Homework 4 Feedback for Stat 495 **Class Feedback**

Please note that this is a compilation of feedback from several years of having Stat 495 students work on this or similar open-ended data analysis assignments. It is updated every semester, and not all comments may be relevant to this particular assignment. However, they should provide general guidance for your data analysis, and in some of them, I have tailored them to this assignment.

If I saw particular issues in your submission, I made comments in Gradescope, so be sure to look for those. If the issues were covered here though, I may not have made a comment – expecting you to look through these.

1. Check that your understanding of the problem matches that of other consultants (your peers) and the client (me, in lieu of the real estate developer). This is why discussing the problem with the client is so vital. See also #17 below.

2. Be extremely wary of “when” in the course of an analysis you decide to drop observations with missing values (assumes missing values are present). Example: I have 3 predictor variables, 2 are nearly complete and the third has 80% missing data. I drop missing values at the start of my analysis. Then I build a model and only use the 2 nearly complete variables in it. Was this an appropriate place to drop my NAs? Also, sometimes NAs are meaningful. You should check them out before doing anything to them.

3. Check that every variable is being treated by R in a manner that makes sense. (e.g. are numerically coded categorical variables being treated as numbers or factors?) This should be done BEFORE you fit anything. If they aren’t, when R treats them as numbers, it assumes 1 unit increases are meaningful. For example, if a variable has levels 1 through 5, and you treat it as a number, then 4 is really twice 2, rather than being just 2 levels different.

4. Consider variable relationships among predictors. You want to avoid rank deficient matrices / linear dependencies. Discuss your variable selection.

5. Consider useful re-expressions of predictors (even if they don’t end up in your final model). I.E. Don’t just assume that you should or have to use the predictors as they are presented to you. This may be useful in situations where you see predictors don’t have a lot of variability – perhaps a presence/absence or other indicator could be more useful than the original variable itself.

6. Latitude and longitude are present in the data set. Even if you don’t overlay the observations on an actual map, you can make plots to examine variables with lat/long as the grid. Should these be predictors?

7. You should consider if there are any issues with independence or randomness for you to comment on. How far can your results be generalized?

8. Examine the distribution of your response variable to help set baseline expectations for prediction.

9. You should check for unusual points. If you remove outliers (for the response or predictors), how do you assess their impact on your analysis?

10. Determine if a training/test set is useful for your analysis. Should you use different training/test sets for your tree and GLM? Use a method that controls the size of your train/test set. It should NOT have an individual probability each observation is in either set. See feedback on Hmk 3.

11. Make sure your code is reproducible. Reported values in sentences should match what the output shows.

12. Check over your code and output to make sure that R is running the analysis you wanted it to. (If you ran GLM without specifying the family option, you got ordinary least squares fits as a result, which was probably not what you wanted.)

13. For each method, refine models obtained from automated selection procedures, if you ran them, and examine your model summaries. Look at and talk about model coefficients and properties (such as significant predictors), not just assessment measures like MSE. We shouldn't assume that forward selection or backward elimination resulted in a model with no problems. It's your job to check out the final model before presenting it!

14. Discuss all options you are setting for the models and why you are setting them at the values you are.

15. Use the power of the tidyverse and ggplot2 for visuals. The results look much nicer than base R commands.

16. How are you treating zipcode if it is being included in your model? Are there enough observations in each zipcode to not overfit the model and reliably estimate other parameters (especially if interacting it with anything)?

17. **What are appropriate GLMs and trees? This depends on your response variable. What response variable are you using?** If it wasn't clear from earlier comments, or the discussion in class, we aren't predicting "price". We are predicting whether or not price is greater than half a million dollars. This means you need to create a new variable to serve as the response variable. There ARE ways to predict "price" and then convert everything into the "over" or "under" idea, but that was not the intention, and you can get different results if you go this way (and have to make different decisions based on variable relationships).

18. When using *predict* or any other function that has multiple potential input objects, be sure you are using the right inputs for your object, and that you are getting the output desired out.

In other words, predict.glm and predict.rpart might have different inputs, or give different results, or have different arguments you need to set. Some of you are using these incorrectly trying to get MSEs. Think about what we said the MSE for a logistic regression should be equivalent to in class.

Looking ahead to Hmk 5 - suggestions:

1. Follow the guidance provided about the material for each section including what you brainstormed in class. Remember the real estate developer has a question of interest and didn't give you the data. They only know what you tell them.

2. Check that your code doesn't run off the page in your .pdf. (Same for comments in code).

3. Check that your figures don't run off the page in your .pdf. Check that they aren't too small either. You may not have precise control over their positioning, but you have control over their size. Watch the titles and make sure those don't get cut off.

4. Use RMarkdown formatting to help keep your presentation organized.

5. Check that your figures have appropriate captions for someone else to follow along. Do you need to rename any variables to make output readable for your audience?

6. Appropriately intersperse comments between figures/output so that someone else can follow along. A page of output can be hard to parse if all comments about it are at the end.

7. Ask a classmate to review/read over your submission to see if they can follow along.

8. Run spell-check – this exists within RStudio. Spell-check works on text, not R chunks – you have to check those yourself. Proofread your .pdf.

9. Check that your final models make sense. This is your responsibility as the analyst. Do you have any cautions for someone using your model? If so, you should state them!

10. Remember your audience. The idea is to present the analysis so the developer can follow along the entire way, without needing to make assumptions about why you are doing what you are doing.

11. Read over your output and look for warning messages/notes. Be sure you understand why you are seeing them, and comment appropriately.

12. Coding conventions. Adopt an R coding style, such as the one we used in Stat 231, or Google's, shown here: https://google.github.io/styleguide/Rguide.xml

Even if the variable names, etc. are already set in your dataset, you can follow a style guide for spacing, etc. to help make your code more readable.

13. Work on guidance for the reader / flow in your writing. Don't underestimate the power of a transition sentence – the signposts. In our labs, you will often see me do something like this:

CODE for X

Now that we've done X, we can proceed to do Y.

CODE for Y

This can help with flow, organization, and understanding the analysis. It helps the reader know where you are at in the analysis.