Proposal

June 16, 2020

1 Machine Learning Engineer Capstone Project Proposal

1.0.1 Background

Data Science as a field is relatively young, with new discoveries being made and new technique being developed at a breakneck pace. With this comes a variety of specialist roles flooding the job market as businesses try to find the very best talent to create cutting edge solutions.

Data Scientist who utilize their skills in both technology and social science to find trends and manage data; Data Engineers who prepare the "big data" infrastructure to be analyzed by Data Scientists; Machine Learning Engineers responsible for taking theoretical data science models and helping scale them out to production-level models. Each of these roles comes with it's own responsibilities and prerequisites, but you may have noticed a fair amount of overlap in their descriptions. Machine Learning Engineers and Data Engineers are often described as sitting "between Software Engineering and Data Science". Data Scientists are often expected to carry out the roles of a Data Engineer, especially in start-ups who can't afford seperate specialists.

These are simply the three most common categories. It's not unusual to see roles like Software Engineer with ML, Deep Learning Engineer, Insights Analyst and so on when look at roles in the field. Some even predict these roles will evolve or be eliminated in the near future (https://towardsdatascience.com/machine-learning-engineers-will-not-exist-in-10-years-c9cbbf4472f3). How is anyone supposed to keep everything straight this fast moving field?

1.0.2 Problem Statement

As a jobseeker, it can be confusing to know which roles are most relevant or suitable to ones given skillset, and the process of sifting through job adverts to find the right roles to dedicate time to can be tedious.

There are several issues that contribute to this: many businesses aren't clear on their own requirements and just using buzzwords to try to attract talented individuals with false promises; search engines are far from perfect, and it's not unusual for a search of "Data Science" roles to come up with more general but unrelated institutional research and scientific roles; position titles alone can be misleading, making brute force methods such as string pattern matching less effective.

Wouldn't it be nice if Data Science and Machine Learning, the source of this confusion, could help tackle this problem?

1.0.3 Solution Statement

Using NLP techniques developed earlier in the course, the project aims to **determine the relevance of a job advert based on the job description provided**. As part of my own work, I am developing a webspider to crawl job sites for Data Science positions. However, the spider is out of scope for this project, and a pre-existing dataset from Kaggle will be used instead (more on that later).

This project is merely a proof of concept to better understand the challenges involved. The expectation is that, in the unlikely event that the NLP model doesn't have perfect 100% accuracy on the first try /s, the process should reveal weak points for when my own crawler is complete. For example, perhaps there are additional features missing from the Kaggle dataset that could improve the model's performace, and this could be included in my own crawler.

Extensions (Optional) While also out of scope for this project, I'd like to discuss my intentions for future iterations. While the base model attempts to answer the question "Is the role relevant or related to Data Science and therefore worth an application?", future models will try to answer the following (harder) questions:

1) Can we determine the experience level required (Junion, Mid, Senior) better than string matching? This often isn't explicitly state, or job adverts list astronomical requirements on the number of years of experience to scare off those with impostor syndrome. Junior data scientist with 3-5 years experience, yeah right.

2) Can we estimate salary expectations, where not provided, based on company name/size/location? This will require extra data from sites like Glassdoor. What even is a competitive salary anyway, just give me a number.

3) Can we create a sequence-to-sequence model to identify relevant jobs from a list of requirements? i.e. I'm proficient in "Python, SQL, Spark", what are the most relevant jobs.

4) Can we autogenerate job responses/cover letters based on job descriptions?

1.0.4 The Dataset

The dataset comes from kaggle user Shanshan Lu who scraped 7000 US job listings from Indeed.com.

The following fields are found in the dataset: - Position - the position title/high level decription - Company Name - Description - detailed job description and requirements - Reviews - the number of review of the company, generally an indiciation of company size. - Location

For the purpose of this project, only Position and Description will be used to build the model. Positions will be used to categorise the descriptions as relevant or not relevant, as well as a breakdown into and prediction of specific roles.

1.0.5 Benchmark

For the purposes of benchmarking, the Scikit-learn Dummy Classifier will be used. The intention is to create a classifier that is only slightly better than random chance (if that) to provide an output for which we can determine the statistical significance of the NLP model.

1.0.6 Evaluation Metrics

Though there are several options, accuracy on the test set will be used as the primary evaluation metric. There is little benefit in this context to be gained from optimising for precision or recall, as there is no severe detriment in the case of a false negatives or a missed true positive beyond minor annoyance. A confusion matrix will be built as a secondary source of evaluation.

1.0.7 Project Design

The project will follow a similar design to review project from earlier in the course.

1) The data will be explored using a word cloud to try to identify correlations in the classes
2) Each role will be categorised based on the position title
3) The job description will be embedded using a bag-of-words model for prediction
4) A Scikit-learn dummy classifier will be created as a benchmark
5) The model will be based off of the Pytorch LSTM from earlier in the course
6) Both Dummy and LSTM models will be validated on the test set
7) McNemar's test will be used to determine the significance of the LSTM

The first pass will attempt to determine whether or not a role is relevant (i.e. a binary classification problem), while the second will try to predict the exact role from a set of 5 classes. This will require some more tweaks to the code in order to handle a multiclass problem.