Task 1 Course 3 --- Find the Errors Task Report

In this report, I will present the prediction results of the Linear Models given the car dataset and the iris dataset. These predictions will also be supplemented by scatter plots. After that, I will discuss about any issues that I may have encountered during this project. Finally, I will give my thoughts R and RStudio (installation process, tutorial, lessons learned) and how I would compare them to Python and Jupyter Notebook.

In the last section, regarding my thoughts about R and RStudio, sentences that are bolded are the ones that answer the secondary requirements of the prompt that this document is responding to. No need to read the meandering stream-of-consciousness, especially since a lot of it is just whiny.

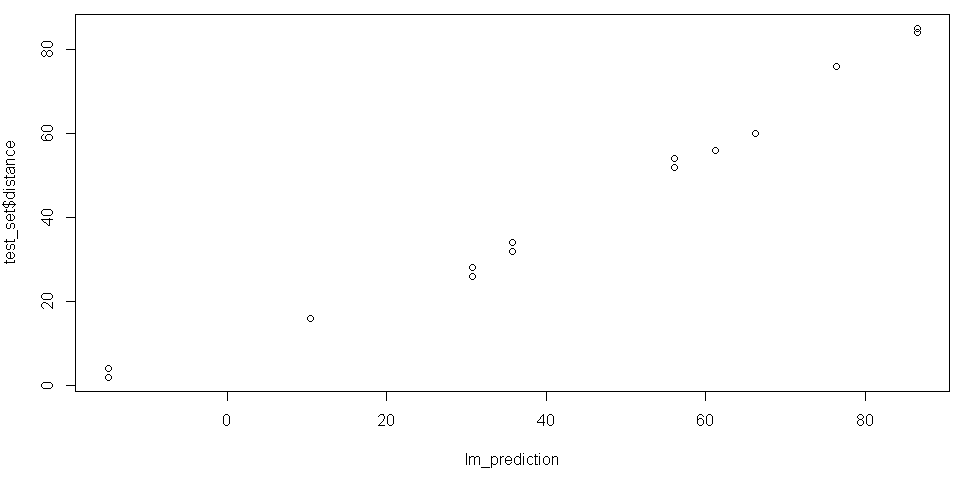
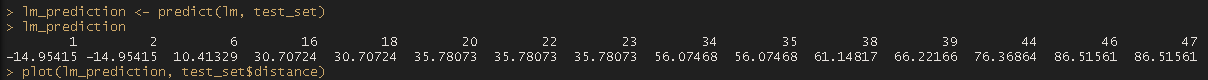
Results for the car and iris datasets

The prediction results and their respective graphs are as follows, starting with the car dataset followed by the iris dataset:

Images for the cat dataset

While not pictured, the Multiple R-Squared score was 0.9248 and the p-value was well below 0.05 (2.2e-16). It appears that the Linear Regression model fits well with the test set for the car dataset. It appears that the dependent variable of distance is related to the independent variable of speed.

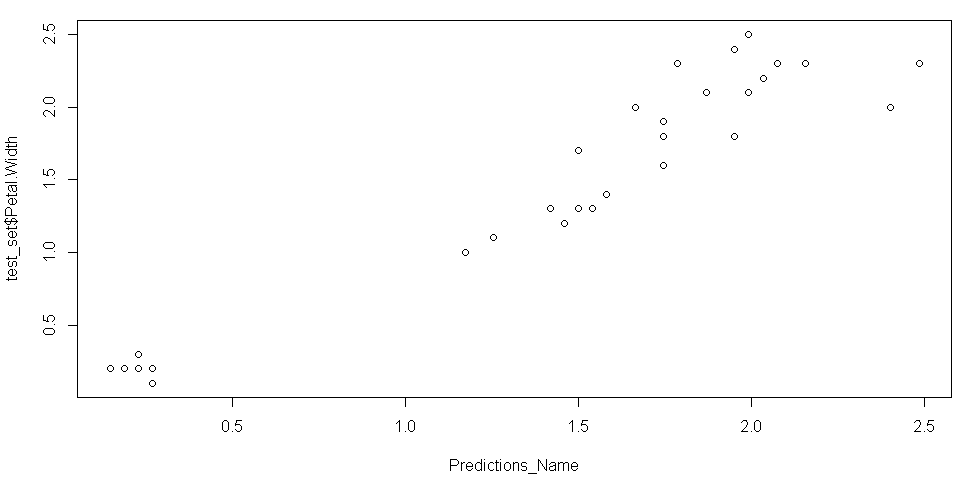
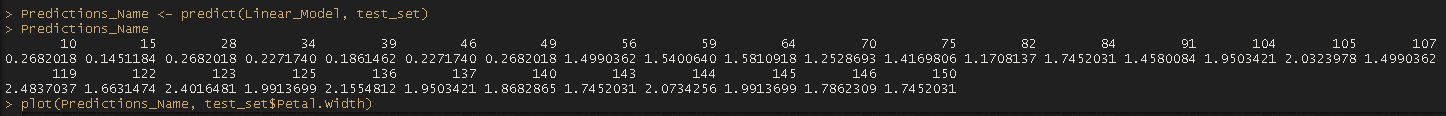
That being said, It would be worth better understanding how explanatory the graph is or the metrics mentioned above, in ascertaining how truly performant the Linear Regression algorithm is in making sense of the relationship between distance and speed. This suggestion is so that we don’t come up with inaccurate generalizations from the results of using the Linear Regression algorithm for this dataset.



