378_final_project

May 8, 2018

1 I Deserve $118,000 per Year

1.0.1 By C2C Joshua Rackham

1.0.2 Math 378 (T2)

That's right! Upon completion of the "Data Scientist with Python" Career Track in DataCamp, the estimated salary noted on the website is a whopping $118,000/year. Of course no one could expect to make this type of salary in the world of data science without substantial work experience first, but the skill sets I learned by completing this career track are a springboard to immersing myself in the world of data science, and yes, with that a successful career.

In fact, this career track is the most comprehensive Python career track that DataCamp offers; the other two career tracks (including "Python Programmer" and "Data Analyst with Python") simply include course selections from the full list of Python courses found in the "Data Scientist with Python" career track. I completed all of these courses, which are listed below:

List of courses

- Intro to Python for Data Science
- Intermediate Python for Data Science
- Python Data Science Toolbox (Part 1)
- Python Data Science Toolbox (Part 2)
- Importing Data in Python (Part 1)
- Importing Data in Python (Part 2)
- Cleaning Data in Python
- pandas Foundations
- Manipulating DataFrames with pandas
- Merging DataFrames with pandas
- Introduction to Databases in Python
- Introduction to Data Visualization with Python
- Interactive Data Visualization with Bokeh
- Statistical Thinking in Python (Part 1)
- Statistical Thinking in Python (Part 2)
- Supervised Learning with scikit-learn
- Machine Learning with the Experts: School Budgets
- Unsupervised Learning in Python
- Deep Learning in Python
- Network Analysis in Python (Part 1)

Why I picked this career track The thorough, comprehensive nature of this track in DataCamp is the very reason that I selected to complete it for this final project. I wanted to push myself with the most in-depth course load in whichever language I selected. My mindset in determining which language to pick was driven by applicability to real-world analyses. I knew that I wanted to get *really* good at whichever language I picked, and continue to pursue a deep skill in that language until I could leverage its power in the real world. Upon chatting with both military and civilian friends I have that are in the field of data science and upon reading in some forums online, I elected to pursue either R or Python as my language of expertise.

My end-goal with this project was to get a jump start into the world of data science. This fits into my goal to be a 61-A in the Air Force, and eventually drive top-level decisions in the corporate and political world. I believe this is best done with effective data analysis and compelling, interpretible visualizations. I wanted to pick the language that would best help me achieve this end goal. Upon reading more on "R vs Python" in various forums online, almost every trusted source I found essentially said that "Python is used by everyone so you should too". I trusted the masses in this, taking this recommendation (albeit somewhat blindly). As I completed courses in DataCamp, however, I soon discovered that Python has many advantages over R, MATLAB, Mathematica, etc., and I still intend to continue pursuing expertise in analysis and visualization using Python.

My thoughts on this career track I was very pleased with my experience in this career track. Upon starting each course, I felt like I had the prerequisite knowledge sufficient to complete it and understand it, and I felt like the DataCamp instructors were experts who taught well to my level.

To me, the material got more exciting as I completed more courses (in the order they are listed above). For example, I was pretty bored by "Importing Data in Python (Part 2)", because while I knew that the information was important (critical, even), it is all preparation work for actually doing analyses with the data. While that was my least favorite course, my favorite course was "Deep Learning in Python", which was the second-to-last course in the career track. This was my favorite because I was very excited to be learning about neural networks for the first time, and I felt like it was a compelling introduction to the potential that can be unleashed if I spend more time writing machine learning algorithms in Python.

There did not seem to be significant overlap between the information that instructors taught in the courses. In my opinion, the courses did a great job of teaching the skills needed for subsequent courses, and then using those skills frequently throughout the career track, but they did not instruct on the same functions or principles more than once, in most cases. In the one or two instances that instructors did cover a topic twice, it was because they were using a unique function that had not been reviewed or used for several lessons. For example, lambda functions were taught in the Course "Python Data Science Toolbox (Part 1)", and were not used again until the Course, "Deep Learning in Python", 16 courses later. Instructors at DataCamp must have realized that lambda functions are not extremely intuitive, and included a well-needed review on the topic. (To see the unique nature of a lambda function, see code snippet below. I felt like the instructors did a good job of only including redundancies for weird Python-isms like this one.)

```
In [2]: # This is a normal user-defined function that takes the square of a number
        def square(num):
            num = num ** 2
            return num

        # This is a lambda function that also takes the square of a number.
```

```
squareIt = lambda num: num ** 2

# See? They both work but the lambda function just looked weirder and no one would rem
# how to build that unless they use it a lot.
print(square(2))
print(squareIt(2))
```

```
4
4
```

My thoughts on interactive online learning The option to engage in online learning has brought innumerable new prospects to my future as a data scientist. I feel enabled to complete any project, given enough time to research, because the resources are all online. A few specific advantages to online learning (especially in an environment like DataCamp) include the almost infinite accessibility, the interactive nature of having an online coding environment, and the instant feedback that I receive as I complete the exercises. These tools accelerate the pace greatly from the traditional textbook-learning methods that I am used to, and honestly make learning much more fun.

