For several years I worked adjacent to data science teams. I would build the data lake, and the data scientists would use the lake contents to build models and then deploy those models. The first part was always much faster and smoother than the second. Teams would come up with great models but then spend an inordinate amount of time getting them into production. This is particularly true for teams using Airflow and pure Python based solutions vs. those using higher level tooling. In retrospect I think that some of the difficulty has to do with the different skillsets involved. In this module we’ve learned about scaling, encoding, transforming, and creating new features. At the end of this we build a model – we’re done with step one. Our model is based on data that has been through both a cleansing process and feature engineering. This pre-processing must be applied to any data that we want to use for prediction. Further, this must be done in a fault tolerant, production quality pipeline. That scaling black box and the model itself need to be persisted somewhere. Issues like we one-hot encoded the color, but then got a color that we haven’t seen before – how does that even work? I’m looking forward to finding or figuring this out.

Another thing that I found interesting was Evitan’s use of the cabin numbers. In one of our first learning responses, I expressed disappointment that Jedamski excluded this feature altogether. It seemed to me that proximity to warning, lifeboats, or exits would mean getting to the available lifeboats faster and therefore a better chance at one of the limited seats. By extracting this data and combining with some research Evitan demonstrated how some hard work can marginally increase the quality of a model.

I was surprised not only with the extraction of the title but also by the fact that it made a difference. This is a good lesson to not discount any data until it is explicitly excluded as not valuable to modeling. The titles are quite sparse, yet they provided incremental value above Pclass, which I would have thought sufficient by itself.