes:

- sore_throat = burning
- caugh = annoying
- fever = no
- congestion = yes

Answer the questions below. For these questions you may use R or work them by hand.

1. Use Hamming distance to calculate the the distance between the new patient and each of the 14 known patients.
2. Identify the 5 nearest neighbors. That is, identify the 5 most similar patients (smallest Hamming distance) to the new patient.
3. Based on the class of the 5 nearest neighbors, classify the new patient as having a cold or not.

Part C - Decision tree classification

Next, we'll use R to construct a decision tree classifier for the same cold symptom data we used in Part B. For this, we will use the `rpart`, `rpart.plot`, and `caret` packages.

1. **Data preparation.** The cold symptom data is available as a .csv file on blackboard. Download the data and load it into R as a dataframe. To use the `rpart` function, we should turn our predictor and target attributes into factors. After loading the data into R, turn the following columns into factors: `sore_throat`, `caugh`, `fever`, `congestion`, `cold`.

2. **Build a decision tree.** Use the `rpart` function to train a decision tree on the cold symptom data. Your target attribute should be `cold`, and your predictor attributes should be `sore_throat`, `caugh`, `fever`, and `congestion`. Set the `minsplit` parameter to 1 by adding the following argument to your `rpart` function call: `minsplit = 1`. Remember to assign the model built by calling the `rpart` function to a new variable.

3. **Visualize your decision tree.** Use the `rpart.plot` function (part of the `rpart.plot` package) to visualize your tree.

4. **Use your decision tree to make a prediction.** Using your decision tree, predict whether a new patient with the following attributes has a cold: sore_throat = burning, caugh = annoying, fever = no, and congestion = yes. You may use R to make the prediction or trace your decision tree by hand.

5. What is the root split in the decision tree that you trained? Why?

6. Would you trust the decision tree that you trained to diagnose patients in a healthcare facility? Why or why not?