A couple challenges I found this semester include the following, which all boil down to not giving the user enough creativity or independence:

DataCamp requires the solution to each exercise to be structured in a specific way in order to give the user instant feedback. The end goal of each exercise could be achieved using many different structures and code-sequences, but to get this specific structure, DataCamp has to guide the user enough that their submission matches the DataCamp answer key, even down to the order of values multiplied in some cases. This is a distinct disadvantage because it reduces the number of ways a problem can be solved down to one. It also means that I was given step-by-step directions on how to solve every problem, instead of being able to learn from the struggle to find an approach myself, realizing through trial-and-error why some approaches work and others do not. As a side-note, sometimes I would arrive at the solution and still get an error from DataCamp because I had included arguments to functions in a different order, or other insignificant differences.

Because of this step-by-step approach, I feel like I would now struggle to pick up many basic problems and execute them start-to-finish. I never had the opportunity to be given an open-ended problem and come up with a solution, simply due to the required format from DataCamp.

Lastly, I think that the "show hint" and "show solution" functions were a crutch to me. While I tried to debug most snippets myself, there were definitely instances in which I relied on the hints or solutions because I could not debug a certain code snippet before getting impatient. In reality, I will be solving most problems by myself without some robot monitoring my work with a solution already in mind. I do not feel prepared to debug an entire project start-to-finish as a consequence of the error messages, hints, and solutions that are available through DataCamp.

While a couple of these challenges are with regard to the actual completion of DataCamp courses, I think that most of the challenges of an interactive online course like this come after the fact, when the student realizes that they are not as prepared as they may have thought for a real-world problem. I think DataCamp does a fantastic job with its courses, learning in its interface was a pleasure, and I think the challenges it faces are mostly unavoidable without a personal tutor and instructor for each student, much like an online university environment. I recognize that would not be practical or scaleable for DataCamp's purposes.

Additional thoughts I had several courses overlap with concepts I have learned at USAFA. Besides simple concepts such as the slope-intercept form of a line and basic concepts I have known for a long time, I found that all of the content overlap was from Sports Analytics, Math 377, and Math 378. A list of courses that had noteworthy overlap is described here: * "Importing Data in Python (Parts 1 and 2)": The instructors for OR495 (Sports Analytics) used these two courses to teach themselves how to webscrape, and then used their notes to teach us. I found this out within a week or two when I found myself learning from the exact same examples and code snippets when I was doing this course on DataCamp. * "Statistical Thinking in Python (Parts 1 and 2)": I found that these two courses were simply a review principles I learned in Math 377, with a focus on describing discrete and continuous variables and hypothesis testing in Python. * "Supervised Learning with scikit-learn": This was also largely a review of Math 378 (and also Econometrics 1) in which we learned the basics of linear regression and classification. Many of the methods it taught were the exact same as I learned in Math 378, including OLS, K-fold CV, Lasso, Bootstrapping, and K-nearest neighbors. * "Unsupervised Learning in Python": This course was a near replicate of our unsupervised learning unit in Math 378, with code in Python instead of R (which I think is easier, for the record). It taught about clustering, hierarchical clustering, and dimension reduction using PCA.

Looking back and moving forward Looking back, I have no regrets about the plan I chose. I really think if I had to do it again, I would pick the same learning plan to become well-acquainted with Python, and get a comprehensive overview of its capabilities.

However, in a perfect world (wouldn't that be nice?), I would have enough time to blaze through the courses so that I could practice on some more realistic projects. This goes back to that whole "Real experience comes from real, unguided projects" idea (see challenges listed above). Throughout the project, I frequently found myself wishing I was done with the DataCamp Courses so that I could practice and reinforce the things I had learned so far on personal projects. I had several ideas for projects I could do on my own as I was doing the course work. If I was forced to make at least one change, I would probably pick a career track with fewer courses and then make a writeup for an applied project that demonstrates understanding of course material. I would then turn in that mini project with this final project write-up that you are reading. As I mentioned above, however, I really am glad for the understanding of deep machine learning I gained from the last few courses of my career track, and I don't regret taking the heavier course load in the end.

From completing this project, I have an awareness of how much I can learn on my own with nothing more than a computer and internet access. I truly am an aspiring data scientist and aspiring Python wizard, and thus have many ambitions for continued online learning. I have already taken several steps to continue this. First of all, I have configured a professional-grade code environment in Visual Studio Code, which is something that DataCamp doesn't teach. Additionally, I am now subscribed to several data science blogs, and I have been exploring open source machine learning projects to see what the community is doing.

My plan to drive towards expertise in machine learning is to complete several personal projects, and thus learn by doing. The two projects I plan to complete this summer are 1) a Kaggle competition for predicting demand given online ads for Russia's version of Craigslist, and 2) a predictive model for the stock market that is accurate to a profitable degree. I intend to reference courses like these DataCamp courses, but to the end that I can apply what I'm learning to the real-world machine learning project.

I have thoroughly enjoyed this project, and I hope that I can improve my skill set until I am

credible enough to drive change in the world around me. And who knows? Maybe someday I'll be good enough at Data Science in Python to make an estimated $118,000 per year.

Documentation Statement None