Tell me about yourself – short and crisp

* I have a bachelors and masters degree in Electrical Engineering from UIUC and Georgia Tech both of which are top 5 schools for EE. After I graduated from Georgia Tech, I was hired by IBM in their Semiconductor development team. It’s a highly technical job, but I realized really early on that a huge chunk of the job involves driving value out of the data that we collect in order to maximize yield, performance and reliability… but the tools we were using were super basic and no one was using any state of the art data science techniques.
* So I asked my boss about 1 year into my career at IBM if I can do a second Masters but in data science and Machine Learning at UC Berkeley. He was pretty enthusiastic about it and convinced IBM to pay for my entire tuition. So I started doing full time work and full time school.
* I’d say his belief in me paid off, because a few months after I started at Berkeley, I built my first defect classification model on python. Then I started pulling data directly into python, I automated a lot of my work, and I started using machine learning as well. This jump started what we call the “transformation” protocol in my team and next thing I know, I was teaching my coworkers how to pull data and use python and some basics in data science.
* I’ve been coding in Python specifically in the realm of data science and machine learning since 2017 (7 years) but overall I’ve been coding since 2010 with my Electrical and Computer Engineering background. My strongest object oriented programming language by far is in Python and I am very experienced with SQL queries. But since I’ve been in tech for so long, I have experience with other languages such as C and R and Spark through work and school projects.
* So I have been working at IBM for 8 years now in hardware development, but my role had been pivoting strongly towards data science for the past 6 years. I think the next step in my career is to get a job as an official data scientist and machine learning engineer. I’ve learned a lot at Berkeley, and a lot on my own through my work at IBM, and I’m hoping to learn even more by working at Capitol One.

Projects to discuss

* Fin Res Metric
  + In my job, we deal with diagnosing defects that occur on our semiconductor microprocessors. If you can drive down these defects, you increase yield and save the company a lot of money.
  + Back when I first started studying data science at Berkeley, there was one particular defect that was very hard to diagnose and very time consuming to manually classify. Using what little I’ve learned at Berkeley at the time, I figured I will approach classifying these defects to at the very least automate the classification process but hopefully I’d be able to help my team find correlations easier
  + The first thing I did was build my own classifier with SQL and Python. I pulled all the relevant data of what this fail looks like and I coded up a list of instructions to help the machine label whether or not this defect has occurred or not. Then when I learned about decision trees and random forest, I build a random forest classifier and regression to label these defects, piggy backing off all the features I’ve built in my own version of the classifier.
  + Using the classifier’s outputs as a metric, I stacked up all the wafers with this fail and looked at which segments failed on each specific chip. It showed certain segments failed on these chips which showed that there are hotspots. This suggests some thing physically happening to cause these fails, maybe a tool issue. Then I used my metric to rank how bad each wafer is in each lot and use that information to combt through all our process steps and found a slot order correlation at the FC RIE cut process step. Futher digging showed that one of 2 tools was causing the issue. Using all this information we were able to
* Dashboards
* Change Point Detection
  + There was a specific parameter that shifted and it correlated strongly to a defect that showed up months later. The problem is that we have about 10 thousand parameters and a vast majority of them are not monitored by an engineer. I took the initiative to come up with a solution which might’ve been able to catch this parameter shift earlier and saved IBM a lot of money.
  + My task was to build an automated change point detection module which would feed in our 10 thousand parameters, sift through each of them to see if there was a significant shift in recent history, and then report the results for an engineer to do a deep dive on.
  + I ran a design thinking workshop with all the potential customers of this tool to get a sense of what kind of outputs they would want to do their job easier.
    - Then I experimented with a bunch of change point detection algorithms and landed on the Pruned Exact Linear Time (PELT) algorithm because it gave the best results.
    - Then I built a prototype of the data pipeline on Jupyter Notebooks and on my local computer which would recursively read in each of the parameters, run it through the PELT algorithm, and generate a detailed graph of the shift if it exists as well as log all the important details regarding the shift.
    - After the prototype is proven to work, I began improving it. I introduced multithreading which means we can process many parameters at once instead of one at a time, which significantly improved the speed. I moved the algorithm to a VM and set up a CRON job for it so that it would be automatically run every couple of weeks. Then the results are uploaded to a dashboard showing the details of the shifts that occurred and the people in charge of certain parameters would be notified of a shift
    - As a result, I created a tool that gave my team the ability to monitor data that it didn’t have before. It also generated a list of parameters to data mine and test correlations against.

Be memorable with a nugget about yourself

* Python
  + Scikitlearn, pandas, seaborn, dash, matplotlib, numpy, tensorflow, PyTorch, Plotly
  + Building python APIs? Spark?
* Spark
  + Language used to efficiently work with big data
  + Spark jobs create a DAG (Directed Acyclic Graph) of task stages to be performed on the cluster
  + RDD – Resilient Distributed Dataset: immutable distributed collection of datasets partitioned across a set of nodes of the cluster that can be recoverd if a partition is lost. Has built in memory computing and referencing datasets stored in external storage systems
  + DAG – Directed Acyclic Graph: set of vertices and edges where vertices are RDDS and edges are the operations to be applied to the RDD. Splits the work up into stages of parallel work and reduction and the internal scheduler figures it all out.
* AWS (S3, RS, EMR)
  + S3 – Amazon storage built to store and retrieve any amount of data from anywhere
  + EMR – Framework to build big data apps using Spark
  + EC2 – Cloud that you can customize to use on AWS server
* Kubeflow
  + Traditional development is put all the apps on one physical server. This is bad because these apps will interfere with each other, one might eat more CPU, etc.
  + Then came VMs. On the same machine, we can have lots of isolated OS but this is resource intensive and slow
  + Docker offers lightweight containerization platform where the host OS is shared but the apps live in containers. Fast deployment. Use docker to build, deploy and run containers, but it doesn’t help you manage all the containers unlike Kubernetes.
  + Kubernetes Containers – Kind of like VM split on a computer but more relaxed and light weight. It automates deployment, scaling and management of containerized applications. Not as fast deployment as docker but better for large-scale container management. Can be used with and without docker. You can deploy a container with docker and manage it with Kubernetes.
  + Kubeflow – helps you manage the deployment of machine learning workflows on Kubernetes simple, portable, and scalable.

Clear reason to apply to Capitol One

* I’ve heard and read that Capitol One has a great company culture and encourages a lot of growth which is one of the main things I’m looking for. I think it was ranked number 15 as a company to work for. Also I have a friend who works as a developer at Capitol One and he tells me that Capitol One is a financial company but it’s a technology company at its core. I also see that Capitol One is doing a huge push into AI and machine learning and I really want to see what I can achieve as a part of this push.

They want experience working on AWS

How do you monitor your machine learning module?