Capstone Project Proposal

< Eric Armstrong >

Business Goal

Project Overview and Goal What is the industry problem you are trying to solve? Why use ML/AI in solving this task? Be as specific as you can when describing how ML/AI can provide value. For example, if you're labeling images, how will this help the business?	Using ML to screen large volumes of job applications: In the glamorous world of recruiting, screening applicant resumes at scale is a real challenge. More often than not, this involves a manual review of each resume: recruiters must compare what they read in an application with their understanding of the job, and make an assessment. Besides the time commitment, this screening process can be mentally exhausting for resume reviewers, and it invites a wide degree of bias and miscommunication in the earliest stage of the recruiting process. Some companies have tried to remedy this problem by using automated screening in their applicant tracking systems (ATS). A rule-based algorithm accepts the resumes containing the desired keywords, while rejecting the resumes that do not. While systems like these may ease the workload, they have not proven to be a sufficient solution. What we are proposing is a system that can predict, using machine learning, how likely a job application is to be moved forward by a recruiter, and on to interview. This will help businesses to optimize their management of incoming job applications or prospected/sourced candidates.
Business Case Why is this an important problem to solve? Make a case for building this product in terms of its impact on recurring revenue, market share, customer happiness and/or other drivers of business success.	Recruiting is a vitally important challenge for all companies to get right. A business succeeds not just because of good ideas, but because of the people willing and able to execute those ideas effectively. It's no secret that a great hire can revolutionize an industry, while a bad hire can cost the business more than just money. When viewed as a funnel, the first stage of the recruiting and interview process undoubtedly sees the most volume. Countless work hours may be spent spent combing through resumes, with typically upwards of half being rejected manually. It's an imperfect process, at that: sometimes good candidates are rejected, and sometimes bad candidates are advanced in the process. Recruiters are human and make mistakes, but there's plenty of room for improvement here. Building this ML product for application screening will have many benefits. Companies will spend less money on recruiters. Recruiters will be able to spend more time interfacing with candidates and conducting interviews. Candidates will receive decisions on their applications much faster. Finally, candidate quality will be able to be assessed and tracked in the early stages of the recruitment process.
Application of ML/AI What precise task will you use ML/AI to accomplish? What business outcome or objective will you achieve?	Precisely, our ML product will predict whether a recruiter would advance an application or reject an application from consideration. In the case of uncertainty, the product will also indicate that an application needs additional review by the recruiter. Based on these three categories, recruiters can then decide how to prioritize the applications and take action. In each future iteration, the system will learn from cases where it had to be corrected. The business objective will be to guide and automate a portion of the job application screening process, and thereby lower costs (fewer employees & hours needed). The system could be used to provide some 'training wheels' for new resume screeners, or a 'gut check' for more experienced recruiters. In an ideal scenario, this system will also increase revenue by helping to add high-impact talent to the organization.

Success Metrics

Success Metrics What business metrics will you apply to determine the success of your product? Good metrics are clearly defined and easily measurable. Specify how you will establish a baseline value to provide a point of comparison.	There are a ton of metrics that this system could impact. Here are a few: - Time it takes to process and potentially hire someone decreases - Number of interviews needed to make a hire decreases - Cost per hire decreases - Increased positive brand recognition, thanks to fast and reliable recruiting - Decrease in unconscious bias during resume screening - Fewer 'ghosted' job applicants, if applications are easier to sort and manage - Increased candidate quality: fewer false positive and true negatives - Boosted recruiter morale, they've got a dedicated and consistent helper To establish a baseline within a company, past resume screening metrics could be collected and analyzed. This would be helpful for showing correlation, but not causation. To show more of an effect, system usage could be randomly assigned to recruiters for some interval of time (stagger these intervals among users to control for variables). Then we could compare recruiting performance metrics with and without system usage.

