1. Problem: Unsupervised clustering of players in the FIFA soccer video game. I assume that the in game positions assigned to players are really not representative of all of the positions that really exist in soccer. I want to use the FIFA player rating dataset to find the “true” soccer positions. Full player rating datasets are available on Kaggle.

Success: Silhouette score may be a good place to start, but me just looking at the players grouped together and knowing the similarities intuitively may be a better measure of success. Basically, seeing if the results make sense in the real world given my soccer knowledge.

Risks: I think a risk here is that I will not find any meaningful relationships between that aren’t already obvious or already very clearly delineated by regular position names. There’s also the fact that better players have higher stats across all categories generally, so I don’t want to just group players by worst to best. Maybe take the ratio of stats compared to one another to alleviate this problem?

1. Problem: I have a “reach” final project scenario that may need some tweaking. Open to suggestions! We have a problem at work of matching Television and Movie titles that are written in different manners. For example, in our database, we may have “CSI: Miami Pilot” while the data we are sent may say “Crime Scene Investigation: Miami, Season 1 Episode 1”. Matching these two titles with a normal algorithm (Like FuzzyWuzzy) yields poor results, even though they are the same thing! Of course, this problem extends to titles written in different languages, but I think for simplicity I have to completely ignore that aspect for now. Conversely, “Ironman” and “Ironman 2” have very high normal match scores, even though they are very different. There are a few ideas I have to tackle this. A TF-IDF matrix is one idea and Word2Vec is another. I need large dataset of titles for either model. I had the thought of using the Wikipedia English corpus for training. I also need an extract of my company’s internal movie and series title database for the testing phase. I believe I can get my hands on this. I hope it doesn’t overtax my computer.

Success: I think the only measure of success is looking at the title matches that are predicted and see if that greatly outperforms the title match predictions that are already generated using the normal matching algorithms. This has to be by eye.

Risks: There is a risk it will be too much data for my computer. There is also the risk in the Word2Vec implementation that the series and episode titles themselves will not match that well as titles often don’t have a semantic connection as a normal sentence would. For example, “No Country for Old Men” doesn’t really mean anything outside of its title meaning. Word2Vec may capture semantic meanings in titles that don’t exist because they’re meaningless titles. A TF-IDF implementation doesn’t really carry that many risks as it’s really more like a weighting system for individual words or N-grams.