Connor Emmons

Final Project Assessment

Connor-Lemons/Palmer-Penguins-DevOps: Chapters 1-6 of DevOps for Data Science (github.com)

Note: In some portions of this write up, I'll use the phrase "thus far". The reason here is that I intend to continue working on the project after the deadline and after this class is over because I'm having a really good time with the modeling and I feel like I'm really solidifying my understanding of the models. **As it stands for the official assignment, my project is good to go. I do not intend for me saying "thus far" to indicate potential incomplete or late work, just that I plan to continue developing this on my own time.**

Learning Goals

Develop an applied knowledge of traditional and computationally intensive statistical models

While this assignment was likely intended to target the other two learning goals of this course, I do think that I achieved this learning goal as well. Thus far, I have implemented linear discriminant analysis to predict both the species and the island of origin of the penguins based on the four numerical measurements, and I have also implemented quadratic discriminant analysis, Naïve Bayes, and k-nearest neighbors methods to predict the island of origin for the penguins. In implementing these methods, I reinforced my understanding of each of these models and exactly how they work. For example, when implementing linear discriminant analysis, I actually took a closer look at the fitted model because I wanted to include a paragraph explaining what it was. When I did so, I saw that the model output actually tells you the coefficients for each of the parameters which make up the linear combinations the model uses to classify. Then, in the plot, each data point could be expressed in terms of the weight given to each combination. Conceptually, I kinda knew that this is what LDA was doing, but now I really understand how the modeling works and how it classifies certain points.

In addition to implementing these models, I also performed k-fold cross validation with 10 folds on the k-nearest neighbors model to find the optimal number of neighbors. Similar to above, this reinforced my understanding of the concept and strengthened my comfort in the code, which I think is really important. Every time I have to make these models or implement cross-validation, I reference my past code or ChatGPT a little less, which to me indicates a stronger understanding of the processes as well as an increased familiarity with the code.

In the future, I hope to implement a variety of tree-based methods to model this same classification problem, all with cross validation. I'm mostly doing this because I think it's super cool that the code I'm writing can be found on a website instead of just in R studio or posit cloud, but also because I'm hoping to learn more about these modeling techniques just like I did with LDA over the course of this project.

Advance, use, and improve computer programming skills

This was definitely the learning goal that was hit hardest during this project, and I'm happy that I struggled through as much as I did. I learned a ton of different things about programming, and I'll try to go through them in order of importance. I think the most important thing I learned in this project was

how to use GitHub, which doesn't really make sense because I've known how to use it for a long time and I'd consider myself quite an adept GitHub user. However, this project taught me how to do GitHub actions, which I though were super cool and very useful (pushing to a repo and automatically publishing the website was something I never knew you could do). To speak more generally, I think that knowing Git is absolutely critical for any programmer, even a recreational one like myself, and any opportunity to learn a new skill in Git or just to practice and solidify old skills is huge. I think that the next most important programming skill I learned from this project was how to manage python. When it comes to coding in python, I'm fairly well-versed, but I always just open up a python IDE and start there. Learning how to manage the virtual environment and dealing with installing packages on the command line for my code in R studio taught me a lot about how python is integrated into a computer, and while I'm not sure that it will be super useful for the future, I definitely appreciate doing it. I also learned a very important thing about python: If you want to code in python, don't use an R editor.

Lastly, this project helped me develop my skills in statistical programming with R. I've ranked this as the least important because I'm already pretty comfortable with R, but really it's because (in my opinion) learning R is a dead end. With python getting many of the statistical modeling tools that made R so good for stats, I don't think that there's a place for R and I would love to see this class move to python exclusively as soon as possible. On a separate note, I'd like to say that I gave the apps an honest try and I did learn a lot (how to open an API, how to test an API to make sure connectivity is established, how to test an API to make sure it can run things like the predict() function) but I could never get the app to work. It would always give me this error: Error in httr2::req_perform: HTTP 422 Unprocessable Entity, and I could never fix it. In my mind, the website was the more important product anyway and I'm happy that I learned the things that I did from trying the app, but in the end I ended up scrapping those in order to focus on the website which I was having way more fun with.

