Northeastern University
Khoury College of
Computer Sciences

Practicum 1

The practicum week is intended to deepen that material and foster discussion where you can learn from your peers and from problems encountered in solving the various questions. Therefore, interact on the discussion board, explore, investigate, share, and respond to each other. You can earn numerous bonus points -- alternatively, you can choose which problems to solve. You may share the answers (but not the code) for problems 1.8, 1.11, and 2.5.

This is a group practicum which means that you may (but do not have to) work in groups of up to three students. You may fully collaborate and submit the same work. However, you must put all three students' names on all submitted work. If a group member is not adequately contributing, the remaining team members may "vote to eject" the student from the team by emailing me the reason. In such an event, the team member who was "fired" must still complete the project individually by the due date.

Collaboration means that you work together on the problems. It does not mean that each group member works on a different problem as you will not learn with that approach. For example, it would not be acceptable to have a group of two, and have one person do problem 1 while the other does problem 2. We may ask some students to present their work in a one-on-one session so we can be sure that the work was done and understood by the student.

Problem 1 (85 Points)

1. (0 pts) Download the data set Glass Identification Database along with its explanation. Note that the data file does not contain header names; you may wish to add those. The description of each column can be found in the data set explanation. This assignment must be completed within an R Markdown Notebook.
2. (0 pts) Explore the data set as you see fit and that allows you to get a sense of the data and get comfortable with it.
3. (5 pts) Create a histogram of column 2 (refractive index) and overlay a normal curve; visually determine whether the data is normally distributed. You may use the code from this tutorial.
4. (5 pts) Does the *k-NN* algorithm require normally distributed data or is it a non-parametric method? Comment on your findings. Answer this in a code block as a comment only.
5. (5 pts) Identify any outliers for the columns using a *z*-score deviation approach, *i.e.*, consider any values that are more than 2 standard deviations from the mean as outliers. Which are your outliers for each column? What would you do? Do not remove them the outliers.

6. (10 pts) After removing the ID column (column 1), normalize the numeric columns, except the last one (the glass type), using *z*-score standardization. The last column is the glass type and so it is excluded.
7. (10 pts) The data set is sorted, so creating a validation data set requires random selection of elements. Create a stratified sample where you randomly select 20% of each of the cases for each glass type to be part of the validation data set. The remaining cases will form the training data set.
8. (20 pts) Implement the *k-NN* algorithm in R (do not use an implementation of *k-NN* from a package) and use your algorithm with a *k=5* to predict the glass type for the following two cases:
   RI = 1.51621 | 12.53 | 3.48 | 1.39 | 73.39 | 0.60 | 8.55 | 0.00 | Fe = 0.08
   RI = 1.5893 | 12.71 | 1.85 | 1.82 | 72.62 | 0.52 | 10.51 | 0.00 | Fe = 0.05
   Use the whole normalized data set for this; not just the training data set. Note that you need to normalize the values of the new cases the same way as you normalized the original data.
9. (5 pts) Apply the *knn* function from the ***class*** package with *k=5* and redo the cases from Question (8). Compare your answers.
10. (10 pts) Using your own implementation as well as the **class** package implementation of *kNN*, create a plot of *k* (x-axis) from 2 to 10 versus error rate (percentage of incorrect classifications) for both algorithms using *ggplot*.
11. (5 pts) Produce a cross-table confusion matrix showing the accuracy of the classification using *knn* from the **class** package with *k = 5*.
12. (10 pts) Download this (modified) version of the Glass data set containing missing values in column 4. Identify the missing values. Impute the missing values using your version of *kNN* from Problem 2 below using the other columns are predictor features.

Problem 2 (30 Points)

1. (0 pts) Investigate this data set of home prices in King County (USA).
2. (5 pts) Save the *price* column in a separate vector/dataframe called *target_data*. Move all of the columns except the *ID*, *date*, *price*, *yr_renovated*, *zipcode*, *lat*, *long*, *sqft_living15*, and *sqft_lot15* columns into a new data frame called *train_data*.
3. (5 pts) Normalize all of the columns (except the boolean columns *waterfront* and *view*) using min-max normalization.
4. (15 pts) Build a function called *knn.reg* that implements a regression version of *kNN* that averages the prices of the *k* nearest neighbors using a weighted average where the weight is 3 for the closest neighbor, 2 for the second closest and 1 for the remaining neighbors (recall that a weighted average requires that you divide the sum product of the weight and values by the sum of the weights).

   It must use the following signature:

   knn.reg (new_data, target_data, train_data, k)

   where *new_data* is a data frame with new cases, *target_data* is a data frame with a single column of

prices from (2), *train_data* is a data frame with the features from (2) that correspond to a price in *target_data*, and *k* is the number of nearest neighbors to consider. It must return the predicted price.

5. (5 pts) Forecast the price of this new home using your regression *kNN* using *k* = 4:
   bedrooms = 4 | bathrooms = 3 | sqft_living = 4852 | sqft_lot = 10244 | floors = 3 | waterfront = 0 | view = 1 | condition = 3 | grade = 11
   sqft_above = 1960 | sqft_basement = 820 | yr_built = 1978

Submission Details

- Graded out of 100. Maximum points 115/100 (+15 bonus points)
- Your submission must contain two files: the *.Rmd* notebook **and** a knitted PDF or HTML (from the notebook). Name your *.Rmd* R Notebook, DA5030.P1.*LastName*.Rmd and your PDF DA5030.P1.*LastName*.pdf, and your HTML DA5030.P1.*LastName*.html, where *LastName* is ***your* last name**. If you are producing an HTML instead of a PDF, be sure to ZIP the HTML file as Blackboard does not allow uploading of HTML (or it will "mungle" it and it won't be viewable).
- The *.Rmd* file must be fully commented and properly "chunked" R code and detailed explanations. Make sure that it is easy to recognize which question you answer and that your code runs from beginning to end (because that is how we will test it.) Code that doesn't execute, stops, throws errors will receive -- naturally -- receive no points. If the graders have to "debug" your code or spend any effort getting it to run, substantial points will be deducted.
- Not submitting a knitted PDF or HTML will result in reduction of 30 points.
- Not submitting the *.Rmd* file (or both) will result in a score of 0.
- We must be able to run your code. No points are awarded for code that does not run.
- The submission link is in the Assignments folder on Blackboard. No email submissions are accepted.

Useful Resources

- Quick Guide to Creating Scatterplots in R with ggplot
- R Markdown Tutorial
- Kumar, S. (2017, Oct. 2). The Art of Story Telling in Data Science and how to create data stories? Analytics Vidhya.

Learning

Blackboard

Lynda.com

Data Camp

Support

Contact Instructor

Virtual Office

Book Appointment

Search