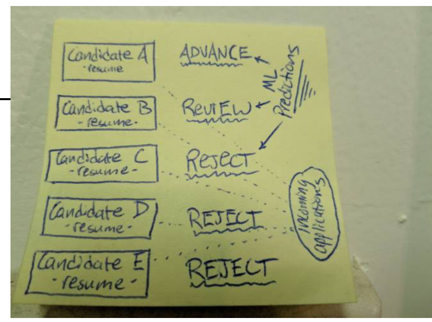
Data

Data Acquisition Where will you source your data from? What is the cost to acquire these data? Are there any personally identifying information (PII) or data sensitivity issues you will need to overcome? Will data become available on an ongoing basis, or will you acquire a large batch of data that will need to be refreshed?	Data will be sourced from existing companies. Most medium to large companies track recruiting information in their applicant tracking systems. For each business area, we will need full records for applications that were accepted in the screening process, and those that were rejected during screening. The cost to acquire this information will fall on the business itself, but some cost on our end will be incurred from extracting and redacting the text, as well as quality assurance (we need to verify what they had over is usable). As mentioned, the personally identifying information in this data will need to be removed. This includes anything personal that can be found on a resume: name, address, email address, phone number, any pictures, date of birth, etc. Since there is a wide degree of info that a resume may contain --depending sometimes on region and culture--, great care will need to be taken in identifying and removing what seems to be 'non-relevant' info. For the purpose of our system, data will ideally be available on an ongoing basis: jobs and people are constantly changing, and the system needs to be able to adapt to this as it happens. Otherwise, the tool will lose its value as new roles and backgrounds come along. For setting things up initially though, training etc., a batch should be fine. Or companies can start fresh, if they'd like - only collect and use data as it comes... and forget whatever dark recruiting past they had.
Data Source Consider the size and source of your data; what biases are built into the data and how might the data be improved?	Unfortunately, the size of the data will be small in most cases. It's not extremely common for companies to have application records in the tens of thousands or more. Most companies will be lucky to get somewhere in the thousands of applications. For now, we will just focus on a singular large company, but in the future there may be an opportunity to collect info from multiple companies across similar roles. One bias built into the data is the variability of recruiter screening ability. Some recruiters are very accurate and confident in screening, while others or perhaps more junior reviewers will not be as apt. Having some initial understanding of recruiters' standards, biases, and consistency is important, although not absolutely necessary. The data can be improved by removing major recruiter errors from the training data. The data could also be structured into relevant fields (titles, dates, descriptions, etc.
Choice of Data Labels What labels did you decide to add to your data? And why did you decide on these labels versus any other option?	The labels I decided to add to my data are "Advance," "Reject," and "Review." For the training data, the only cases will be Advance and Reject- since those are the only two options for viewing the past application decisions. For the predictive labels the system produces, Review will be added. This will apply to any applications that don't seem to meet the threshold for either Advance or Reject. I want users to be reminded of the uncertainty and skepticism needed in reviewing applications, and not just relying on a binary decision that may in fact have been a very close call. Any other labels would not be as intuitive as these three simple action verbs. These data labels are not without their weaknesses, however. For starters, they reductionist in their presentation of recruiting outcomes. They also give no information on the rationale behind a rejection, leaving the recruiters to wonder. Knowing the reason for rejection (ex: too junior, overqualified, visa, lacking skills, incomplete application). Potentially worse, recruiters could misinterpret these 3 predictions as directives, and categorically reject, advance, and review candidates "as the system tells them."

Model

Model Building How will you resource building the model that you need? Will you outsource model training and/or hosting to an external platform, or will you build the model using an in-house team, and why?	Since I'm familiar with Google AutoML, that may be the ideal resource for building the model I need. Financially, it would not make sense to hire a costly ML engineer so early on. What is important for now is testing the concept on a small scale before making gradual improvements and scaling. Nothing would be worse than sinking time and resources into this only to find out that it doesn't work as intended, or that businesses are uninterested in using it. If things really pick up, we will need to assemble a team of relevant specialists: ML engineer, data scientist, software engineer, database engineer, solutions and integrations engineers, etc. This way we'll have much more control over the fine-tuning and deployment of the system.
Evaluating Results Which model performance metrics are appropriate to measure the success of your model? What level of performance is required?	Concerning model performance metrics, we would like to have both high precision and even higher recall: Precision- In training, was the system able to identify a strong ratio of true 'Advance' cases over all cases (true and false positives)? We don't want too many unqualified candidates being moved forward in the process, although the risk in that is low. Recall- It's very important that we maintain the highest recall possible. The last thing we would want is for the system to unreasonably reject someone who is actually qualified for an interview opportunity. Not only does that reflect poorly on the company, but it's an easily missed chance to bring in valuable talent. A costly mistake indeed.