Continue to develop the habits of mind to be an independent learner

This one was also definitely hit hard during the course of this project. The project did a great job of explaining the conceptual aspect of the labs, but had little guidance in the way of execution. I think that's what made this project so fun and useful, because I had to practice being given a task and finding the solutions myself, even if I didn't know how. This typically involved a lot of ChatGPT, which is another skill I think is critical to an independent learner. Knowing how to structure prompts and work with ChatGPT was absolutely essential to completing this project, and I had a lot of fun trying to solve problems myself with only ChatGPT to assist me. As I've mentioned in the previous paragraphs, I plan to keep working on this project after the assignment and after this class just because I've been having so much fun with it, and I think this really shows that I've taken the independent learner mindset to heart. Overall, though I know that most will complain about the lack of guidance, I thought it was really nice (though I guess some guidance could have gotten my apps to work so there's that argument).

What skills, knowledge, or experience did you gain from working on this project? Discuss how you went above and beyond.

First, to answer what I gained from this project. First and foremost, I gained a real appreciation for how difficult it is to present your work to others. Making the models in R and doing the analysis is (relatively) simple, but having to deploy a website, figure out GitHub integration, make apps, and generally present your work to others in an easy-to-understand way is really, really, really difficult. I will say, I have never really known how to make a website, so this is definitely a skill that I picked up in this project. As I

mentioned above, I learned a lot of new things about Git and strengthened my understanding of things I already knew, which is absolutely essential for modern programming (seriously, I think that before people are taught “Hello world” they should be taught Git).

I think that one of the most valuable experiences I gained through this project was working on a project with a larger scope than what I’m used to. Most of the coding I do for classes or on my free time are small tools which really only accomplish one task. These run locally on my computer, and sharing them with friends typically involves a team message. This project was much larger in scope, requiring me to think about a lot of different aspects, from the code itself to the website, from the apps to GitHub integration. The thing about a project of this scale is that it’s really easy to get overwhelmed or to try and work on the whole thing at once instead of breaking it up into components, and getting the chance to practice “taking small bites to eat the elephant” was really valuable to me.

As for what I added to this project, I’d first like to start off by acknowledging that I did not successfully complete labs 1-6. I truly gave it my best shot, but I couldn’t get past the error I listed above when trying the app. My additions to the project mainly focused on data visualization and the modeling. I created various graphs to visualize the data (specifically, I created box plots comparing each of the three levels in the category, islands or species depending on which page of the website you look at, as well as distributions comparing each of the penguins versus each of the four numerical variables). For the modeling, I implemented LDA, QDA, Naïve Bayes, and KNN, as well as k-fold cross validation. I also tried my best to explain each chunk of code as if I was walking someone through their first adventure into statistics, and I’m decently proud of the analysis I put in there. Setting the apps aside, I believe that I deserve an “A” on this project because I did do my best to create a product that went above and beyond the labs and, more importantly, a product that I was proud of. Taking the apps into consideration, it depends on the goal of the assignment. If the goal is completion, then I could not have earned above a B because I didn’t complete the assignment. If the goal was learning, then I think the argument could still be made for an “A” because I did learn a lot about apps and APIs, I just don’t have anything to show for it. As a note, I kept the two initial files made in lab 1 as proof that I did them (they are labeled proof 1 for the r file and proof 2 for the python file) but they are not intended to be a part of my website. I also added short descriptions to the Home and About pages.

What are your strengths and weaknesses after working on this project?

Honestly, I think my greatest strength when it comes to difficult projects like this is that I enjoy them and have fun with them. I really do enjoy solving every error that pops up and learning how to do things from scratch. It’s definitely tough going, but the reward of having that lightbulb moment after doing all the legwork to learn on your own is a feeling like no other, and I live for it. I’ve always been a relatively patient person, which I also think helps with dealing with frustration and not getting overwhelmed when stuff doesn’t work out the first time.

In terms of technical skill, I’d say that one of my strengths after this project is Git. No, I’m definitely not an expert at it, but I’ve reached the level where I’ve started to use Git in my own projects and in small projects for other classes just because I like it, not because it’s required. This started with this project, and I think that this can be considered a strength of mine.

As for my weaknesses, I’d have to say it’s just my complete lack of knowledge of any of this deployment stuff. I can program alright, but I’ve never really taken it to the next level and presented my code in an

app, or a website, or really in any way other than running it on my computer and showing it to my friends. I've gotten better at this as a result of this project, but it's still definitely a weak point in my programming literacy. I'd like to improve this in the future, though realistically, as an engineer, I'd guess that I'd be more focused on coding for design rather than for presentation.

What was the most interesting part of the project?