Images for the iris data

While not pictured, the Multiple R-Squared score was 0.9281 and the p-value was well below 0.05 (2.2e-16). It appears that the Linear Regression model fits well with the test set for the iris dataset. It appears that the dependent variable of petal width is related to the independent variable of petal length. However, there appears to be some noise, according to the graph, so this relationship may not be as clear or strong as that of the car dataset.

That being said, It would be worth better understanding how explanatory the graph is or the metrics mentioned above, in ascertaining how truly performant the Linear Regression algorithm is in making sense of the relationship between petal width and petal length. This suggestion is so that we don’t come up with inaccurate generalizations from the results of using the Linear Regression algorithm for this dataset.



Initial thoughts on R, RStudio, and comparing them with Python and Jupyter Notebooks

**It was straightforward to install R and RStudio. As far as the tutorial was concerned, it was decent** in guiding me on what I needed to do to actually “do work” with the language and the IDE, but I would have preferred and benefited greatly from an experience similar to the Python pre-work that was done just before the program formally started. To be fair, I already came into this program with some familiarity and comfort with using Python and Jupyter Notebook (although I certainly learned so much more about both of these tools) --- **I think the R tutorial overall could be improved by providing far more depth (**again, **think of the Python pre-work).** For example, although it probably is outside of the scope of the program, or may be “non-essential”, it would be nice to understand how and why the “tidyverse” package-collection is so valuable for conducting data analytics and data science using R and RStudio. I mean, after all, with Python, it’s been hammered down that we are to use numpy, pandas, matplotlib.pyplot and sci-kitlearn as primary Python libraries – but no such priming was given as it relates to R. Of course, to be fair, all of this can be learned independently and outside of the scope of the program, but it just does not seem to me that the treatment between the two tools (Python and R) are remotely comparable or “even”. I am left wondering why it is even considered that students should be exposed to R apart from the basic fact that it is a commonly used tool for data analytics and data science. **That being said, I would recommend the tutorial to others.**

Having said that**, I believe the “Fix the Errors” assignment was informative, albeit at times irritating (given lack of instruction/clarity). The most important aspect of the assignment is that it reinforced to me --- again --- that understanding data types and the “functional limits” (what kinds of operations can be applied to certain data types) is fundamentally important and must always be at the forefront of my decision making.** At the end of the day, R and Python can accomplish the same things and they are both tools. The same considerations of statistics, domain knowledge, the data science pipeline (understanding problem, data collection/data retrieval, data cleaning, EDA, feature engineering/selection, model creation, model deployment, etc.) are practically the same when using Python and R. While they have different function names, different libraries and different syntaxes, both languages are fully capable of engaging with the DS pipeline.

**As far as recommendations are concerned, my recommendations are as follows**:

1. **“Match up” R functions with different tasks or aspects of the Data Science pipeline** (understanding the problem (why does this project exist and what should it mean for the organization), data collection/retrieval, data cleaning, EDA, visualizations, feature selection, model creation, model deployment, presenting, etc.)
2. **Consult resources**, like the official R documentation, Stack Overflow, YouTube, blog posts and cheat sheets.
3. (Not related to the scope of Course 3, given that we are not introduced to this) **Utilize** the **Tidyverse** package collection. The packages that make up the Tidyverse are capable of dealing with recommendation 2 (at least some of it, as far as I’m aware)
4. **Have a side-by-side comparison of doing DS work between Python and R.** Do this comparison line-by-line of both languages. This should help to reinforce their similarities, differences and clear up syntactic confusion.
5. **Use your tools for your context, as needed. Focus on being conversant with your tools.** Don’t over think or worry too much about every aspect of tooling (R and Python are tools), especially as a novice learner. The other aspects of tooling, such as memory, runtime, debugging, code maintenance, etc., matter when you’re already professional. If your context with the tools is doing dashboarding then don’t stress out over feature engineering. The context of dashboarding requires different concerns and tools than the context of feature engineering.