Minimum Viable Product (MVP)

Design What does your minimum viable product look like? Include sketches of your product.	
Use Cases What persona are you designing for? Can you describe the major epic-level use cases your product addresses? How will users access this product?	This is Christina. She is a recruiter at a large tech company hoping to hire aggressively in the coming months. Most days, Christina spends her morning reviewing incoming job applications for a few different open positions. This takes about 45 minutes per role on average, which can feel quite mind-numbing after a while. Thankfully, her team has had access to a tool that helps with some screening. She opens the company's Applicant Tracking System in a browser, which has this new tool added as an extension. Looking into one role, she can see that all the new applications have been grouped and labeled into 3 different classes- Advance, Reject, and Review. She briefly looks over the applications in Reject just to be sure she's not missing any thing good, but the ML seems to be working. She turns down these apps, which removes them from the queue and send rejection emails. Next, Christina can review the Advance resumes- setting up calls for the best ones, and keeping the remainder in reserves. She spots one 'Advance' profile that's not a match, and so rejects. She heard the system will learn from this valuable feedback in the next month. Finally, she has time to review the edge cases in Review. Most of them she is able to make a decision, but for a few she plans to consult with some peers before taking action. In total, Christina spent an hour on what normally would have taken her up to 3 hours.
Roll-out How will this be adopted? What does the go-to-market plan look like?	The roll-out of the product will be gradual. Once the MVP has been tested, we plan to move to beta testing with a select company. Ideally, we would like a company that has a reasonably large volume of applications, across time, for fairly similar roles. Additionally, we need to do this beta testing only where we can see seasoned and reliable high-volume recruiters. The best results depend upon accurate training data, and lots of it. Once we have proved feasibility and sound ROI with the beta test, can start rolling out the system to similar companies- in hopes that it will be just as successful there. When we get enough clients to vouch for our performance, we can advertise and grow. A good approach could be to specialize in one type of job first before branching out. Since there's high demand and supply, software engineers may be a good target.

Post-MVP-Deployment

Designing for Longevity How might you improve your product in the long-term? How might real-world data be different from the training data? How will your product learn from new data? How might you employ A/B testing to improve your product?	Thinking long-term, there are a number of improvements that can be made or planned. Much of this will revolve around course corrections needed to maintain our goals of high precision and higher recall. We will collaborate with clients to make the UI as functional as possible for them, and also advise them on the best way to use the product (it is not meant to be a fully automated solution, and we will advise against using it like that). Real-world data may be different than training data, especially if resumes change or evolve. If candidates start including some new and highly impactful information, we will need to take that into account. The real-world data will also contain information that is redacted in the training. We would always advise that companies review anonymized applications to cut down on implicit bias, but that is out of our control unfortunately. Our product will learn from the new data. Depending on the company's needs, we will occasionally batch and send recent job applications and decisions to be used in updating training data. Setting up a real-time system could be worthwhile down the road, but it's an overkill for now. For A/B testing, models
Monitor Bias How do you plan to monitor or mitigate unwanted bias in your model?	Unwanted bias is a huge hazard in this system. If we see that unwanted bias starts arising, we can take action to change our approach. We can start using training data that is a greater reflection of the results we would like to see. Additionally, we can redact any additional resume info that seems to be the source of some bias- and really experiment to find the right balance. What we want to avoid is turning companies into echo-chambers if the system converges on one or more types of individuals. Hiring discrimination is a very real thing, and the last thing we would want is for our system to embody biased screening. We need to be selective of the companies we engage with, and look for existing in diversity backgrounds (sex, gender, ethnicity, nationality, religion, thought, ability, etc.).