The part that I found most interesting was the push-to-GitHub-and-autodeploy-the-website part. I still think it's super cool that you can set up a GitHub action to automatically publish your website every time you push code to the repo. Really, I find GitHub interesting every time I use it because it is just that useful for programmers. I also found the statistical modeling really interesting, which makes sense because the statistical modeling is what has made this class my favorite this semester. To be honest, I though that the whole project was interesting.

What are some of the challenges you faced while working on the project? How did you persist through difficulties?

The first issue I encountered was "How the heck to I set up this python virtual environment in R studio?" I kept wanting to do it within R studio, but I eventually (with the help of ChatGPT) I learned that you have to handle all that stuff in Windows powershell. It took me a while to get comfortable with that workflow, mainly because I thought I was doing it wrong the whole time, but I eventually got it and it makes sense now. The next major issue I ran into was trying to get the website to render. It was super inconsistent and I had to restart R like 5 times every time I wanted to render. I found out that the problem was that I was not initializing python correctly in R studio and that as a result, I didn't have any of the correct packages installed. I realized that the only times it worked was when I got lucky and somehow the workspace that I loaded on the restart happened to contain the packages I needed to run the code, but learning how to properly initialize python fixed this. The next major problem I ran into was how to get the GitHub action to work, which I spent forever trying to do until I realized that the template he posted to go along with that section had the code running on an Ubuntu VM, and I needed to run it on a Windows VM because I had windows-specific python packages. I was really happy to solve this one because after trying and failing with ChatGPT, I went back and read the book and figured it out myself! I've already talked about the app problems, but I will say that I was happy that I gave it as long as I did. Obviously, I didn't do that great because the app still doesn't work, but I did try my best to persevere through the errors and I do enjoy that I learned a lot.

What other resources did you use to complete this project? How did you utilize these resources to learn and improve your skills?

My primary resource when starting these labs was always the book, and I tried to troubleshoot as much as I could from the book before moving to ChatGPT. Realistically, this phase lasted about 2 minutes before I moved to ChatGPT, but I do think that the book was a better troubleshooting source than ChatGPT for easier problems to solve.

About 50% of my Chat GPT usage was me trying to learn how to do various things. These prompts typically weren't very structured, they were just how I would type them into Google. I'd ask things like "How to initialize python virtual environment" or "How to run a python app in R studio". Because I never really gave it my code or told it specifically what I was working on, most of the replies were generic and

contained code that I ended up using, though I had to heavily modify it to fit my project. Even so, almost all the code generated from prompts like these weren't actually content in the project, it was more just setting stuff up.

About 40% of my prompts were just me pasting my code and the error it generated into ChatGPT and asking it what happened. I found this to be the most helpful way that I used ChatGPT, which was to expand on error messages that I got so that I could find fixes. I did use some of the code that ChatGPT generated to fix errors, but again the code generated was general (mostly because I was too lazy to take the time/didn't really want to give it the necessary resources to understand my project and give me a solution).

The last 10% of the prompts were requests to generate code, and I only did this for the data visualization plots in the statistical modeling. Realistically, I probably could have worked through the code to write these myself, but I find that using ggplot2 in R takes a lot of research into the specific functions required and I wanted to focus more on the analysis and the modeling rather than trying to find the one function that I needed. Every model that is presented on the website was typed by me, ChatGPT had no influence on these (I did use my previous code from the course when I got stuck on some things though). The link to my ChatGPT transcript is here: https://chat.openai.com/share/2e9ea816-e197-4a78-befe-0048659a02a5

What is one way you could improve your DevOps skills?

I think there's only one answer to this question, and that's to practice. If I want to get better at DevOps, I have to practice with projects like these, and hopefully every time I deploy a website or make a GitHub action, I'll get just a little better and get a little closer to being a competent developer. For now, I'm really satisfied with the work that I've done and I want to keep making progress on this project, but I also look forward to that time in the future when I take that next step and try to do something like this on my own with a piece of software that I'm proud enough of to want to bring it to the world.

As a final note to Dr. Hitt/Dr. Hauschild, thank you so much for walking us all through this project. I'm sure that having us come to you with bugs or send you messages late at night or generally ask dumb questions that we probably could have figured out if we just read the book got real old real fast, so thank you for having the patience with us to enable us to see this project through. This has been a super helpful and important experience, and I really think that I've learned a lot of good, actually useful skills that I'll be able to use in the future. Thanks!